



HAL
open science

Network Coding for Event-Centric Wireless Sensor Networks

Ismail Salhi, Yacine Ghamri-Doudane, Stéphane Lohier, Gilles Roussel

► **To cite this version:**

Ismail Salhi, Yacine Ghamri-Doudane, Stéphane Lohier, Gilles Roussel. Network Coding for Event-Centric Wireless Sensor Networks. IEEE International Conference on Communication (ICC 2010), IEEE, Jun 2010, Cape Town, South Africa. 6 pp. hal-00693250

HAL Id: hal-00693250

<https://hal.science/hal-00693250>

Submitted on 2 May 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Network Coding for Event-Centric Wireless Sensor Networks

Ismail Salhi¹, Yacine Ghamri-Doudane^{1,2}, Stéphane Lohier¹ and Gilles Roussel¹

¹ Université Paris-Est ; LIGM ; 77420 Marne-la-Vallée Cedex 2, France

² ENSIIE ; 1 Square de la Résistance, 91025 Evry Cedex, France

isalhi@etudiant.univ-mlv.fr, ghamri@ensiie.fr, {lohier, gilles.roussel}@univ-paris-est.fr

Abstract—We propose **Wireless Sensor Coding (WSC)** an event-centric data-dissemination scheme for **Wireless Sensor Networks (WSN)**. WSC is based on a distributed network coding scheme. In this solution, nodes do not only forward packets, but also perform linear random information coding in order to increase the throughput of the network, to reduce the number of transmissions and the end-to-end delay. These improvements involve a reduction of the energy consumption, which is one of the key-issues in WSN architecture design.

Index Terms—Network coding, wireless sensor networks, event-centric dissemination

I. INTRODUCTION

One of the most promising examples of pervasive computing are wireless sensor networks (WSNs). Their ability to automatically produce localized and relevant knowledge in a particular area, makes them a major step-forward in the way we will behave with technology. Indeed, distributed sensing and computing will increasingly expand people's ability to remotely interact with the physical world, whether in outdoor or indoor environments (farms, oil extraction stations, streets, houses, factories, etc.). Several research entities and engineering corporations see in the WSNs a new class of computer systems and a real breakthrough, just like the internet was at the end-seventies.

We will focus in this work on Event-centric wireless sensor networks (EC-WSN). An event-centric WSN can be seen as an applicative view of wireless sensor networks; it is characterized by its real-time nature and the fact that the communication phase is triggered only when a node perceives a notable change in its immediate sensing-range. As well as in classic WSN, the main issues of EC-WSNs remain: the energy consumption cutback, the information delivery efficiency, the robustness, the reliability and the computation constraint.

The application considered in this paper is the following dynamic event-subscription service. Clients use the WSN to get updates from sensors about a specific event interest. For example, mobile clients interested in parking spots availability in a specific area may subscribe to this service and get updates each time the sensor network detects that off-street parking is available. To this end, a complete "fresh" view of the sensed area has to be replicated across the entire network since clients maybe connected anywhere and also be mobile. Such vision of a sensor network reveals some questions, What is the scalability of the system when the frequency of events grows? What is the most appropriate dissemination scheme? What is the convergence time of the system?

We describe in this paper, a data-dissemination system based on network coding for the distribution of critical pieces of information across the sensor network. This cooperative

solution has the advantage of being completely decentralized and self-organized. It delivers the data from the source-sensors (nodes that detect the event) to the whole network by performing random linear coding dissemination. The aim is to increase the throughput of the network, to reduce the number of transmissions and the end-to-end delay, and thus to increase the WSN lifetime. Through our study, we try to give inclusive answers to the questions exposed above and to illustrate the benefits of using network coding techniques in EC-WSNs. Furthermore, our use of the sensor network is original, since data sinks can move inside the network and subscribe to one or more event-interests.

To quantitatively assess our approach, we perform a detailed performance analysis of WSC (Wireless Sensors Coding), our novel data-dissemination protocol for event-centric WSNs. We compare three different dissemination schemes with WSC in various scenarios (high or low event occurrence frequency and different network density) and shows an important transmission cutback in the network and an enhancement of the convergence time.

