ZInC: Index-Coding for Many-to-One Communications in ZigBee Sensor Networks

Ismail Salhi, Erwan Livolant, Yacine Ghamri-Doudane, Stéphane Lohier

ZInC: Index-Coding for Many-to-One Communications in ZigBee Sensor Networks

Ismail Salhi, Erwan Livolant, Yacine Ghamri-Doudane, Stéphane Lohier
Université Paris-Est – Laboratoire d'Informatique Gaspard Monge (LIGM)
{isalhi, livolant, ghamri, lohier}@univ-mlv.fr

***Abstract*—The main goal in wireless sensor networking remains the reduction of the network lifecycle and the enhancement of its reliability, keeping decent performances in terms of throughput and latency. Given the increasing interest of the research community on wireless network coding (NC), we think such challenges can be tackled using its innovative concepts, especially in the case of many-to-one communications where network coding has shown promising theoretical results. Yet, without a thoughtful adaptation to WSNs, the benefits of NC for sensor networking prove to be too "greedy" and impractical. In this paper, we propose index-coding, a simple and effective packet coding scheme that enhances significantly many-to-one communications in ZigBee sensor networks. Index-coding uses smart bit-shifting operations in order to encode short messages from a set of sensors to a sink using fewer transmissions. Our implementation in a real ZigBee testbed shows substantial enhancement of network performances and resiliency.**

Keywords- Wireless Sensor Networks; ZigBee; Index Coding; IEEE 802.15.4; Many-to-One communications;

I. INTRODUCTION

The main objective of Wireless Sensor Networks (WSN) is to collect data from different sensors to one or multiple destinations. Except in multimedia sensing, the size of the reported data in such low-power/low-rate networks normally does not surpass few bytes per node, and packets are delivered using multi-hops communications that imply a substantial packet overhead. Reducing this overhead and the number of transmissions is one of the most important challenges in WSNs, as it allows alleviating the complexity of the node's communication stack and reducing its energy consumption, thereby prolonging the network lifecycle.

The network coding (NC) theory [1] provides a set of coding systems that ensure a notable reduction of the number transmissions in wireless networks. An interesting way to tackle the inherent issues of WSNs is thus to make use of the concepts of this theory to improve existing communication protocols. Indeed, current advances in the applications of wireless NC have attracted the attention of scientists and system designers in computer networks to cope with the sub-optimality of *store-and-forward* schemes. However, the application of NC to low-power/low-rate networks is not trivial. Indeed, an *as is* utilization of current coding methods such as [10]-[12] in WSNs would be unproductive, if not impossible, since it involves a notable overhead and a negative impact on the memory intake of the communication stack[1].

This paper addresses the problem of many-to-one (M2O) or incast communications in ZigBee[2] [16] networks and the underlying Medium Access Control layer, the IEEE 802.15.4 [14]. We present a lightweight yet effective packet coding scheme, so called *ZigBee index-coding* or ZinC, which improves many-to-one communications in terms of throughput, delay and energy efficiency. In M2O communications, several, probably correlated source nodes send their sensed information to a single receiver or sink. In this context, and considering such pattern, ZInC resides in a shim encoding layer that combines incoming packets from different sources into one outgoing packet following an index calculation based on the ZigBee hierarchical addressing. ZInC contrasts from other NC schemes because it is explicitly designed to run on constrained devices with very low memory and finite energy level. The principle behind ZInC is to convey the address of source nodes through an index they process using their own ZigBee addresses. Intermediate node can then XOR incoming packets, and generate a representation of the packet produced by each source node. This allows recovering the content of each combined packet and its source address at the sink.

To evaluate our solution, we measure in a real WSN testbed the impact of ZInC on different performance metrics central to sensor networks such as the number of transmissions, the end-to-end delay and the energy intake. We show how ZigBee routers can deliver the same amount of data from a set of sources towards one destination with fewer transmissions, thereby improving throughput and latency with minimum overhead. We demonstrate how ZInC can operate within very constrained devices, and reveal the substantial benefits that index-coding may provide under realistic environments, namely, very low RAM memory, limited buffer management efficiency and unreliable channel conditions. Furthermore, we estimate in this experiential work that using ZInC can imply an energy consumption cutback up to 50% compared to classic ZigBee routing.

The rest of the paper is organized as follows: Section II discusses the related work in wireless network coding and the prior efforts in adapting its concepts to wireless sensor networking. Section III defines the context and the system's characteristics. In Section IV we detail our coding scheme and its main features, while Section V shows the practical benefits of ZInC via the detail of our implementation of index-coding in a real ZigBee sensor network testbed. The paper concludes with a brief discussion in Sections VI.

