Highly-Efficient Bulk Data Transfer for Structured Dissemination in Wireless Embedded Network Systems

Zhiwei Zhao[a], Zi Wang[a], Geyong Min[b], Yue Cao[*,c]

[a]College of Computer Science and Engineering, University of Electronic Science and Technology of China, China
[b]Collge of Engineering, Mathematics and Physical Sciences, University of Exeter, UK
[c]Department of Computer Science and Digital Technologies, Northumbria University, UK

Abstract

Recent years have witnessed the remarkable development of wireless embedded network systems (WENS) such as cyber-physical systems and sensor networks. Reliable bulk data dissemination is an important building module in WENS, supporting various applications, e.g., remote software update, video distribution. The existing studies often construct network structures to enable time-slotted multi-hop pipelining for data dissemination. However, the adopted transmission mechanism was originally designed for structureless protocols, and thus posing significant challenges on efficient structured data dissemination. In this paper, we investigate the problem of structured bulk data dissemination. Specifically, we propose reliable out-of-order transmission and bursty encoding mechanisms to transmit packets as many as possible in each transmission slot. As a consequence, the resulting transmission protocol (ULTRA) can fully utilize each transmission slot and propagate data in the network as fast as possible. The performance results obtained from both testbed and simulation experiments demonstrate that, compared to the state-of-the-art protocols, ULTRA can greatly enhance the dissemination performance by reducing the dissemination delay by 34.8%.

Keywords: wireless embedded network systems, bulk data dissemination, structured protocol, transmission mechanism

---

1. Introduction

Wireless embedded network systems (WENS) are composed of a number of low-power embedded devices which are capable of computing and wireless communications [1, 2]. WENS have been widely deployed recently to support Internet-of-things [3, 4, 5, 6]. Bulk data dissemination is used to distribute a large data object reliably from a sink node to all network nodes in WENS, becoming an essential building module for a variety of WENS systems, e.g., remote software management [7], security patches [8], reprogramming [9, 10] and video distribution [11].

The existing studies [12, 13, 14] often employ a Connected Dominating Set (CDS) structure for bulk data dissemination. In such a structure, a set of nodes are selected as core nodes, which are responsible for disseminating the object to the rest nodes that are one hop away from the core nodes. This facilitates the transmission and sleep scheduling to achieve more efficient data propagation. Structured dissemination leads to less broadcast overhead as compared to structureless dissemination which is prone to the broadcast storm problem, and hence, offers a good solution for dense and low-power WENS.

Specifically, a data object is divided into small pages, each of which consists of a number of packets [12, 13, 14]. Time is sliced into fixed slots, and the slot length approximates the transmission time of a page. Each node operates in three types of slots for transmission (T slot), reception (R slot) and sleep (S slot), respectively. Then the data propagation is done in a page-by-page manner: a node starts transmitting the next page only when the current page is entirely received by all its child nodes in the structure. The page-by-page design is motivated by two reasons. First, it enables multi-hop pipelining. Different pages can be simultaneously transmitted at different hops, thus reducing the overall propagation delay. Second, to establish retransmission and ensure eventual reliability, a bitmap indicating

---

[*]Corresponding author

Email addresses: `zzw@uestc.edu.cn` (Zhiwei Zhao), `ziv_wang@std.uestc.edu.cn` (Zi Wang), `g.min@exeter.ac.uk` (Geyong Min), `yue.cao@northumbria.ac.uk` (Yue Cao*,)

the missing packets is carried by a request message (REQ). The bitmap for the entire object may be too large for the limited payload length (e.g., 127 octets in 802.15.4 packets). Therefore, the bitmap for a page is much smaller and can be included in an REQ message. The eventual reliability is achieved when all pages are received.

While the above mechanism can roughly establish reliable page-level pipeline for structured protocols, it suffers from the following key limitations. First, the transmission slots are not fully utilized. In current dissemination protocols, a node transmits the packets requested by its child nodes. Although the slot length approximates a page transmission time, *the slots are rarely full of transmissions* since the requested packets in REQs are often fewer than the page size. Second, the three-way handshake, which works in the Advertisement-Request-Data (ADV-REQ-DATA) paradigm, produces *redundant controlling overhead*. The ADV messages are designed for two purposes: neighbor discovery and new data page declaration. However, when applied in the structured dissemination, neighbor discovery is no longer necessary since each node has fixed parent and children. If we disable the page-by-page transmission and transmit out-of-order packets to fully utilize the slots, the data page declaration is neither necessary.

In this paper, we investigate the transmission for structured data dissemination and propose a f**ul**l-slot reliable **tra**nsmission mechanism (called *ULTRA*) for bulk dissemination, which fully exploit the benefits of the structure. ULTRA has three salient features. 1) We break the page-by-page transmission manner and enable out-of-order packets transmission. Instead of only transmitting the packets within one page, a node transmits as many packets as possible in each slot. 2) ULTRA adopts a bursty coding scheme to fully utilize the transmission slots when there are not enough native packets during the propagation, which effectively recovers packet errors without the need of extra transmission rounds. While the above two designs greatly improve the data propagation speed, they pose new challenges in terms of reliability. 3) We propose a slot optimization model, specifically for the proposed transmission mechanism to minimize the propagation delay.

We implement ULTRA in low power wireless embedded TelosB nodes and conduct both testbed and simulation experiments. The evaluation results show that ULTRA outperforms the state-of-the-art bulk data dissemination mechanisms and greatly enhances the dissemination performance by reducing completion delay by 34.8%.

The major contributions of this paper are summarized as follows:

- We investigate the limitations of the existing structured dissemination protocols in WENS and propose an out-of-order, XOR enabled transmission mechanism to fully utilize the transmission slots.

- We propose a slot length optimization model, tailored for the proposed full-slot transmission mechanism to further enhance the propagation.

- Based on the above components, we propose a holistic solution, i.e., full-slot reliable transmission for structured bulk data dissemination (ULTRA). Compared to the existing protocols, ULTRA can fully utilize each transmission slot and greatly reduces the negotiation overhead.

- We implement ULTRA in both TelosB [15] testbed and TOSSIM [16, 17] simulation. The performance results show that ULTRA greatly outperform the state-of-the-art protocols in terms of the end-to-end delay.

The remainder of the paper is organized as follows. Section 2 summarizes the related works and compares ULTRA with them. Section 3 presents the motivation of our work. Section 3.2 identifies the design opportunities for efficient dissemination. Section 4 presents the main design of the ULTRA protocol. Section 5 evaluates ULTRA via testbed and simulation experiments. Section 6 concludes the work.