The reminder of the paper is organized as follows: in Section II we give definitions of EC-WSNs and of network coding; followed in section III by a state of the art about the data dissemination in WSNs. Then, we proceed by describing characteristics of our system in Section IV. We introduce WSC, our distributed propagation scheme, in Section V. WSC is compared with three different protocols in Section VI. The paper concludes with a brief discussion in Section VII.

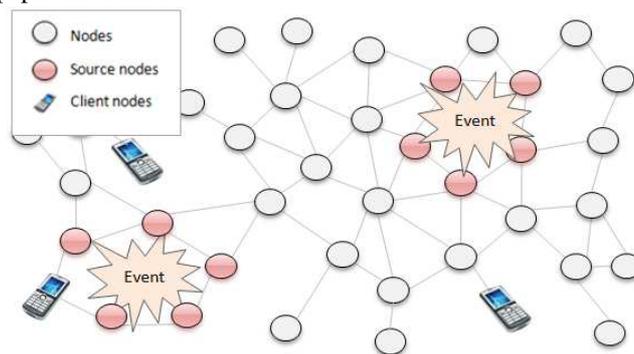


Fig. 1 - Event-centric WSN overview

II. BACKGROUND

A. Event-Centric Wireless Sensor Networks

Event-centric wireless sensor networks differ from common WSNs, where the communications are triggered either by an on-demand or a sink-based process. The purpose of an EC-WSN is to provide a "real-time" view of all the occurring events in a specific area at all the edges of the network (see

Figure 1). EC-WSNs have been well investigated in prior academic works because of their inherent energy consumption efficiency due to the fact that they perform necessarily less radio-transmissions compared to the more traditional periodic or on-demand approach. Authors of [12] for example, use a very large platform of wireless sensors to test their event triggered dissemination scheme for intrusion detection scenarios. In [13], authors propose a new architecture to detect and classify rare, random and ephemeral events for different event-detection applications. While [14] and [15] deal with the network throughput optimization for EC-WSNs. In such architectures, clients (or sinks) are not necessarily static and the communication are no longer on-demand neither sink-based. It is the occurrence of an event that triggers a data transmission.

In this work, we propose to use the EC-WSN as an information provider to mobile clients. In fact, each user can subscribe to one or more event-interest and gets information about it wherever he is, given the assumption that he is connected to the network. We can envision several applications for this type of architecture, particularly in an urban environment. Clients may connect to a large scale event-centric sensor network using their cell-phones or other devices and subscribe to various event-interests like: parking lots or public bikes availability, queues and lines status in different offices (postal office, supermarkets, fuel stations, etc.), or even pollution levels in parks and public places.

B. Network Coding

The network coding theory introduced by Ahlswede *et al.* [5] had a significant impact on the way researchers consider an information network, whether at the practical level (protocol engineering) or the theoretical level (graph and information theories). Indeed, the well established forwarding schemes based on the *store-and-forward* transportation scheme are more to be insufficient in many kinds of communication networks, such as multicast networks [5], or as in wireless ad hoc networks: where [6] for example, shows that the inherent shared and broadcast nature of the radio channel is not well used to increase wireless networks performances. Katti *et al.* provided with their two major contributions: [6] and [7] very interesting ways of using the network coding in wireless ad hoc networks. They have identified important wireless networks properties that the network coding can use to increase the performance of such networks.

The basis of a network coding based communication system is to employ coding mechanisms at intermediate nodes in order to achieve bandwidth optimality [5]. These mechanisms consist essentially of a linear combination of two or more messages in one output message using random coding vectors built from a finite field. The well-known “butterfly example” introduced in [5] explains more in details how network coding helps to reach this optimality.

III. RELATED WORKS

A. Information dissemination in EC-WSN

Previous work has been done in data dissemination within event-centric wireless sensor networks. [1] for example treats

the case of single static data-sink, while [2] deals with N non mobile data-sinks. However, and to the best of our knowledge, the case of data dissemination toward multiple mobile sinks is not treated yet in the literature. Nevertheless, we have chosen three efficient dissemination mechanisms for WSNs to compare with the performance of our solution. It is important to notice that none of these data propagation protocols was designed for our dynamic subscription-based approach of event-centric WSNs, especially for *gossiping* which is not designed for fast data dissemination but rather for minimal energy consumption. Nevertheless, these dissemination algorithms are the most adequate solutions in the literature to be used as an event-centric propagation model.