[1] If we consider the hardware used in our experiments, the size of the RAM is 10^3-10^6 % smaller than in PCs or new generation cellphones.

[2] ZigBee is the most widespread technology for sensor networking and wireless personal area networks (WPAN)

II. BACKGROUND & RELATED WORK

The utilization of network coding in wireless networks has shown notable results in terms of bandwidth utilization, reliability and latency [5]. The original work of Ahlswede *et. al.* [1] and all the follow-up works during the last 10 years have turned practical NC from a complex solution devoted to multicast networks [7], to a generic and effective scheme that mixes flows of packets to improve the performance of multi-hop wireless networks [6]. Today, practical NC is a whole research field by itself, and is studied for many applications such as satellite communications, large scale content distribution, delay tolerant networks or network security.

The first practical application of wireless network coding, namely COPE [11], has sparked the interest of researchers on using one-hop coding to improve wireless communications. Ever since, several protocols such as, CodeCast [6] or I2NC [12] were developed and assessed on real testbeds using one-hop XOR coding for unicast, anycast, multicast and dissemination over wireless networks [17]. Part of these efforts has been interested in the analysis of NC in terms of computational costs and coding efficiency. Other works have considered the coding reliability problem in lossy environments. Yet, few of these initiatives have been dedicated to the assessment of NC over realistic low-power/low-rate networks such as WSNs or WPANs. In fact, while NC has been analytically adapted to operate on WSNs in the literature [2]-[3] and optimal solutions approached via heuristic-based schemes, none of the proposed solutions has provided conclusive answers to the practical problems that still prevent NC from implementable solutions (finite energy level, small buffer size, limited computational resources, etc.).

In [2], authors demonstrate that in a many-to-one network, the maximum available throughput can be achieved by finding a disjoint path from each source to the destination. Finding such solutions is hard, particularly under a dynamic topology and when the network is error-prone as it is the case in WSNs. In their pioneering paper [7] Ho *et al.* induced briefly the potential of NC for multicast and a wider family of problems including the M2O communication in sensor networks. Since then, different versions of existing NC techniques were proposed to improve M2O communications. Still, since the performance analysis of most solutions is strictly theoretical, practical matters inherent to low-power/low rate networks like memory usage, real-time energy consumption or reliability are repeatedly untreated.

Authors of [15] propose a solution that significantly enhances the prior efforts on using NC in constrained networks. However, its performance evaluation emphases only the achievable throughput and the reliability in case of node failure, which, we think, makes their solution incomplete for practical application as energy consumption and deployment issues are not treated. In [13], the authors bring together the outcomes of the implementation of Partial Network Coding [9] (PNC) in order to remove outdated data without necessarily decoding any packet. Though PNC shows good performances under a real environment and proves that partial NC can be applied to low-memory and low-power devices. Its target remains application-specific. Indeed, PNC addresses the problem of efficiently discarding obsolete information in code-and-forward sensor networks which is one particular use case. Furthermore, the authors do not consider in their empirical analysis metrics central to WSNs like the distribution of the energy consumption, the latency or the number of transmissions. Similarly, authors of [8] define AdapCode an interesting coding technique for data dissemination in WSNs where packets on intermediate nodes are linearly combined and decoded by Gaussian elimination. Although AdapCode provides promising results in terms of number of transmissions and reliability, it considers only dissemination-based networks, which is not the dominant traffic pattern in sensor networks. Besides, AdapCode does not provide any analysis on the energy consumption cost of NC.

In the most comprehensive work on energy-aware NC [3], the authors propose COPR (Coding with Opportunistic Reception) a coding scheme that ensures decoding of each coded packet at its next hop. COPR is based on a backpressure algorithm that uses the reception of both coded and non-coded packets. It is important to know that COPR has not been intended for WSNs. It was designed to reduce the energy consumption of COPE in multi-hop wireless networks. Indeed, COPR was tested on IEEE 802.11b MAC layer which cannot be used in low-power/low-memory devices. Besides, while COPR ensures an energy cutback of almost 25% compared to XOR coding, it requires, on the other hand, the management of several queues for each unicast flow a node is involved in. This induces a large memory overhead that is incompatible with the limitations of state-of-the-art sensing devices. Moreover, in order to COPR to operate, the MAC addresses of all the one-hop destinations have to be inserted in the coded data frame. This imply $N \times 6\, bytes$ per frame, with N the number of neighbors. Such overhead is small in an IEEE 802.11 data frame where the payload size equals 2312 bytes, but if we consider sensor networks MAC protocols such as IEEE 802.15.4 where the data frame does not exceed 118 bytes, the per packet coding overhead would be excessive.