2. Related Works

The existing bulk data dissemination protocols can be mainly classified into two categories: structureless protocols and structured protocols.

2.1. Structureless dissemination

Structureless approaches mainly include Deluge [18], MNP [19], ECD [20], etc. These protocols employ three-way handshake and NACKs to ensure reliability [21, 22, 23], and divide code images into pages to enable pipelining. Deluge is the default dissemination protocol in TinyOS [24]. It randomly selects forwarders and transmit data objects in page-by-page manner. Based on Deluge, MNP and ECD provide sender selection algorithms to reduce the number of concurrent senders. MNP also adopts sleep scheduling by turning off the radio if a node fails in sender contention phase. ECD enables dynamically configurable packet sizes [25] to further improve the performance. Some other approaches also employ rateless coding to enhance the dissemination propagation [26, 27, 28]. The main difference of these works from Deluge is it transmits encoded packets instead of native packets. Compared with structured dissemination, structureless dissemination requires much negotiation overhead for choosing better links. As a result, structureless dissemination is more applicable for sparse networks with highly time-varying links while structured dissemination is more suitable for dense networks where negotiation may cause serious collisions and delay.

2.2. *Structured dissemination*

Structured dissemination have less broadcast overheads as compared to structureless dissemination which is prone to the broadcast storm problem, and hence offer a good solution for dense and low-power wireless embedded network systems. We focus on the structured protocols, with a specific interest in the transmission mechanism.

Structured dissemination protocols including Sprinkler [12] and CORD [13] typically build a topology structure, e.g., CDS, before data dissemination, in which all nodes are divided into two categories: core nodes and non-core nodes. Each non-core node is associated with a core node. Data dissemination is conducted in two phases. First, the sink transmits the data object to all the core nodes; then each core node disseminates data to all its neighboring core nodes.

Sprinkler [12] requires geography information and tends to establish a minimum connected dominating set (MCDS). The rationale is that by minimizing the number of core nodes (forwarding nodes), the number of transmissions can also be minimized.

CORD [12] follows the same principle as Sprinkler but improves Sprinkler in two ways. First, CORD considers link quality when constructing the core structure. It first eliminates the poor quality links, and then selects the node with the most neighboring nodes in a neighborhood as a core node. Second, CORD enables coordinated schedules by employing object segmentation, page-by-page transmission, and three-way handshaking. Coordinated schedules divide time into three fixed-size slots: P, C and Q, for transmitting, receiving and sleeping, respectively. In slot TX, a node acts as a parent, broadcasting ADV messages to inform downstream nodes of its received pages, and transmits data packets within certain page when REQ messages are received. In slot RX, a node acts as a child, transmitting REQ messages when receiving ADV messages that contain more pages, and then receives packets from its parent node. In slot S, a node turns off its radio until the slot ends to save energy consumption. Note that the three slots have an equal length. CoCo [14, 29] is a recent structured dissemination work established on the sleep scheduling considering link correlation [30]. The key difference of CoCo from CORD is to comprehensively consider link characteristics during the core structure construction.

The novelty of the proposed ULTRA protocol: All these works start the three-way handshake enabled, page-by-page transmission after the structure is established. The transmission, however, as we will discuss in Section 3, suffers from the slot under-utilization problem.

Instead of designing yet another new core construction approach, we aim to facilitate efficient transmission mechanism that can fully utilize the potential of the underlying structure. More specifically, we abandon the widely employed three-way handshake and design a novel f**ul**l-slot reliable **tra**nsmission mechanism (ULTRA). Compared to the existing works, the novelty of ULTRA includes: (1) ULTRA adopts a novel DATA-REQ paradigm, instead of the ADV-REQ-DATA handshake, reducing much negotiation transmission and delay overhead. (2) Besides, the packets are sent in an out-of-order manner. A sender always tries to send packets as many as possible to its child nodes, thus the data object can be pumped into the network as soon as possible. (3) We design a novel bursty XOR coding, to fill up the slots when there are not enough native packets to send. The proposed coding can efficiently recover the packet losses without the need of retransmission. It is worth noting that compared to the network coding based approaches [26, 27], ULTRA has two main differences. First, while the network coding based approaches can also occupy the entire slots, they are essentially based on the page-by-page transmission. The reason is that only when the receivers have recovered the entire page, the sender is able to transmit the encoded packets of the next page. Otherwise, the receiver will be unable to decode the native packets for further encoding and forwarding.

3. Motivation and Design Opportunities

In most of the structured dissemination protocols for WENS, all network nodes first establish an underlying structure and start dissemination along the structure (for stationary networks). These protocols exploit various link characteristics to construct the core structure [12, 13, 14, 31, 32, 33] and ensure reliability using the three-way handshake mechanism, where a sender node first broadcasts ADV messages, declaring its source ID and the data it can provide. A receiver node which overhears the ADV will reply REQ messages, requesting the packets in need. Then the sender will start data transmission after the REQ messages are received. Structured dissemination works in a time-slotted manner, where each node works in the RX-TX-SLEEP cycle. A node receives packets in RX slots, transmits packets in TX slots and turns off the radio for energy efficiency in SLEEP slots.

The representation of the data pages and the requested packets are done by virtue of bitmaps. When the data object is large, the bitmap will be long and may exceed the limited packet length of the embedded nodes. For example, the maximum 802.15.4 packet payload is 127 octets. However, a data object of 4000 packets is represented in a bitmap of 4000/8=500 bytes, which cannot be transmitted in a single REQ packet.

3.1. Motivation

To deal with the problem and enable three-way handshake, the current dissemination protocols divide a data object into small pages, and propagates the data object in a page-by-page manner. As a result, a node can transmit the packets in a specific page only when it has received the entire page; And the next page can be transmitted (is requested) only when the current page is received by all receivers. This design brings the following benefits.

- The bitmaps in REQ messages are shortened. Instead of a long bitmap for the whole image, a small bitmap for only one page can be enough to indicate the missing packets. The reliability is ensured page by page.
- ADV messages can be short. In *structureless* protocols, ADV messages are periodically broadcasted, and it is important to reduce the ADV message size. Using the page-by-page transmission, One number *n* in the ADV message is enough to declare that all pages with page numbers smaller than *n* can be provided by the source node.
- It explicitly establishes data pipelining by propagating different pages at different hops.