1) Backoff-based Flooding

In backoff-based flooding, nodes disseminate data to all their neighbors by broadcasting the information. When a node receives a message, it makes a copy of it and waits a random limited duration, and sends it to all its neighbors. The convergence time (i.e. the propagation duration to all the nodes of the network) of this algorithm is relatively short comparing to other approach like *Gossiping* (c.f. below) but several problems emerge from this approach, like data implosion [4] and convergence duration which is a linear function of the network degree.

2) Gossiping

Gossiping [3] is a classical dissemination protocol for WSNs based on randomization. Instead of forwarding data to all the sender’s neighbors, it transmits its flow to one randomly selected neighbor. *Gossiping* reduces data implosion by sending only one copy of each message at any node. However, in order to lower the energy consumption, its data dissemination becomes slow, which is not suitable for event-centric WSNs where fast data propagation is required.

3) SPIN

SPIN [4] is a family of adaptive dissemination protocols based on negotiation and resource-adaptation. It is one of the most efficient propagation schemes for WSNs. However, it requires the utilization of *meta-data* (descriptors about the sensed data) and the presence of a resource-manager that keeps track of resource consumption in every sensor, which is not always feasible in “real-world” situations – in most manufactured sensors, energy consumption status is not available. Moreover, SPIN uses a three phase process for its data propagation with a sequence of three distinct types of messages each time an event occurs, which can lead to a serious overhead in the number of transmissions, even if the meta-data size is less important than the size of the data itself.

B. Network Coding based dissemination

In a network-coding-based dissemination protocol, a linear combination of all available messages is generated by the source using randomly picked coding coefficients, and then sent to the destination. When a node receives enough linearly independent combinations of messages, it can reconstruct the entire original message using a decoding matrix.

In this paper, we propose to use network coding techniques to improve the data dissemination in event-centric wireless sensor networks. Gkantsidis *et al.* have already proposed in [8] a content distribution scheme of large files that is based on

network coding but for Internet peer-to-peer networks. In a comparable approach, we propose to use the randomization and cooperative information propagation induced by the network coding to rapidly disseminate collected information across an event-centric WSN avoiding the implosion problem and reducing the energy consumption.

Our dissemination scheme (detailed in section V) is however different from the solution proposed in [8] where the messages are blocks of one file from one original node. In our case, the messages may be generated by different nodes detecting the event. Another significant particularity of our approach is the fact that there is no need for previous knowledge about the number of messages (blocks in the case of [8]) that were generated at the source level.

IV. SYSTEM DESCRIPTION

A. Event Occurrence Distribution

Consider a system of n sensors scattered uniformly through a square area. We denote by $S = \{1, 2, \dots, n\}$ the set of all the sensors. E is the number of simultaneous new events that may occur and thus are detected by sensors within this area at the same time. We define the event occurrence frequency μ by the number of occurring events every unit of time. μ may follow different random process depending on the EC-WSN application. Each occurring event can be detected by m sensors chosen from S with a certain probability, with $1 \leq m \leq n$. These m nodes compose $D_i = \{1, 2, \dots, m\}$ the set of all sensors that detect an event i . As a result, a node j detects event according to a process of rate $x_j \geq 0$, $j \in D_i$ such that $\sum x_j = \mu$. In the rest of the paper we refer to the detector nodes by *source* nodes.

Any occurring event in an EC-WSN can be detected by several sensors. This redundancy is essential to guarantee lossless data propagation and a better availability of the information. In some cases, this surplus of information can also be used to increase the confidence about the event incidence. For example, in critical application such as fire or flood detection, false positives have to be discarded by validating the event from different sources.

B. Message Propagation

We assume that every source node $j \in D_i$ generates one single message $M_i(j)$, i.e. every event-detection triggers one message transmission from all nodes of D_i . Where $M_i(j)$ includes information about the event i (timestamp, id, meta-data...). The goal of the network after the message generation, is to disseminate the messages $M_i(j)$ describing the event i across the entire network. We denote by $In(k)$ the set of messages already received by a node k . $T_c(i)$ is the *convergence time* of an event i , i.e. the duration necessary for all the messages to be received by nodes of S (equation 1 is verified).

$$\forall j \in D_i, \forall k \in S, \quad M_i(j) \in In(k) \quad (1)$$

C. Event-Interest Subscription

In our approach, final users (subscribers) may be from different types: information display units (digital billboards), mobile users (cell phone, laptops, PDA), or internet gateways

for instance. The only condition is that subscribers have to use the same communication protocol as the one used in the WSN. Since the network is event-centric and all the data are replicated in every node, it becomes simple to any user to get data about its interest anywhere in the network.