In this paper, we define ZInC, a coding scheme that is, unlike other prevalent network coding solution, not based on packet eavesdropping, as we claim it implies a considerable energy consumption surplus and a significant memory overhead due to multiple buffers maintenance. ZInC enhances many-to-one communications in terms of latency, reliability and achievable throughput and ensures less power-greedy end-to-end data delivery. It also differs from other coding schemes because it involves minor computational and network overheads. Thanks to the implementation of ZInC in a testbed, we show that it provides better performances and reliability than classic store-and-forward routing and an equitable energy consumption distribution across the routes toward the sink.

III. DESIGN CONSIDERATIONS

A. Motivation and Key Idea

The idea behind ZInC has emerged from our observation that the payload's size in WSNs is particularly small compared to other wireless technologies. For example, the payload of an IEEE 802.15.4 data frame is 20 times smaller than the payload of an IEEE 802.11 one. And the size of the sensed information is often considerably inferior to the available payload used to transmit them. Additionally, the required NWK and MAC

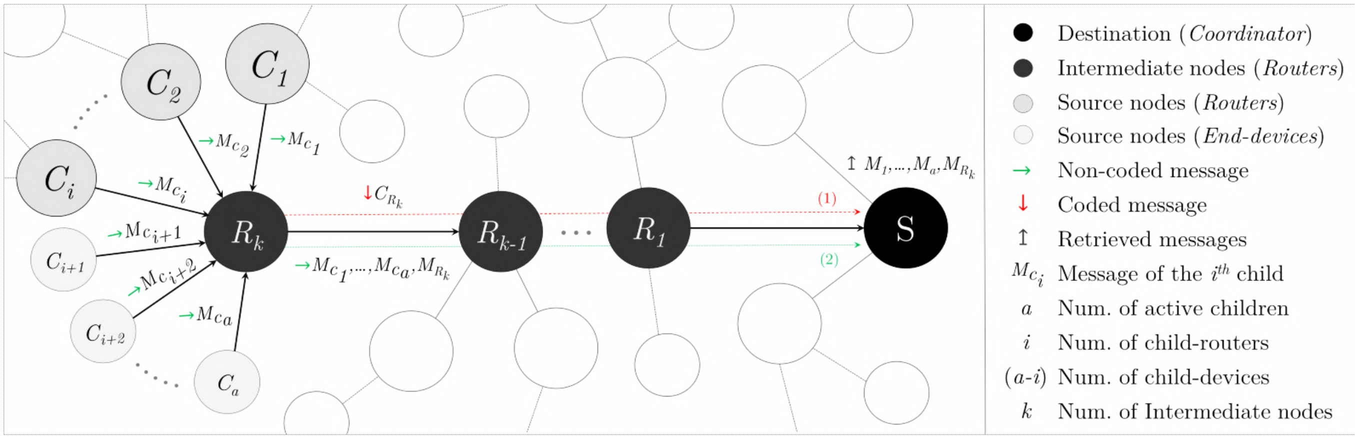

Figure 1 – Nodes $C_1,\ldots,C_a$ send their data to S via a k hops route either with ZigBee routing (in green) or via ZInC coding (in red)

headers are bigger than the effective payload itself. Indeed, in scalar WSNs, sources transmit data such as temperature, pressure or humidity. These can be coded in one message of few bits or bytes. For instance, to send the air temperature value, which we assume is a short integer, only 1 byte is sufficient. Yet, in classic WSN routing protocols like ZigBee or 6lowPan, such value would represent only 0,8% of the payload and the NWK and MAC headers represent between 170% and 330% of its size. This suggests that classic NC schemes such as [11]-[12] where the size of headers is function of the number of encoded packet must be avoided for overhead minimization. Moreover, these intrinsic characteristics to WSNs led us to think about a solution that takes advantage of the unused payload to improve the bandwidth utilization and the energy consumption. Indeed, this gap between the effective data and the carriage capacity of each packet makes it sub-optimal to transmit data in a store-and-forward manner.

B. System Assumptions

We consider a ZigBee scalar sensor network with the following configuration parameters [16] (C_m, R_m, L_m) with $C_m$being the maximum number of children, $R_m$the maximum number of children-routers and $L_m$the maximum depth of the network. In ZigBee, M2O communications are performed using hierarchical- or mesh-based routing. Either way, we assume in this work that M2O communications occur between C_a active nodes ($C_a \le C_m$) and the sink via their parent router R_k through a route of k hops (cf. Figure 1). We denote by $R = \{R_1, \ldots, R_k\}$ the set of intermediate routers that constitute this route. Note that among the C_a active nodes, C_i is the number of active ZigBee routers and $(C_a - C_i)$ the number of active end-devices (with $0 \le C_a - C_i \le C_m - R_m$). We assume that sensed data of an arbitrary node c_j are represented in a vector I_{c_j} of size N with $I_{c_j} = [m_{c_j,1}, m_{c_j,2}, \ldots, m_{c_j,N}]$.