However, the above benefits are mainly for the data negotiation, which may no longer exist when applied in structured dissemination because there is no need for periodic negotiation. Moreover, the page-by-page transmission poses significant challenges on improving the dissemination performance. Specifically, the designs for enabling the three-way handshake often leads to that many slots are under-utilized and thus the efficiency of pipelining is greatly reduced. Next we discuss on the limitations and design opportunities.

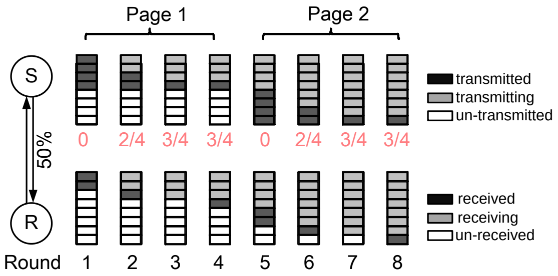


Figure 1: Illustration of the page-by-page transmission: The red numbers denote the wasted transmission opportunities, showing that some slots are significantly under-utilized.

3.2. Analysis of Design Opportunities

In this section, we analyze the design opportunities for structured bulk data dissemination.

3.2.1. Single hop transmission

Figure 1 shows the page-by-page transmission process. Suppose the data object contains eight packets and is divided into two pages (each consisting of four packets). The slot duration is the transmission time of an entire page. The percentage beside the links denote the corresponding packet reception probability. In the first slot, four packets are transmitted and two packets are lost. In the second slot, node S retransmits the two missing packets and one packet is lost. In the third and fourth slots, node S retransmits the last missing packet.

We can see that eight slots are used to disseminate the data object. For each page transmission, only the first slot is fully utilized and much fewer packets are transmitted in other slots. We denote the wasted transmission opportunities using the red colored fractions under the slots. It is clear that many transmission opportunities are wasted especially in Slots 2-4, considering node S actually has more packets to send in the next page.

3.2.2. Multi-hop transmission

To reduce the ADV size, it is required that a node can start forwarding a new page only when it has received the entire page, such that one number *n* in the ADV message can indicate that all pages smaller than *n*.

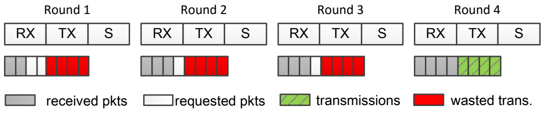


Figure 2: Illustration of the multi-hop propagation (node R in Figure 1): three transmission rounds are needed before it can forward the data to the next hop nodes. The transmission opportunities in first three TX slots are wasted.

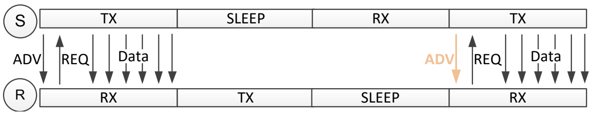


Figure 3: ADVs are transmitted in each TX slot in current three-way handshake.

However, the downstream nodes, which have already received some page fragments, may under-utilize the transmission slots.

We take into account the same topology as shown in Figure 1, and focus on the data propagation from R to its next hop receivers. Figure 2 shows first four rounds of periodic slots of node R (a round denotes a cycle of RX-TX-DATA). In the first round, R has received two packets in the RX slot, but transmits no packets in its TX slot because it has not received the whole page. We can see that in the three TX slots before the full reception of the page, no packets are transmitted to R's receivers, although it has already received some packets for forwarding.

To address the above problems and fully utilize the slots, an intuitive approach is to directly fetch packets from the following pages for transmission. However, under the current framework, if a sender transmits mixed packets from different pages, its receiver will not be aware of *how many and which packets* the sender has sent. As a result, the receiver will be unable to compose bitmaps indicating the missing packets and thus the retransmissions cannot be done.

3.2.3. ADVs in structured protocols

ADVs are used for neighbor discovery and data declaration in structureless protocols.

Figure 3 shows the typical transmission slot in the current structured dissemination. In each slot, the sender node always transmits an ADV message first to claim which pages it can provide, then the receiver replies an REQ message for requesting the packets in need.

When in structured protocols, each node has fixed parent node and the neighbor discovery is no longer necessary. Also, during a page transmission, there is no need for page declaration either. Considering the limited benefits, and considerable transmission, delay and contention overhead incurred by ADVs in each slot, we argue that the ADVs should be eliminated and the reliability should be carefully re-designed for the Transmission-Request paradigm.

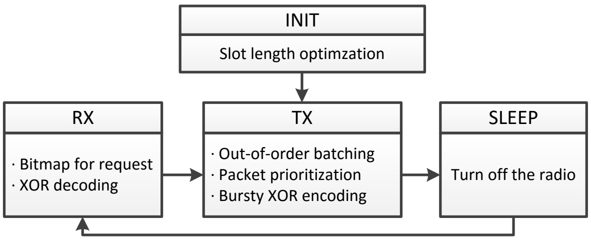


Figure 4: State transition of ULTRA.

3.2.4. Summary

From the above analysis, we can identify two key designing opportunities. First, the slot is not fully utilized in the existing protocols, both in terms of single hop page propagation and multi-hop propagation. Second, the ADVs brings few benefits to the performance but incurs considerable transmission and delay overhead. To fully exploit the transmission slot and enhance the propagation, our key idea to enable out-of-order transmissions, i.e., a node can transmit packets as many as possible to its downstream nodes and these packets are not required to be within the same data page. When there are not enough packets for a full-slot transmission, we use network coding to fill the slot, which is expected to recover the packet errors without retransmissions. Besides, we would like to eliminate the ADV messages and carefully design the transmission and bitmaps in REQs.

4. Main Design of ULTRA

In this section, we present the design of ULTRA. We first give the high level overview and then present each building block of ULTRA in detail.

4.1. System overview

Figure 4 shows the state transitions of ULTRA. When the structure is established, each node operates in RX-TX-SLEEP modes. Figure 5 shows the multi-hop pipelining with ULTRA. The green parts denote differences from the existing page-by-page transmission.

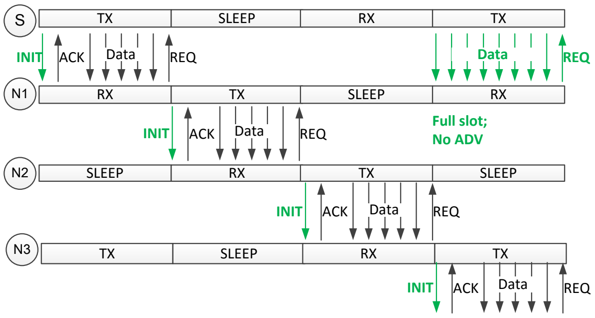


Figure 5: Structured data propagation with ULTRA. The green parts denotes the difference from the page-by-page transmission.