V. WIRELESS SENSOR CODING

Source nodes D_i share data of detected event i with a classic *backoff* message broadcast (i.e., each node waits a specific random backoff time before sending its message to avoid collisions). Every message is time-stamped, contains information about the event and has a unique identifier. In order to provide to subscribers a fresh view of the sensed data across the network, information about event i has to be replicated from D_i to every node of S as fast as possible, avoiding the implosion and the famine problems.

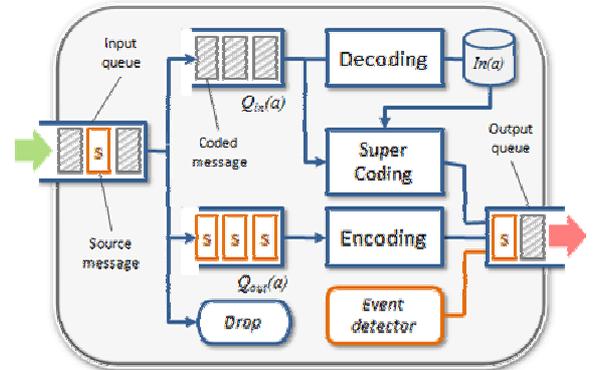


Fig. 2 – WSC queuing system overview

In order to reach this goal, the scheme described by Figure 2 is used after the reception of a message by any node.

A. Message Queuing

Whenever a node $a \in S$ receives a new message $M_i(j)$ from a source node j with ($j \neq a$, $j \in D_i$), if this message is a source message (it has not been encoded), a enqueues the message in what we call its *coding queue* $Q_{out}(a)$ as depicted in Figure 2. If a has already received the message ($M_i(j) \in In(a)$) it simply drops it. However, if the message is already coded, a proceeds to what we call a *Super-coding* operation as detailed in section V.D.

B. Message Encoding

We assume that messages are transmitted as vectors of bits which are of equal length $l(M) = N$, represented as elements in the finite field \mathbb{F}_{2^N} . The length of a vector is equal in all transmissions and for all events. After T_{buf} units of time (the buffering period), all the messages in the coding queue of the node a are encoded following equation 2; the encoded message CM_a is sent with a backoff broadcast to its neighbors.

$$\forall M_i \in Q_{out}(a), \quad CM_a = \sum_{i=0}^{|Q_{out}(a)|} g_a(i).M_i \quad (2)$$

We denote $\vec{g} = (g_a(i))$, the *coding vector* of the message CM_a . In a similar approach as the one proposed in [8] for the case of P2P, we make use of a randomized propagation method where each node selects autonomously and randomly the coding vector's coefficients $g_a(i)$ from a finite field \mathbb{F}_{2^N} .

Each encoded message contains in addition to the coded data CM_a : the message Id , a combination of the coded messages Ids (CODED_IDS) - this value is used by the receiver to know which messages are encoded within CM_a , the coding vector, the source address, the message lifetime, and the number of all encoded messages (see Figure 3).

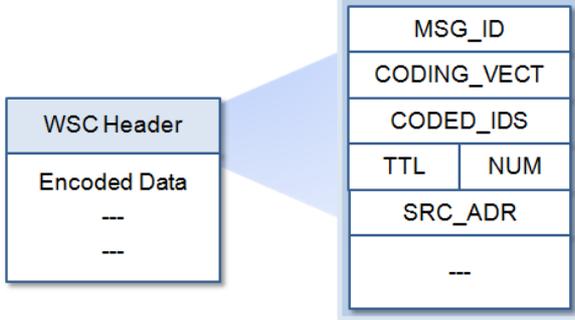


Fig. 3 – WSC message header format

In case where the coding queue is empty, the node reinitiates its buffering timer and waits for the next round. While if $Q_{out}(a)$ contains only one message, node a sends it only once with a backoff broadcast. Note that the number of messages available in $Q_{out}(a)$ is highly correlated with the buffering duration T_{buf} : If the value of T_{buf} is too small, then the node will not have enough time to receive additional messages in its coding queue; in this special case WSC works exactly like *Backoff-based Flooding*. Obviously, when the value of T_{buf} is too large, the coding queue may also runoff space, the coding operations will take more time and messages may be lost. But depending on the expected event occurrence frequency μ , it is simple to choose an appropriate value for T_{buf} that avoids those problems.