In this context, we propose ZInC. A coding scheme that exploits the unused payload of forwarded packets to encode data from other source nodes in a decentralized way. Indeed, with index-coding, source nodes apply a bit-shift operation using their hierarchical ZigBee addresses. This shift allows intermediate nodes to perform a simple linear operation on the payloads of incoming packets, and transmit the whole data segment in one transmission instead of forwarding each packet independently. The sink retrieves then each source payload and its associated node address using reverse-index-coding. Note that ZInC is interesting only in the case of scalar sensor networking where the effective data are small enough to include them within the payload of other packets.

IV. ZINC OVERVIEW

The general principle behind ZInC is illustrated in Figure 1. Instead of forwarding incoming packets individually, the first intermediate router along the route to the sink bufferizes them during a predefined period of time. Then using index-coding, it combines their respective payloads in a single outgoing coded packet. This latter will then be forwarded to the destination for decoding. The aim of such coding technique is to lower the number of transmissions, reduce the end-to-end latency and improve the lifetime of routers involved in M2O communication by expanding their sleeping period.

A. Information Vector Transformation

To operate ZInC, active nodes c_j ($\forall j = [1, a]$) apply a bit-shifting operation to their respective information vector I_{c_j} before any transmission. This is done by processing an offset using the node's own ZigBee address as defined in equation 1. Where $O_i(I_{cj})$ is the function that applies a i-bits offset to I_{cj}.

$$\forall\, c_j \in C_a,\ I_{c_j} = [m_{c_j,1}, m_{c_j,2}, \ldots, m_{c_j,N}] \Rightarrow O_i(I_{cj}) = M_{c_j} \quad (1)$$

$$\text{With } O_i(I_{cj}) = M_{c_j} = \begin{array}{c} \begin{array}{cccc} & & & i \end{array} \\ \begin{bmatrix} 0 & 0 & \cdots & m_{c_j,1} \\ 0 & 0 & \cdots & m_{c_j,2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & m_{c_j,N} \end{bmatrix} \end{array} \text{ and } i = idx(c_j)$$

B. Local Index Calculation

The index i (with $i = idx(c_j)$) used to know by how many bits I_{cj} is shifted has to be unique for each child node to avoid overlapping data when coding at intermediate routers. The value of i is processed using the node's ZigBee address $A(c_j)$. Equation 2 shows exactly how i is processed.

$$idx(c_j) = \begin{cases} \dfrac{A(c_j) - A(\dot{c}_j) - 1 + C_{skip}(\dot{c}_j)}{C_{skip}(\dot{c}_j)} & c_j \text{ is a router} \\ A(c_j) - A(\dot{c}_j) - [1 - C_{skip}(\dot{c}_j)] . R_m & otherwise \end{cases} \quad (2)$$

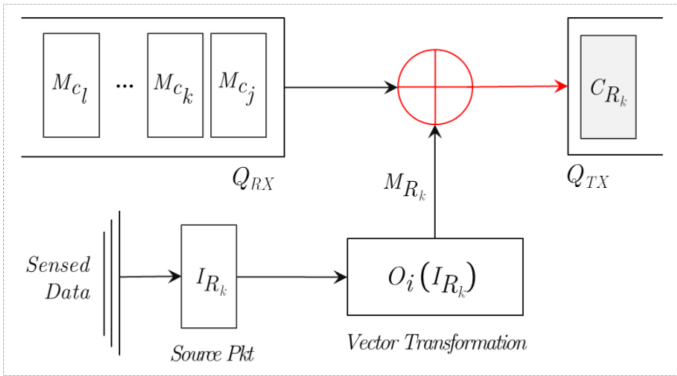

Figure 2 – Index-coding phases at an intermediate ZigBee router R_k

Knowing that: $C_{skip}(x) = \frac{C_m.R_m^{Lm-d(x)-1}+R_m-C_m-1}{R_m-1}$

With $\dot{c_j}$ being the designation of the direct parent of c_j in the network hierarchy. And C_{skip} a configuration parameter in the ZigBee Tree Addressing algorithm [16]. Its value is processed by each node to reserve address space for their children. Note that nodes process *idx* locally and independently, without any centralized knowledge of the network or any control messages.