ULTRA starts with an INIT message, informing the network nodes about the start of the dissemination. The INIT message contains the object size S_o and slot length (denoted as the number of transmissions within one slot, n). The slot length is obtained by the slot length optimization scheme (**Section 4.4**). After receiving the INIT message, each network node allocates two local bitmaps with $\lceil S_o/8 \rceil$ bytes to indicate the packet to receive and the packets to send. When all packets are received at all network nodes, the dissemination is finished.

After the initiation, ULTRA starts working in the "transmission-request" paradigm. In a TX slot, a node first composes n packets for a full-slot transmission using the out-of-order packet batching scheme (**Section 4.2**). The packets that are requested by more receivers are prioritized so that the transmission can be more beneficial (**Section 4.2**). Each packet is added with a slot sequence to mark its position in the current slot. The sequence is used for its receivers to identify missing packets. Then after n transmissions, the node waits for the REQ messages and updates its packets-to-send bitmap using the received bitmaps. When there are not enough packets for a full slot, ULTRA adaptively adds XOR encoded packets to fill up the slot (**Section 4.3**).

During the RX slot, a node receives the out-of-order data packets and records the missing packets in the current batch, using the slot sequence in each packet. At the end of the RX slot, the node transmits the bitmap back to its sender in REQ messages.

The above process repeats until the whole data object is received by all receivers.

4.2. Packet prioritization and out-of-order transmission

We enable the out-of-order transmissions, and transmit n packets in each slot. To maximize the propagation benefits of each slot transmission, the first n most "beneficial" packets are always chosen for transmission.

The benefits of the packets are denoted as the utilities, i.e., how many receivers have not received the packets. The higher the packet's utility, the more beneficial it would be to transmit the packet. For example, a packet with utility=2 is more beneficial then a packet with utility=1, and should be prioritized for transmission.

Therefore, we record the utilities for each packet at the sender side, and prioritize the packets with higher utilities. When composing the packets to send during a slot, the sender picks the first n packets according to the utilities.

It is possible that there are not enough native packets for a full slot transmission. Under such situation, we use XOR coded packets to utilize a full slot (Section 4.3). These coded packets can possibly recover missing packets without retransmissions.

Reliability for out-of-order transmission. In the three way handshake, each receiver is receiving packets in the same page. Therefore it is easy to identify which packets are lost at different receivers.

However, when the transmissions are out of order, the receivers are unaware of which packets have been sent by the sender, and thereby cannot identify which packets have been lost in the current slot.

To deal with this problem, we add a slot sequence number in each packet, indicating the index in the current transmission round. The receivers can identify which packets are lost in the current slot, and then reply the bitmap back to the sender. The sender then marks the bitmap's corresponding positions as packets for retransmission. It is worth noting that the slot sequence is stored in the five reserved bits when $n \leq 32$, thus incurring no extra overhead. When $n > 32$, several bits overhead will be required in the payload.

Different from the traditional approaches, the missing packets are not immediately transmitted. The reason is that the missing packets are no longer useful for all receivers, and the packets which have never been sent are more beneficial. The retransmission follows the utility based transmission.

4.3. Bursty encoding

When there are not enough native packets, we add coded packets to fill up a full slot. Several existing network coding approaches use fountain code [27, 34, 35] or random linear code [26] for error recovery. However, these approaches often need Gaussian Elimination, which is time consuming for the resource constrained wireless embedded devices.

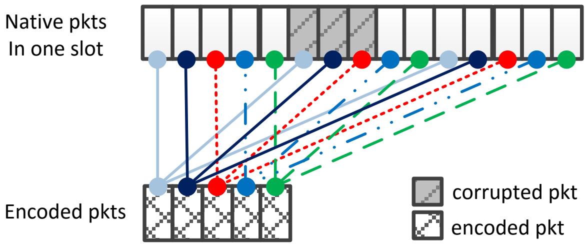


Figure 6: The bursty XOR encoding.

For example, the decoding of 20 random linear coded packets will require ∼100ms for decoding in the MSP430 platform.

In this subsection, we aim to design a novel coding scheme which is lightweight as well as capable of forward error recovery, exploiting the link burtiness opportunities.

4.3.1. Link burstiness

Many recent works [36, 37, 38] have observed that the packet errors are most likely to be bursty, i.e., the erroneous packets are often consecutive. As the key of network coding is to generate the non-linear combinations of the native packets, the consecutiveness poses new chance for lightweight error recovery code.

The key for error recovery is to combine the erroneous packets into different encoded packets, such that these packets can be decoded by XOR the encoded packet and the combined native packets. Figure 6 shows the typical bursty packet errors, where three packets are corrupted in a row. If we can distribute the packet errors into three different encoded packets, then they can be recovered. However, the challenge is that the sender can never know which packets will be corrupted before transmission.

4.3.2. Modulo encoding

We exploit the burstiness and propose a modulo encoding approach. Our key insight is that although we cannot know which packets are corrupted, we know that they are most likely consecutive. Therefore, if we can estimate the number of packet corruptions, It is possible to use the modulo operation to separate the packet errors. We explain the modulo encoding as follows.

Suppose the packet reception probability is p, and m native packets are transmitted in an n-packet slot. We can expect $m(1-p)$ *consecutive* native packet losses, and the goal of the encoded packets is to recover these $m(1-p)$ packets.

We pick one native packet every batch of $m(1-p)$ packets for XOR coding, such that all packet errors will be in different XOR encoded packets. Therefore, the missing packet can be decoded by the XOR operation of the encoded packet and other combined received packets. More formally, each encoded packet in a slot is obtained by:

$$E_i = \bigoplus_{k\%N_e=i} p_k, k \in [0, N_p-1] \tag{1}$$

where E_i denotes the ith encoded packet, p_k denotes the kth native packet, N_p denotes the number of native packets in the current slot, and N_e denotes the number of encoded packets in the current slot.

Consider the example shown in Figure 7, there are 15 native packets left for a 20-packets slot. The link quality is 0.8, and there are 3 consecutive missing packets expected. Then we add the five packets using modulo-5 XOR coding. The three corrupted packets are effectively distributed into different encoded packets, such that the receivers can recover all the three packets by the XOR operation of the encoded packets and the combined native packets.