C. Message Decoding

The decoding process in WSC consists in solving a linear equation system of u unknowns. We define the *decoding queue* $Q_{in}(a)$ for each node $a \in S$ where all the coded messages that a received are enqueued (see Figure 2). u is the number of distinct *ids* present in the CODED_IDS field of every message within $Q_{in}(a)$. In order to solve the system of equations, each node receives messages which are linear combinations of source messages and stores them into a matrix with their respective coding vectors as shown in equation 3. Where G_t symbolizes the matrix of all the received coding vectors, and h the number of coded messages received and to decode. As shown in [9], it is possible to recover all the source messages M_0, \dots, M_h as long as the matrix G_t has full rank h (see equation 4).

$$\underbrace{\begin{bmatrix} CM_0 \\ \vdots \\ CM_h \end{bmatrix}}_{Q_{in}(a)} = \underbrace{\begin{bmatrix} g_0(0) & \dots & g_h(0) \\ \vdots & \ddots & \vdots \\ g_0(h) & \dots & g_h(h) \end{bmatrix}}_{G_t} \begin{bmatrix} M_0 \\ \vdots \\ M_h \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} M_0 \\ \vdots \\ M_h \end{bmatrix} = G_t^{-1} \begin{bmatrix} CM_0 \\ \vdots \\ CM_h \end{bmatrix} \quad (4)$$

According also to [9], G_t will be invertible with high probability if the coefficients c_i of the coding vectors are random and picked from a finite field \mathbb{F}_{2^N} that is large enough. Authors of [10] set that a size of 2^{16} gives a probability higher than 0.996 for G_t to be invertible.

D. Super-Coding

In many situations, nodes may receive two or more already coded messages. If a node chooses - like nodes l and k in Figure 4 - to forward only one of them (CM_i in this case) and not to proceed to a re-coding operation, neighbors like m might get the same coded message from two different sources (l and k in this case), which can reduce the randomization within the network and thus decrease the decoding speed.

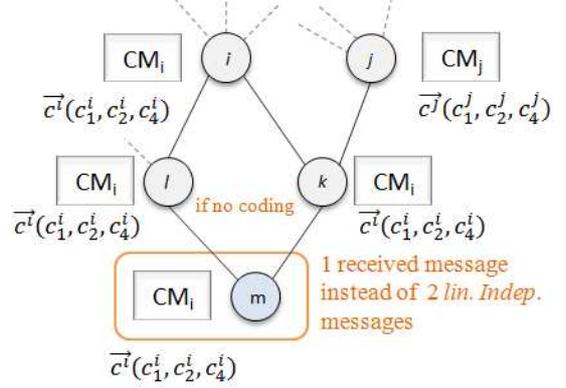


Fig. 4 – Linear dependency problem

To cope with this problem, we propose to adapt the operation of re-coding previously encoded messages employed by Gkantsidis *et al* in [8], to augment the randomization and to create more innovative messages that increase the number of linearly independent equations at the receiver's level.

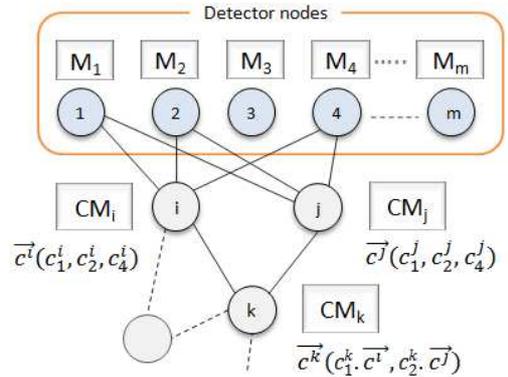


Fig. 5 – Super-coding illustration

Figure 5 shows a case where WSC makes use of super coding. First, the event is detected by nodes $1, 2, \dots, m$, each one generates and broadcasts a message M_1, \dots, M_m . When i and j receives M_1, M_2 and M_4 , they respectively choose $\vec{c}^i = (c_1^i, c_2^i, c_4^i)$ and $\vec{c}^j = (c_1^j, c_2^j, c_4^j)$ as their coding vectors then, they code the source messages into CM_i and CM_j before broadcasting them. When k receives CM_i and CM_j , instead of sending only one of them to avoid redundant message transmission (since both include the same source messages), k