C. Packet Encoding

In order for the sink to recover all source messages, distinct coded packets must be generated by each linear combination at intermediate nodes. Indeed, as depicted in Figure 2, when a router R_k receives packets from a subset of active source nodes C_i ($\forall i,\ 1 \le i \le a$), it first queues them in its reception buffer Q_{RX}. Then at a specific period of time, it linearly combines their pre-shifted payloads M_{C_i} with its own one, into one packet C_{R_k} (cf. Equation 3), and forwards it to the next upstream hop. Obviously, in the case where R_k is not associated to any source, it simply transmits its packet M_{R_k}.

$$C_{R_k} = \overset{j \in Q_{RX}}{XOR}(M_j) \oplus M_{R_k} \tag{3}$$

As formerly stated, ZInC only addresses scalar WSNs where small amounts of data are reported. Figure 3 shows that ZInC can transmit up to 5 times less amount of data compared to ZigBee, specifically when the size of the sensed information is low (<4 bytes in this case[3]). However as the size of data reports increase, the benefit of index-coding recedes because the pre-shifted source packets transmitted to the router become too large, thus, increasing the network overhead.

D. Packet Decoding

Whenever the destination receives an encoded packet, it applies a *reverse* index-coding operation. I.e., the sink extracts from the incoming packets C_{R_k} (cf. Equation 4) the original vectors $v_i = [m_{c_i,1}, m_{c_i,2}, \dots, m_{c_i,N}]$. Then, using the router source address field in the coded packet and each vector index i, the sink processes the addresses of each original payload using Equation 5, and retrieves the identities of all the sources nodes involved in the communication.

[3] We used in this numerical estimation a canonical topology in which 10 child-nodes report their sensed data to the sink via a 8-hops route.

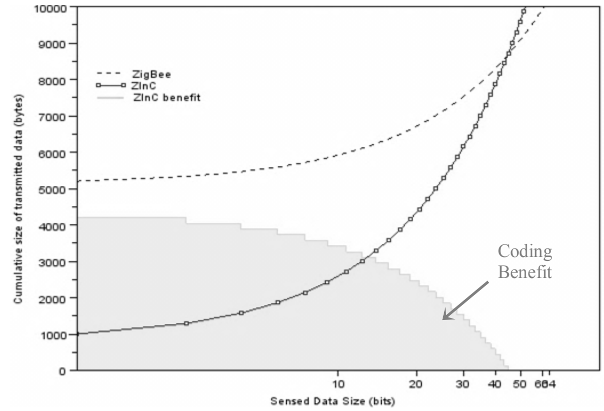

Figure 3 – Index-coding phases at an intermediate ZigBee router R_k

$$C_{R_k} = \begin{bmatrix} m_{c_1,1} & \cdots & m_{c_i,1} & \cdots & m_{c_a,1} & m_{R_k,1} \\ m_{c_1,2} & \cdots & m_{c_i,2} & \cdots & m_{c_a,2} & m_{R_k,2} \\ \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ m_{1,N} & \cdots & m_{c_i,N} & \cdots & m_{c_a,N} & m_{R_k,N} \end{bmatrix} \tag{4}$$

$$A(c_j) = \begin{cases} A(\dot{c_j}) + C_{skip}(\dot{c_j}).R_m + Idx(c_j) & Idx(c_j) \le R_m \\ A(\dot{c_j}) + 1 + C_{skip}(\dot{c_j}).(Idx(c_j) - 1) & Idx(c_j) > R_m \end{cases} \tag{5}$$

The careful reader might notice that a special code needs to be defined to differentiate non-active nodes from active ones in coded packets. Indeed, in the case of an idle source, we use the vector [0 0] as special code exclusively used when nodes do not transmit any data. This will prevent the destination from retrieving ambiguous information.

E. Canonical Example

For the sake of clarity, consider the canonical example depicted in Figure 4. In this part of the network, 4 of the 5 source nodes $\{R_1, R_2, d_1, d_2\}$ transmit 2-bits messages $M_{i,j}$ to their parent router R_k. In this case, ZInC selects for each source node c_i its corresponding shifted vector using $O_i(I_{cj})$. For example, it assigns columns 1 to 2 to the node R_1. Node R_2 is assigned in the same way columns 2 to 4 and so on.

$$M_{R1} = [\, m_{R1,2}\ \ m_{R1,2}\ \ 0\ \cdots\ 0\,], M_{R2} = [\,0\ \ 0\ \ m_{R2,2}\ \ m_{R2,2}\ \ 0\ \cdots\ 0\,], \dots$$