When the number of encoded packets is larger than the number of native packets, the above coding will not work well. However, under such case, it means that the slot allows for retransmission of all the native packets. And the room for encoded packets can be filled up by the native packets.

4.4. Slot length optimization

The slot length is of paramount importance for reducing the dissemination delay. Considering the case where each slot is long enough to transmit the whole data object, the negotiation delay is minimized since each node is required to reply REQs only once. However, no pipelining will be exploited in such a case and the propagation delay is likely to increase. From the above case, it can be inferred that: When the slot length increases, the negotiation delay is likely to decrease and the propagation delay is likely to increase. When the slot length decreases, the negotiation delay is likely to increase but the propagation delay is likely to decrease.

To achieve a good trade-off between the negotiation delay and the propagation delay, we model the relationship between slot length and the end-to-end delay performance. Then we can obtain the optimized slot length for ULTRA.

We assume to disseminate a data object containing N packets to a h-hop network. The slot length is denoted

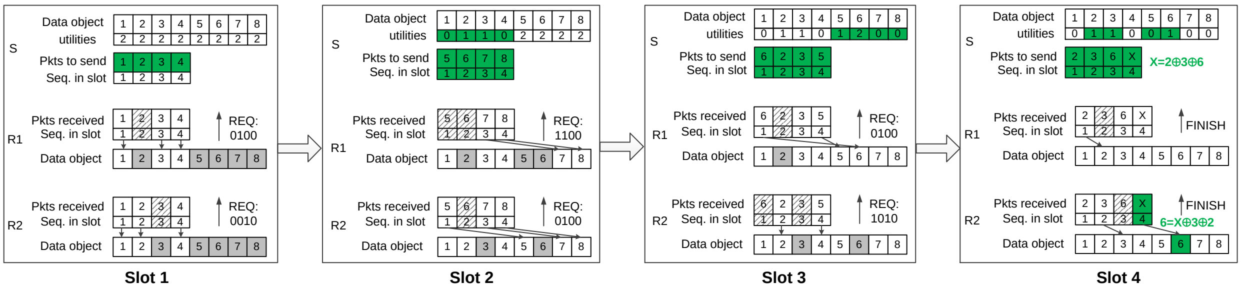


Figure 7: Illustration of ULTRA transmission: Node S tends to transmit an eight-packet object to two receivers R1 and R2, and four packets can be transmitted at most in one slot.

by the packet batch size n, which can be transmitted in one slot. Then the expected delay can be estimated by:

$$D_{overall} = D_{firstArr} + D_{prop} \tag{2}$$

where $D_{firstArr}$ denotes the delay when the first batch of packets arrive at the last hop and D_{prop} denotes the propagation delay of the remaining batches. $D_{firstArr}$ is given by:

$$D_{firstArr} = h\tau\frac{n}{q} \tag{3}$$

where τ denotes the transmission time of a single packet, q denotes the averaged link quality and $\frac{n}{q}$ denotes the expected transmissions for a batch.

Considering the multi-hop pipelining (Figure 5), when the first batch of packets arrive at the hth hop nodes, the second batch is propagated to the hop $h-3$. D_{prop} is given by:

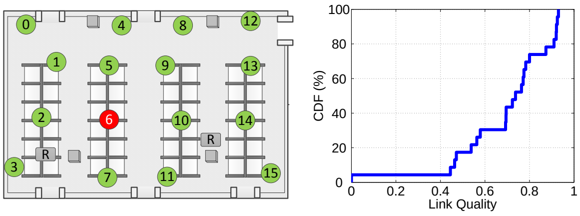
$$D_{prop} = 3\cdot\frac{n}{q}(\frac{N}{n}-1) \tag{4}$$

Combining Eqs.(2∼4), we can calculate the expected dissemination delay using the slot length n. The optimized slot length n can be obtained by solving the following equation.

$$D'_{overall}(n) = 0 \tag{5}$$

4.5. Integration of all components

We use Figure 7 to show how ULTRA disseminates an eight-packet object in the same settings with Figure 1. Recall that there are one sender S and two receivers R1 and R2. S tends to transmit an eight-packet object to R1 and R2. The slot is set as the transmission time of four packets. In the first TX slot of node S, it transmits packets 1-4, and the receivers reply the REQ messages indicating the missing packets. In the second TX slot of node S, after updating the utility of each packet, S picks the four packets with the most utilities (i.e., packets 5-8 with utility=2) for transmission. R1 and R2 identify the missing packets using the in-slot sequences, and reply the bitmaps. In the third TX slot, although there are no four consecutive packets for transmission, S still picks four out-of-order packets to fully utilize the slot. Packet 6 is firstly picked because it has the highest utility (2), and packets 2,3 and 5 with utility=1 are then picked to compose a four-packet batch. In the fourth slot, there are only three packets (2,3,6) which are not received by at least one receivers. We pick all the three remaining packets, and add up a XORed packet X (X=2⊕3⊕6) for a full-slot batch transmission. After the transmission, receiver R2 still loses packet 6, but can effectively recover packet 6 by XORing packets X, 2 and 3.



(a) The TelosB testbed.

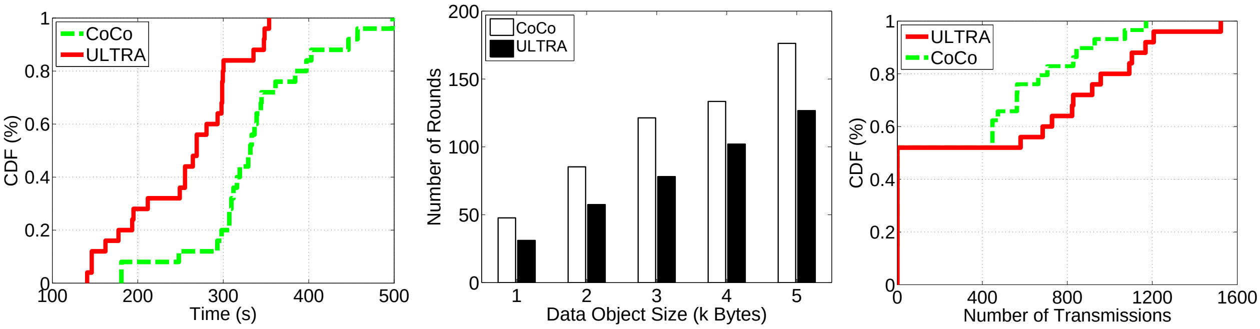
(b) Link quality distribution.