chooses two random coefficients c_1^k, c_2^k and generates a linear combination of CM_i and CM_j into one “super-coded” message CM_k (see equation 5) Before k sends CM_k it also builds a new coding vector as depicted in equation 6:

$$CM_k = c_1^k \cdot CM_i + c_2^k \cdot CM_j \quad (5)$$

$$\vec{c}^k = c_1^k \cdot \vec{c}^i + c_2^k \cdot \vec{c}^j \quad (6)$$

Note that the decoding process remains the same in this case, since the coding vectors are modified each time a super coding operation is used.

VI. PERFORMANCE EVALUATION

A. Preliminaries

We have considered the performance of WSC and compared the results with other dissemination schemes (namely: SPIN, Gossiping and backoff-based flooding) using the Qualnet 4.5 simulation framework (see [11]). We focused in this study on the most important characteristics that may alter the behavior of the information dissemination. Specifically: 1) The event occurrence frequency μ , 2) The number of sensors $|S|$, and 3) The average number of source nodes D_{avg} , with:

$$D_{avg} = \frac{\sum_{i=1}^V |D_i|}{V}$$

Where V is the number of events that occurred during the simulation.

In order to evaluate the performance of WSC and to compare it to each of the three chosen protocols, we have considered two important event-centric WSNs metrics:

1) *The average convergence time*: the convergence time T_c is a key parameter in EC-WSNs, it indicates the duration in which event data is distributed to all the nodes of the network. We suggest to study the impact of μ , $|S|$ and D_{avg} on T_c for WSC, SPIN, Gossiping and backoff-based flooding. Indeed, these three parameters are those that influence the most the performance of these protocols while used in EC-wireless sensor networks.

2) *The number of transmissions*: in order to determine which scheme implies less overhead, we computed the number of transmissions N_{msg} for all the sensors, compared the results and thus deduced the bandwidth efficiency of each scheme.

B. Simulation environment

To implement the selected algorithms, we designed for each one a specific routing module directly linked with the MAC layer of the simulation platform. We also modified the design of the 802.11b physical layer so that the transmission radius and the bit rate correspond more properly to a realistic WSN experience. Moreover, we developed an extension feature for Qualnet that allows to randomly trigger events through the network with a given occurrence frequency.

For our simulation, we used a fully connected wireless sensor network of 150 nodes uniformly spread through a squared plan area of 1500×1500 squared meters. The power of the radio transmitters is set so that the communication radius is about 50 meters. The rest of the simulation attributes are summarized in table 1.

TABLE I – SIMULATION SETTINGS

Simulation time	600 s
Mobility model	None
MAC protocol	802.11b
Capacity	0.5 Mbps
Trans. rang.	~50 m
Max. num. concurrent evt. E	50
Evt. occurrence frequency μ	0.1 0.5 1 5 40 evt./s
Avg. num. of src. nodes D_{avg}	2 6
Buffering period T_{buf}	adaptive
Coding queue Q_{in} size	$500 \times Msg_size$
Decoding queue Q_{out} size	$500 \times Msg_size$
Message size	64 kb

C. Simulation results

Firstly, we calculated for each scheme the value of the convergence time T_c considering different values of μ (see table 1), so we can see which algorithm is faster for the data propagation. In the case of WSC we configured the value of the buffering time T_{buf} relatively to the event frequency. Note that the combination of μ and the average number of source nodes D_{avg} determines the traffic density of the network (as μ and D_{avg} grow the traffic density increases). Thus, we can also infer the scalability of each protocol.