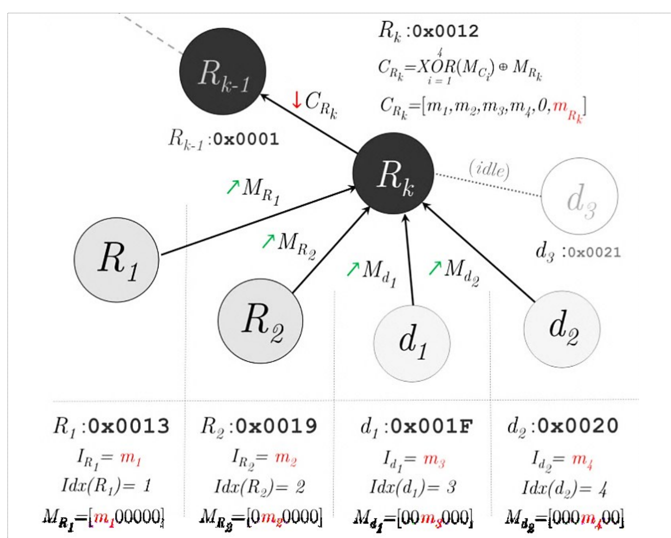

Figure 4 – Index-Coding Illustration

V. Performance Evaluation

In this section we show the performance of ZInC compared to classic ZigBee routing in a realistic scenario. We argue that simulation-based or analytical validation techniques are not appropriate for assessing architectures as technology-dependent and constrained as WSNs. Indeed, such impractical methods often overlook central properties like memory constraints, energy consumption and computational costs. Thus, to demonstrate that our solution is designed to run on real sensors using state-of-the-art WSNs communication protocols, we set up an experimental testbed (cf. Figure 5) that emulates practical many-to-one communications in a 802.15.4 environment.

We deploy 13 ZigBee sensors among which 5 are Memsic's MicaZ end-devices and 8 are TelosB routers. 9 sensors (5 end-devices and 4 routers) are potential source nodes that transmit sensed data at various constant bitrates (0.5 to 4 packets per-second). And 4 routers establish the route that forwards data flows to the sink. The parent of active sensors performs index-coding and forwards coded packets upstream. To mitigate randomness, all results are an average of 20 independent experiments. To the best of our knowledge, ZInC is the first practical coding implementation that deals with the incast (M2O) problem in WSNs and that provides conclusive answers in terms of both performances and WSN requirements.

A. Number of Transmissions

In a classic ZigBee network, intermediate nodes simply forward the messages transmitted by source nodes till they reach their destination. To transmit fewer packets, knowing that there is an under-exploitation of the bandwidth, ZInC allows each intermediate router to combine incoming messages from active sensors into one coded packet. Such coding would significantly reduce the number of transmissions as shown in Figure 6. Indeed, we first compare the average number of transmitted packets (including retransmissions) for each M2O communication using ZigBee forwarding or ZInC. Index-coding outperforms ZigBee for a number of active nodes greater than 1 which is plausible since we address incast communications and not unicast. While the number of transmissions in ZigBee is linear with respect to the number of active sensors, the slope for ZInC is much smaller. Obviously the number of transmissions with ZInC is further reduced as the number of active sensors increase, because coded packets embed a larger number of packets.