Figure 8: The testbed.

Compared to the page-by-page transmission [13, 14] (shown in Figure 1), ULTRA disseminates the whole data object in only four slots, while eight slots are used by the page-by-page scheme. The improvement comes from the out-of-order transmission and the XOR coding.

4.6. Discussion on the coding design

One concern is that why not directly using network coding to fully utilize the slots? We discuss this question by comparing our scheme with the network coding based approach as follows.



(a) Delay.

(b) Transmission rounds.

(c) Total number of transmissions.

Figure 9: The testbed results.

When using network coding, a node can encode and forward packets only when it has received and decoded the native packets. This prevents the out-of-order transmission because of the coding limitations, As a result, each node should encode and transmit the packets within one page and can start transmitting the next page only when the current page is decoded by its receivers. Although the slots are full of transmissions, the data information comes from only one page. Besides, the existing network coding schemes are often time consuming, especially in low-power resource constrained devices. Considering that a node can forward packets only when it decodes the current packets, the decoding delay directly adds to the overall dissemination delay.

ULTRA differs from the network coding based approach in two important ways. First, ULTRA transmits a full-slot sized *new* native data packets in each slot. The receivers are always receiving new data instead of redundancy. Second, ULTRA has almost no decoding overhead. Different from the existing network coding approaches, ULTRA uses simple XOR codes to fill up the slot when there are not enough native packets for a slot. The decoding is highly lightweight and effective for bursty packet errors. When under non-bursty scenarios, it is possible for ULTRA to switch to more general fountain codes. We will study the lightweight implementation and adoption of fountain codes in our future work.

5. Performance Evaluation and Analysis

In this section, we conduct testbed and simulation experiments to study the performance of ULTRA, in comparison with the state-of-the-art work CoCo [14]. After the testbed study, we tune the link quality in the TOSSIM simulation to study ULTRA's performance under different network conditions.

5.1. Methodology

We implement ULTRA in a testbed consisting of 16 TelosB nodes. Figure 8(a) shows the topology of the network and the link quality distributions. The radio power is set to the lowest level (-32.5dBm) to form a 4-hop network. Node 0 is the sink node, and disseminate a data object of 40KB to all network nodes. The cumulative distribution function (CDF) of pairwise link qualities is shown in Figure 8(b). We can see that there are about 30% poor links (packet reception rate < 0.6), 45% intermediate links (packet reception rate between 0.6 and 0.8) and 25% good links (packet reception rate > 0.8).

We choose the state-of-the-art work CoCo [14] for comparison. Both CoCo and ULTRA are structured protocols where each node works in the RX-TX-SLEEP cycle. The main difference of ULTRA from CoCo is the transmission mechanism.

- In CoCo, the data object is divided into pages and propagated in a page-by-page manner, i.e., packets in page n+1 can be transmitted only when all packets in page n have been received by the receiver. This will lead to that some TX slots are not fully utilized.
- In ULTRA, the packets to be transmitted in one slot is not restricted in one page. When there are not enough packets for a full slot transmission, redundant encoded packets will be added to fully utilize the TX slots.

For a fair comparison with CoCo, we set the same slot length and packet payload size with CoCo in

ULTRA, i.e., 23 bytes and the one-page-transmission time (transmitting 48 packets) with the same underlying structures.

Two key metrics are considered: the dissemination delay and the number of transmissions. We also study the performance in terms of the transmission rounds, in order to analyze the performance variations.

When a node has received all the 40KB data, it broadcasts a Report message to report the finish of the dissemination. We place a sniffer node near the network and overhear the report messages to record the dissemination performance. We compare the total packet transmissions and the dissemination completion time, where the completion time denotes the duration from the sink starts transmission to the moment *all* nodes receive the *entire* data object.

5.2. Testbed experiments

Comparison with CoCo.

Figure 9(a) compares ULTRA and CoCo in terms of the dissemination delay. We can see that compared to CoCo, ULTRA greatly reduces the dissemination delay by 34.8%. The reason is that in ULTRA, each slot is fully utilized, and the data object is pumped into the network as soon as possible.

Figure 9(b) compares ULTRA and CoCo in terms of the dissemination rounds. We can see that the number of the transmission rounds are greatly reduced. The reason is straightforward. Since there are more transmissions in each slot, the total number of slots is expected to decrease. This also explains why ULTRA can greatly reduce the dissemination delay.

Figure 9(c) compares ULTRA and CoCo in terms of the number of transmissions. This result is interesting as ULTRA actually experiences more transmissions than CoCo. Since both protocols eventually transmit the whole data object, ULTRA adds some encoded packets during the propagation, thus incurring more transmissions. However, we argue that more transmissions do not necessarily lead to significantly more energy consumption because the energy consumption in wireless embedded devices is dominated by the radio-on time [39, 40]. The energy consumption of idle listening and packet transmissions are similar.

ULTRA adds additional XORed packets to fully utilize the TX slot. Therefore, ULTRA achieves a higher number of transmissions than CoCo. It is worth noting considering the energy consumption is dominated by the radio-on time, the additional transmissions incur little extra energy consumption. On the other hand, the decrease in delay can save much energy consumption compared with CoCo.

Table 1: Comparison with Deluge

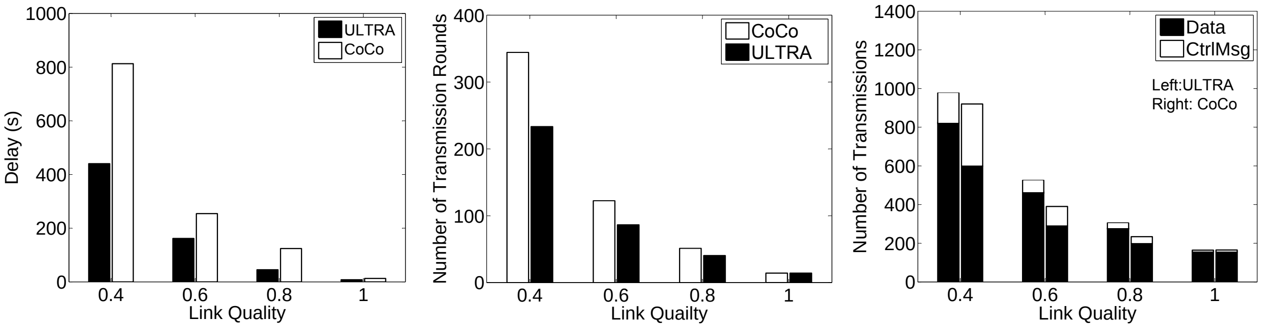
Improvements Density	transmission	delay
2	3.02%	11.32%
4	1.41%	20.75%
6	-4.43%	35.82%
8	-11.58%	47.69%

Comparison with Deluge.