Figure 6 shows the results of this first analysis, and confirms the benefit of using WSC. Indeed, the convergence time of SPIN is higher than WSC, which is due to the fact that SPIN uses a three-step communication system with a sequence of ADV→REQ→DATA messages for its event information propagation. Backoff-based flooding has also a longer convergence time mostly because of its redundant message transmissions. Gossiping has meanwhile the longest convergence duration since its data propagation is very slow (each transmission is addressed to only one destination). The only case where the convergence time of WSC is longer than SPIN is when $\mu = 0.1$; this is due to the fact that the buffering period T_{buf} is longer than the SPIN sequence duration. But in a realistic scenario we can imagine that an EC-WSN has a higher event occurrence frequency than 0.1 event/second.

As depicted in Figure 7, the convergence of WSC remains low as the number of nodes increases comparing to the other protocols. This is due to the fact that WSC is broadcast-based and uses the cooperative data propagation induced by network coding. We can conclude from these results and from Figure 6 that WSC stays efficient even when the traffic load and the number of nodes are important in the network.

We can see from Figure 8 that the number of transmitted messages N_{msg} with WSC is very low compared to SPIN which was predictable because there are no additional messages (ADV and REQ). Since in Gossiping, every node sends only one unicast message to one of its neighbors, the number of messages becomes also very high. N_{msg} in backoff-based flooding is higher than in WSC: the additional messages in backoff-based flooding are due to the fact that each node sends a message to all its neighbors even if they already got it from another path. Note that the energy consumption in WSNs can be inferred from the number of message transmissions. Thus, we can deduce that WSC gives the best speed/energy alternative for event-centric WSNs.

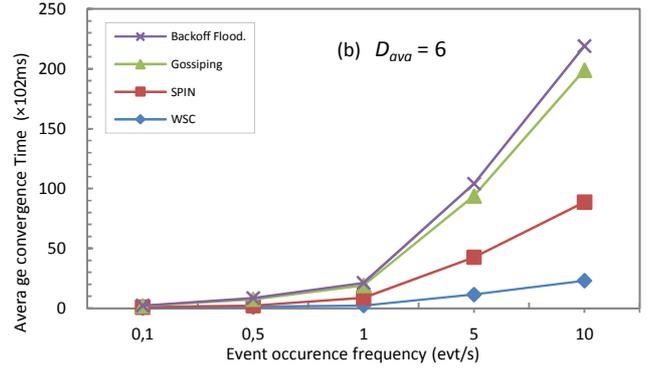
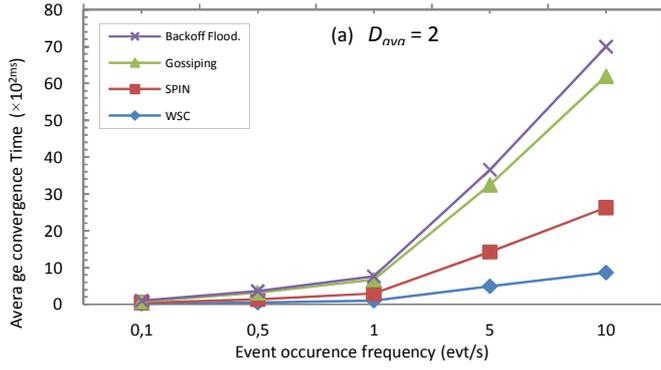


Fig. 6 – Convergence time of the EC-WSN for different event occurrence frequencies (150 nodes)