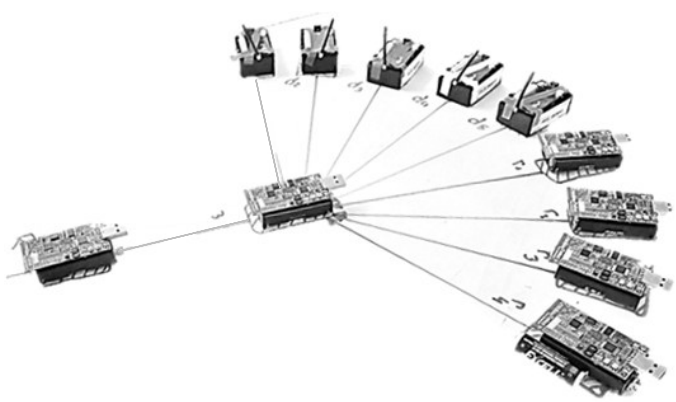

Figure 5 – ZInC M2O Experimental Testbed

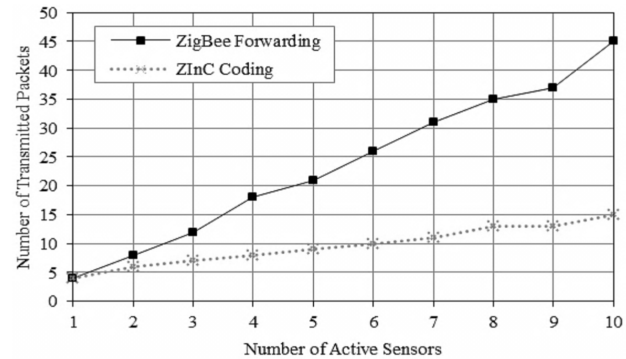

Figure 6 – Average Number of Transmissions vs. Number of Sources

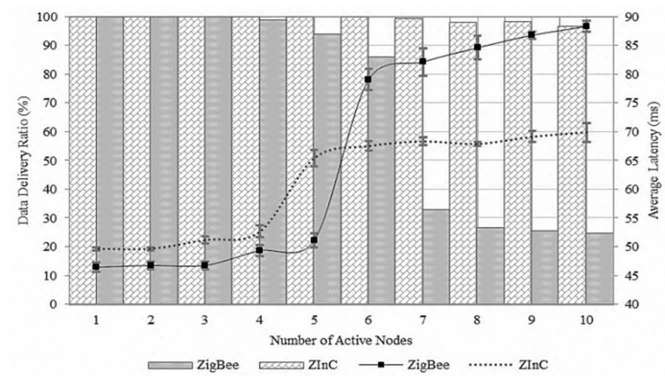

Figure 7 –Latency vs. Number of Sources vs. Data Delivery Ratio

B. End-to-End Latency & Reliabiltiy

An important metric that WSNs rely on is the per-packet end-to-end latency. In sensor networks, the delay is strongly correlated to the loss rate endured by each link of the network. This is due to the use of the CSMA/CA carrier sensing scheme. Thanks to ZInC we can reduce the impact of loss rate on the latency, as nodes in this case transmit considerably less packets and thus reduce the probability of packet loss due to collisions.

As plotted in Figure 7, the end-to-end delay using ZInC is slightly higher than with ZigBee when $C_a \leq 5$, which is due to the bufferization period needed to encode packet. However, when $C_a > 5$, the end-to-end latency provided by ZInC is almost 45ms lower than with ZigBee. Furthermore, the data delivery ratio show that ZInC delivers 3 times the amount of packets delivered by ZigBee as the number of active node increase and collisions occur more frequently.

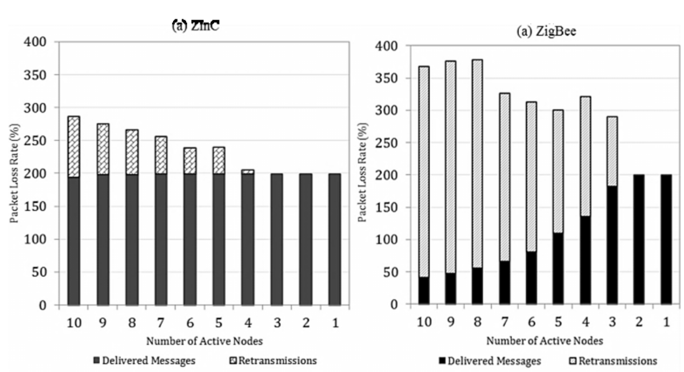

Figure 8 – Number of Sources vs. Number of Delivered Packets

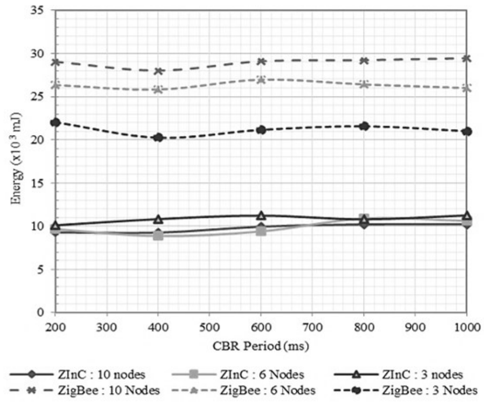

Figure 9 – Energy Consumption vs. CBR Bitrate

To assess the reliability of ZInC and ZigBee as the number of source nodes increases, we generate a CBR traffic that transmit 200 packets from all source nodes and process the number of received packets at the sink. Figure 8 depicts the results of this experiment and shows the number of delivered packets and the number of retransmissions. It is clear that ZInC outperforms ZigBee in terms of reliability. While ZigBee routing suffers rapidly from the increase of active sensors, ZInC ensures positive results and delivers almost 100% of the 200 packets.

C. Energy Consumption

The most important metric WSNs depend on is the energy consumption. Contrary to what one might think, the energy efficiency is not directly correlated with the number of transmissions, but rather with the time during which the communication module of the node is enabled. Indeed, [4] asserts that nodes that listen to the radio channel consume roughly the same amount of energy as those which constantly transmit packets. Thereby, to improve the energy efficiency of nodes, it is preferable to extend the radio modules inactivity time rather than seeking to reduce transmissions. ZInC allows to the routers involved in M2O communications to disable their radio module more often than ZigBee, since ZInC does not require from routers to relay every source packet to the sink. Instead, they only transmit one coded packet, and then deactivate the radio module till the next data reports. Figure 9 shows this energy consumption cutback. More importantly, ZInC allows a fairer distribution of the energy consumption across the network. Indeed, with ZigBee, routers closer to the sink are more solicited because they forward all source packets and have less sleeping time than other nodes of the network. Figure 10 shows how ZInC implies a better distribution in our testbed. One can see that routers (`0x01`, `0x02`, and `0x03`) consume around the same amount of energy as source nodes, since they forward the same amount of packets and then sleep till the next expected data reports.