We further compare the transmission and delay performance of ULTRA and a typical unstructured dissemination protocol Deluge [18], in terms of transmission and delay performance. Table 1 shows the evaluation results. The network density denotes the average number of neighboring nodes for each node. The transmission count and delay denote the average transmission count and delay during the dissemination process. We can see that, 1) ULTRA achieves better delay performance than Deluge, especially in dense networks. The reason is that Deluge requires negotiation (three-way handshake) for each page transmission, while ULTRA can transmit packets in an out-of-order manner. Therefore, ULTRA is expected to propagate data pages faster than Deluge. Specifically, when in dense networks, the negotiation will incur considerable delay and transmission overhead, which leads to more performance improvement of ULTRA than Deluge; 2) ULTRA transmits similar amount of packets compared with Deluge. The reason is that although ULTRA is expected to transmit fewer packets, it actively adds encoded packets in each batch of transmissions.

5.3. Simulation results

We take a step forward to investigate how ULTRA performs under various network conditions. We tune the link quality and study in which conditions ULTRA is more/less superior than the existing approaches. Figure 10(a) compares ULTRA and CoCo in terms of the dissemination delay. We can see that when the links are perfect, ULTRA slightly outperforms CoCo. The reason is that there are no packet losses under such situation, and CoCo also transmits full slots. However, since CoCo transmits ADVs in each slot, its performance is worse than ULTRA. It is worth noting that ULTRA transmits extra slot sequences in each packet, which adds the transmission overhead. However, these bytes are embedded in the packet headers and no extra channel back-offs will be required. Thus the slot sequences do not add the overall dissemination delay.



(a) Delay.

(b) Transmission rounds.

(c) Number of transmissions.

Figure 10: The simulation results.

When links become worse, ULTRA performs better than CoCo because the utilization of each slot will decrease in CoCo when the link quality becomes worse. As a result, the propagation speed of CoCo becomes more and more slow as links become worse. Differently, ULTRA sill fully utilizes the slots, and pumps new data into the network as soon as possible. Its performance degradation is much less than that of CoCo.

Figure 10(b) compares ULTRA and CoCo in terms of the number of transmission rounds. We can get the observations similar to the dissemination delay. We can also infer that the dissemination delay is highly affected by the number of transmission rounds. The out-of-order transmission can effectively reduce the transmission rounds, as compared to the traditional page-by-page transmission.

Figure 10(c) compares ULTRA and CoCo in terms of the number of transmissions. We can see that when the link quality becomes worse, the number of transmissions of ULTRA increases. The reason is that when links become worse, more and more encoded packets will be made up to compose a full slot transmission. Since there will be no transmissions when there are not enough native packets in CoCo, these encoded packets in ULTRA directly add up to the overall number of transmissions. Therefore, although the dissemination delay is reduced, the transmission count increase. Recall that the number of slots dominate the energy consumptions, since the energy consumptions of idle listening and packet transmission are similar when the radio is in "ON" state. And the radio "ON" time is determined by the number of transmission rounds, with which ULTRA performs better than the existing protocols.

6. Conclusion

In this paper, we investigate the problem of structured bulk data dissemination in WENS and propose a f**ul**l-slot reliable **tra**nsmission mechanism (*ULTRA*) that can fully exploit the benefits of the established structures. Compared to the three-way handshake mechanism, ULTRA requires only one round negotiation and can fully exploit each transmission slot. Besides, the slot length is optimized for fast data dissemination. We conduct both testbed and simulation experiments to study the performance of ULTRA. The results show that, although ULTRA may transmit more packets than the state-of-the-art protocol, it greatly reduces the completion time, which is the key metric for bulk data dissemination.

Since another potential of the structured protocol is for multi-channel communications, our future directions lie in optimizing the slot length and slot coordination for multi-channel enabled network systems.

Acknowledgement

This work is supported by the National Science Foundation of China under Grant No. 61602095 and the Fundamental Research Funds for the Central Universities under Grant No. ZYGX2016KYQD098.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless Sensor Networks: A Survey, Computer Networks 38 (4) (2002) 393–422.

[2] S. Sridhar, A. Hahn, M. Govindarasu, Cyber–physical system security for the electric power grid, Proceedings of the IEEE 100 (1) (2012) 210–224.

[3] E. Moreno, T. Webber, C. Marcon, F. Moraes, N. Calazans, Monoc: A monitored network on chip with path adaptation mechanism, Journal of Systems Architecture 60 (10) (2014) 783–795.

[4] Y. Gao, W. Dong, K. Guo, X. Liu, Y. Chen, X. Liu, J. Bu, C. Chen, Mosaic: A Low-Cost Mobile Sensing System for Urban Air Quality Monitoring, in: Proceedings of IEEE INFOCOM, 2016.

[5] Y. Liu, J. Niu, J. Ma, W. Wang, File downloading oriented roadside units deployment for vehicular networks, Journal of Systems Architecture 59 (10) (2013) 938–946.

[6] T. Torfs, T. Sterken, S. Brebels, J. Santana, R. van den Hoven, V. Spiering, N. Bertsch, D. Trapani, D. Zonta, Low power wireless sensor network for building monitoring, IEEE Sensors Journal 13 (3) (2013) 909–915.

[7] O. K. Sahingoz, Large scale wireless sensor networks with multi-level dynamic key management scheme, Journal of Systems Architecture 59 (9) (2013) 801–807.

[8] D. He, C. Chen, S. Chan, J. Bu, Sdrp: A secure and distributed reprogramming protocol for wireless sensor networks, IEEE Transactions on Industrial Electronics 59 (11) (2012) 4155–4163.

[9] C. Miller, C. Poellabauer, Reliable and efficient reprogramming in sensor networks, ACM Transactions on Sensor Networks (TOSN) 7 (1) (2010) 6.

[10] W. Dong, Y. Liu, C. Chen, J. Bu, C. Huang, Z. Zhao, R2: Incremental reprogramming using relocatable code in networked embedded systems, IEEE Transactions on Computers 62 (9) (2013) 1837–1849.

[11] C. Salvadori, M. Petracca, S. Madeo, S. Bocchino, P. Pagano, Video streaming applications in wireless camera networks: A change detection based approach targeted to 6lowpan, Journal of Systems Architecture 59 (10) (2013) 859–869.