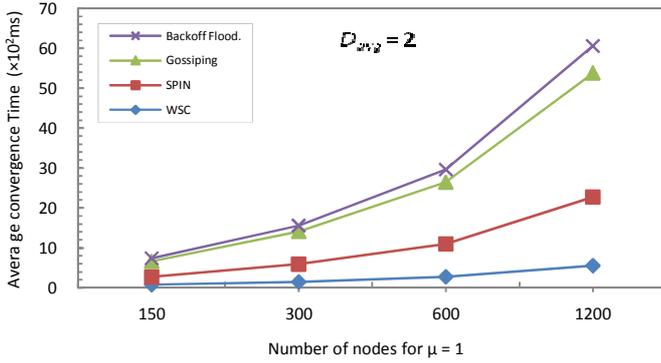


Fig. 7 – Convergence time for different number of nodes

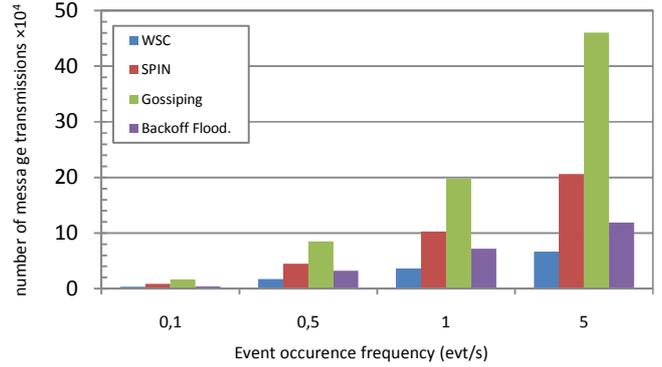


Fig. 8 – N_{msg} for different event occurrence frequencies

VII. CONCLUSION & FUTURE WORK

In this paper, we characterized a new class of event-centric wireless sensor networks, in which every node is an information provider to clients interested in specific events. We introduced WSC, a new dissemination scheme based on Network Coding that allows fast, decentralized and homogenous data propagation from the nodes detecting the events to the entire network. Our performance evaluation compares WSC with SPIN, Gossiping and backoff-based flooding: three major dissemination algorithms for wireless sensor networks. We demonstrate the advantages of using WSC in terms of bandwidth efficiency, where WSC widely outperforms all the other algorithms, and offers the fastest data propagation (convergence time), thanks to its smart exploitation of the shared nature of the wireless medium.

In our study, we do not treat the storage efficiency of the information among the network. Therefore and in the continuation of this work, we plan to extend WSC to a distributed event storage solution in which the information are not distributed to all the nodes but to a set of them, keeping a high information availability and a low energy consumption.

VIII. REFERENCES

- [1] Y. Sankarasubramaniam, O. B. Akan, I. F. Akyildiz. "ESRT: event-to-sink reliable transport in wireless sensor networks." *ACM MobiHoc'03*, June 2003, Annapolis, Maryland, USA.
- [2] C. Intanagonwiwat, R. Govindan, and D. Estrin. "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks." *ACM MobiCom'00*, August 2000, Boston, Massachusetts.

- [3] S. Hedetniemi, S. Hedetniemi, A. Liestman. "A survey of Gossiping and Broadcasting in Communication networks". *Networks*, October 2006.
- [4] W. R. Heinzelman, J. Kulik, H. Balakrishnan. "Adaptive protocols for information dissemination in wireless sensor networks". *MobiCom'99*, August 1999, Washington, USA.
- [5] R. Ahlswede, L. Ning Cai, R. W. Yeung. "Network Information Flow". *IEEE Information Theory*. July 2000.
- [6] S. Deb, M. Effros, T. Ho, D. Karger, R. Koetter, D. S. Lun, M. Médard, N. Ratnakar. "Network coding for wireless applications: A brief tutorial". *IWWAN'05*, May 2005. London, United Kingdom.
- [7] S Katti, H Rahul, W Hu, D Katabi, M Médard, J. Crowcroft. "XORs in the air: practical wireless network coding". *IEEE/ACM Transactions on Networking*. June 2008.
- [8] C. Gkantsidis, P.R. "Rodriguez, Network coding for large scale content distribution", *IEEE/INFOCOM 2005*. Proceedings, August 2005, Miami, USA.
- [9] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting." *IEEE ISIT'03*, June 2003, Yokohama, Japan..
- [10] R. Koetter, M. Medard, "An algebraic approach to network coding", *IEEE/ACM Trans. on Networking*, Nov.2003.
- [11] <http://www.scalable-networks.com>.
- [12] V. Naik, A. Arora, P. Sinha, H. Zhang, "Sprinkler: A Reliable and Scalable Data Dissemination Service for Wireless Embedded Devices", *IEEE RTSS'05*, December 2005, Miami, USA.
- [13] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, D. Culler, "Design of a Wireless Sensor Network Platform for Detecting Rare, Random, and Ephemeral Events", *IEEE IPSN'05*, April 2005, Los Angeles, USA.
- [14] M. Mazo, P. Tabuada, "On Event-Triggered and Self-Triggered Control over Sensor/Actuator Networks", *IEEE CDC'08*, December 2008, Cancun, Mexico.