VI. CONCLUSION

Throughout this work, we present ZInC, an innovative packet coding scheme designed for ZigBee sensor networks. Our implementation of ZInC in a real testbed proves that index-coding significantly enhances the performance, the reliability, and energy efficiency of many-to-one communications.

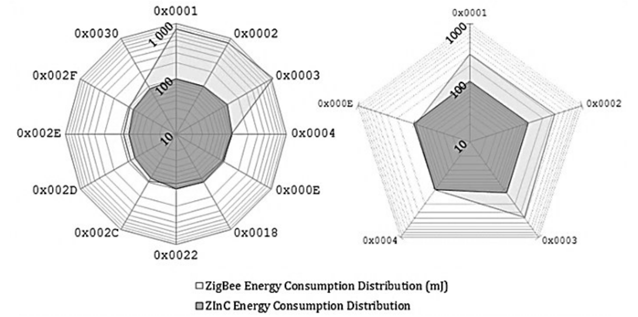

Figure 10 –Latency vs. Number of Sources vs. Data Delivery ratio

The core idea behind ZInC emerged from the under-exploitation of the payload in 802.15.4 networks. Indeed, thanks to index-coding, intermediate nodes can deliver in a distributed manner sensed data using the idle payload spaces at intermediate nodes and thus improve M2O communications.

REFERENCES

[1] R. Ahlswede, N. Cai, S.-yen R. Li, S. Member, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204-1216, Jul. 2000.

[2] J. Barros and S. D. Servetto, "The sensor reachback problem," *in IEEE Trans. Inform. Theory*, vol. 238271, no. November, pp. 1-49, 2003.

[3] T. Cui, L. Chen, and T. Ho, "Energy Efficient Opportunistic Network Coding for Wireless Networks," *IEEE INFOCOM* - Apr. 2008.

[4] A. Dunkels, F. Osterlind, N. Tsiftes, and Z. He, "Software-based on-line energy estimation for sensor nodes," in *Proceedings of the 4th workshop on Embedded networked sensors - EmNets '07*, 2007, p. 28.

[5] C. Fragouli and E. Soljanin, "Network Coding Applications," Foundations and Trends® in Networking, vol. 2, no. 2, , 2007.

[6] C. Fragouli and E. Soljanin, "Network Coding Fundamentals," Foundations and Trends® in Networking, vol. 2, no. 1, pp. 1-133, 2006.

[7] T. Ho et al., "A Random Linear Network Coding Approach to Multicast," IEEE Transactions on Information Theory, Oct. 2006.

[8] I.-H. Hou, Y.-E. Tsai, T. F. Abdelzaher, and I. Gupta, "AdapCode: Adaptive Network Coding for Code Updates in Wireless Sensor Networks," IEEE INFOCOM - 1517-1525, Apr. 2008.

[9] R. Jacobsen, K. Jakobsen, P. Ingtoft, and T. Madsen, "Practical evaluation of partial network coding in wireless sensor networks," 4th International Mobile, 2008.

[10] M. Jafari, L. Keller, C. Fragouli, and K. Argyraki, "Compressed network coding vectors," *IEEE International Symposium on Information Theory*, pp. 109-113, Jun. 2009.

[11] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 3, pp. 497–510, 2008.

[12] H. Seferoglu and A. Markopoulou, "I2NC: Intra-and inter-session network coding for unicast flows in wireless networks," *INFOCOM, 11*.

[13] D. Wang, Q. Zhang, and J. Liu, "Partial Network Coding: Theory and Application for Continuous Sensor Data Collection," IEEE International Workshop on Quality of Service, pp. 93-101, Jun. 2006.

[14] IEEE Standard for Information Technology, "Part 15.4: wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs)". 2003.

[15] A. Toledo and X. Wang, "Efficient Multipath in Sensor Networks using Diffusion and Network Coding," ACISS, , Mar. 2006.

[16] ZigBee Alliance, "ZigBee Specifications". December, 2006.

[17] B. Ishibashi and R. Boutaba. Topology and Mobility Considerations in Mobile Ad hoc Networks, in Ad hoc Networks Journal (Elsevier), Vol. 3, No. 6, pp. 762-776, November 2005.