[12] V. Naik, A. Arora, P. Sinha, H. Zhang, Sprinkler: A reliable and energy efficient data dissemination service for extreme scale wireless networks of embedded devices, IEEE Transactions on Mobile Computing 6 (7) (2007) 777–789.

[13] L. Huang, S. Setia, CORD: Energy-efficient Reliable Bulk Data Dissemination in Sensor Networks, in: Proceedings of IEEE INFOCOM, 2008, pp. 1–9.

[14] Z. Zhao, D. W., J. Bu, T. Gu, C. Chen, Exploiting link correlation for core-based dissemination in wireless sensor networks, in: Proceedings of IEEE SECON, 2014, pp. 372–380.

[15] T. Datasheet, Crossbow inc.

[16] P. Levis, N. Lee, M. Welsh, D. Culler, TOSSIM: Accurate and scalable simulation of entire TinyOS applications, in: Proceedings of ACM SenSys, 2003.

[17] M. Nor, M. Sulaima, M. Baharom, A. T. Ibrahim, F. Idris, Performance study of simple flooding and dominant pruning protocol in wireless sensor network via tossim, Journal of Engineering Technology 2 (2014) 75–88.

[18] J. W. Hui, D. Culler, The Dynamic Behavior of A Data Dissemination Protocol for Network Programming At Scale, in: Proceedings of ACM SenSys, 2004, pp. 81–94.

[19] S. Kulkarni, L. Wang, Energy-efficient multihop reprogramming for sensor networks, ACM Transactions on Sensor Networks (TOSN) 5 (2) (2009) 16.

[20] W. Dong, Y. Liu, Z. Zhao, X. Liu, C. Chen, J. Bu, Link quality aware code dissemination in wireless sensor networks, IEEE Transactions on Parallel & Distributed Systems (1) (2013) 1.

[21] P. A. Levis, N. Patel, D. Culler, S. Shenker, Trickle: A self regulating algorithm for code propagation and maintenance in wireless sensor networks, Computer Science Division, University of California, 2003.

[22] Y. Liu, G. Zhou, J. Zhao, G. Dai, X. Li, M. Gu, H. Ma, L. Mo, Y. He, J. Wang, et al., Long-term large-scale sensing in the forest: recent advances and future directions of greenorbs, Frontiers of Computer Science in China 4 (3) (2010) 334–338.

[23] Y. Cao, N. Wang, Z. Sun, H. Cruickshank, A reliable and efficient encounter-based routing framework for delay/disruption tolerant networks, IEEE Sensors Journal 15 (7) (2015) 4004–4018.

[24] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, et al., TinyOS: An Operating System for Sensor Networks, Ambient intelligence 35 (2005) 115–148.

[25] W. Dong, X. Liu, C. Chen, Y. He, G. Chen, Y. Liu, J. Bu, DPLC: Dynamic Packet Length Control in Wireless Sensor Networks, in: Proc. of IEEE INFOCOM, 2010.

[26] A. Hagedorn, D. Starobinski, A. Trachtenberg, Rateless Deluge: Over-the-air Programming of Wireless Sensor Networks Using Random Linear Codes, in: Proceedings of ACM/IEEE IPSN, 2008, pp. 457–466.

[27] M. Rossi, N. Bui, G. Zanca, L. Stabellini, R. Crepaldi, M. Zorzi, SYNAPSE++: Code Dissemination in Wireless Sensor Networks Using Fountain Codes, IEEE Transactions on Mobile Computing 9 (12) (2010) 1749–1765.

[28] I. Hou, Y. Tsai, T. Abdelzaher, I. Gupta, AdapCode: Adaptive Network Coding for Code Updates in Wireless Sensor Networks, in: Proc. of INFOCOM, 2008.

[29] Z. Zhao, J. Bu, W. Dong, T. Gu, X. Xu, Coco+: Exploiting correlated core for energy efficient dissemination in wireless sensor networks, Ad Hoc Networks 37 (2016) 404–417.

[30] Z. Zhao, W. Dong, G. Guan, J. Bu, T. Gu, C. Chen, Modeling link correlation in low-power wireless networks, in: 2015 IEEE Conference on Computer Communications (INFOCOM), IEEE, 2015, pp. 990–998.

[31] S. Wang, G. Tan, Y. Liu, H. Jiang, T. He, Coding opportunity aware backbone metrics for broadcast in wireless networks, IEEE Transactions on Parallel and Distributed Systems 25 (8) (2014) 1999–2009.

[32] H. Du, W. Wu, Q. Ye, D. Li, W. Lee, X. Xu, Cds-based virtual backbone construction with guaranteed routing cost in wireless sensor networks, IEEE Transactions on Parallel and Distributed Systems 24 (4) (2013) 652–661.

[33] X. Kui, Y. Sheng, H. Du, J. Liang, Constructing a cds-based network backbone for data collection in wireless sensor networks, International Journal of Distributed Sensor Networks 2013.

[34] W. Du, J. C. Liando, H. Zhang, M. Li, When pipelines meet fountain: Fast data dissemination in wireless sensor networks, in: Proceedings of ACM SenSys, 2015, pp. 365–378.

[35] Y. Cui, L. Wang, X. Wang, H. Wang, Y. Wang, Fmtcp: A fountain code-based multipath transmission control protocol, IEEE/ACM Transactions on Networking 23 (2) (2015) 465–478.

[36] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, P. Levis, The β-factor: measuring wireless link burstiness, in: Proceedings of ACM SenSys, 2008, pp. 29–42.

[37] D. Gómez, R. Agüero, M. García-Arranz, L. Muñoz, On the use of hidden markov processes and auto-regressive filters to incorporate indoor bursty wireless channels into network simulation platforms, Wireless Networks 21 (7) (2015) 2137–2154.

[38] E. Ghadimi, O. Landsiedel, P. Soldati, S. Duquennoy, M. Johansson, Opportunistic routing in low duty-cycle wireless sensor networks, ACM Transactions on Sensor Networks 10 (4) (2014) 67.

[39] N. Pantazis, S. A. Nikolidakis, D. D. Vergados, Energy-efficient routing protocols in wireless sensor networks: A survey, IEEE Communications Surveys & Tutorials 15 (2) (2013) 551–591.

[40] S. Guo, L. He, Y. Gu, B. Jiang, T. He, Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links, IEEE Transactions on Computers 63 (11) (2014) 2787–2802.