Network coding in military wireless ad hoc and sensor networks: Experimentation with GardiNet

Ichrak Amdouni, Cédric Adjih, Thierry Plesse

Network Coding in Military Wireless Ad Hoc and Sensor Networks: Experimentation with GardiNet

Ichrak Amdouni*, Cedric Adjih*, Thierry Plesse†
* Inria, France, Email: firstname.name@inria.fr
† DGA, Bruz, France, Email: thierry.plesse@intradef.gouv.fr

***Abstract*—Network coding is a communication paradigm that allows intermediate nodes to mix packets instead of simply relaying them. Motivated by applications in military tactical networks, this paper explores the use of the network coding. It has been proved that network coding has many benefits in wireless communications such as optimal capacity achievement and packet loss recovery. In this paper, we present a generic architecture for network coding for broadcast in wireless networks called GardiNet. GardiNet is generic; its design is divided into building blocks in order to enable it to adapt to different application scenarios. In this paper, we start by describing GardiNet for wireless ad hoc networks and in particular for wireless sensor networks (WSNs). Then, we present experiment results of GardiNet in FIT IoT-LAB, a real testbed of WSNs. Results show that network coding performs well under real wireless conditions. In addition, we highlight the benefit of the Sliding Encoding Window scheme of GardiNet (SEW) to allow sensors to decode packets in real time.**

I. Introduction and Context of This Work

The work presented here was done in the context of the GETRF project [10]. This project targets efficient transmissions in wireless military tactical networks. When sharing information in a military tactical network, it is often necessary to transmit data from one source to multiple destinations. This means that multicast and broadcast are important communication paradigms in military networks. Hence, we focus on these types of communications applied to wireless ad hoc and sensor networks. Our focus to ensure the efficiency of these communications is the use of the network coding. Unlike classical routing methods, the network coding allows intermediate nodes between a source and a destination to send linear combinations of the received packets, instead of simply relaying them. The destination should decode the packets it receives, in order to retrieve the original packets sent by the source. More explanations about this mechanism are provided in Section II-A. Now, we describe our motivations and methodology.

A. Broadcast and Multicast in Tactical Military Networks

In military tactical networks, the transmission optimization is a crucial need. Optimized transmissions reduce overhead and power consumption. Broadcast and multicast transmissions can improve the efficiency of wireless links when transmitting multiple copies of messages using the inherent radio broadcast property.

These two types of communications are used in both control plan and user plan. Concerning the control plan, routing protocols for instance are used in tactical military networks and are based on Mobile Adhoc NETworks (MANET) technologies and architectures. These MANET protocols use broadcast mechanisms for control plan information exchange. As an example, we can site the messages exchanged in OLSR: HELLO and TC (Topology Control) messages.

From a user point of view, a majority of services in a military tactical network are multicast or broadcast in nature as Commands, Situation Awareness information and Push-To-Talk voice services. These services may be present before, during or after the military operation. As examples of these services, we provide the following scenarios:

- The deployment of the forces requires position/status update messages. These messages are of different nature: voice, sensor data, Blue Force Tracking (BFT) and short messages.
- During the operation, most of the data traffic consists of automatic messages (location, status). Speech is used to deliver orders.
- After the military intervention, most of the traffic consists of data, mainly related to the maintenance and logistics.

B. Challenges of Broadcast in Wireless Military Tactical Networks

It appears that the multicast and broadcast are prevalent in military networks. However, some issues need to be solved. First, due to the lossy nature of wireless links, and in the absence of acknowledgement, this type of communication is not reliable. Different TDMA approaches have been proposed in order to implement completely distributed and scalable resource reservation mechanisms for military tactical ad hoc networks. They present certain limitations related to the reliability of the transmissions and the allocation of the radio resources. None of them includes a mechanism to verify the result of the broadcast transmissions (explicit ACK signaling from the receivers), which is an important problem due to the lack of reliability of radio medium. In this paper, we propose a network coding solution that is naturally packet loss resilient. Indeed, a packet has many opportunities to reach the destination as it may appear in many coded packets. Furthermore, the network coding provides an implicit mechanism for acknowledging the broadcast transmissions (via the rank information, as explained in Section II-A).

The second challenge is the efficiency. When multicast/broadcast messages are transmitted in a tactical military MANET, several mechanisms make it likely that a node receives duplicate copies of a message. To ensure an efficient usage of the bandwidth, it is essential to stop a node from transmitting the same message more than once. Thus, the duplicates of a message needs to be detected and eliminated. This is a methodology which is often referred as DPD (Duplicate Packet Detection). The DPD mechanism implies that multicast/broadcast messages are uniquely identifiable. In our work, by applying the network coding, intermediate nodes in the network transmit linear combinations of the received packets and do not forward them directly. This operation significantly reduces the duplication of packets in the network.

C. Our Methodology

Following the context of the GETRF project, we have kept in mind tactical military applications, and focused on two use cases:

- Mobile ad-hoc networks (as used in tactical military networks with microwave or VHF radio communication equipment for instance).
- Wireless sensor networks (used for instance with the goal of monitoring intrusions in strategic areas).

These two cases are characterized by a common feature: they are *multi-hop* wireless networks. This fact has implications on the design of network coding solutions.

During this work, we focused on two objectives:

- Specifying and developing a generic solution for network coding communications: the solution has the name of GardiNet. Our emphasis is on providing a modular and universal solution that can be adopted in the previously cited scenarios and also in many other ones.
- Evaluating the performance of the solution of network coding in general, in the context of multi-hop wireless networks.

II. Related Work

A. Network Coding

In classical wired or wireless networks, coding is restricted to the sources and the end receivers, whereas the intermediate nodes are only in charge of routing and copying payloads. Network coding departs from this traditional end-to-end forwarding paradigm by enabling intermediate nodes to mix the received payloads. Hence, the intermediate nodes may recombine several input payloads into one or several output payloads. The idea of *network coding* has been introduced by Ahlswede, Cai, Li and Yeung in [1]. Since then, research on network coding has attracted significant interest from the research community.

B. Fundamentals of Network Coding

In this section, we present terminology and fundamentals of network coding. We introduce a general framework using the terminology adopted in the network coding taxonomy draft [2].

1) Linear Coding and Random Linear Coding: In network coding, there is **a source** that transmits information to the network. This information is called a **"flow"**. The flow represents a sequence of bytes at the source that needs to be broadcast. The source divides the flow in a sequence of **payloads**. The payloads are numbered, and can be identified by their payload index. The payloads of one flow may optionally be divided in several coding blocks (one by default). The source may have an arbitrary number of flows. We speak about **intra-flow** coding when each flow is coded independently from other flows. We speak about **inter-flow** coding when payloads from different flows can be coded together.

One possible coding algorithm is **linear coding** that performs only linear transformations through **addition and multiplication** (see Li et al. [3] and Koetter et al. [7]). Precisely, linear coding assumes identically sized payloads. These payloads are vectors on a fixed **Galois field**. Let consider any source that multicasts k payloads $(p_j)_{j=1,...,k}$. At any time, any node v receives a payload that is a linear combination of payloads p_j; that is:

$$i^{th} \text{ received coded payload at node v: } y_i^{(v)} = \sum_{j=1}^{j=k} g_{i,j} p_j$$

The sequence of coefficients for a coded payload $y_i^{(v)}$ at the node v denoted $[g_{i,1}, g_{i,2}, ..., g_{i,k}]$ is called the "**coding vector**" of payload $y_i^{(v)}$. The matrix of coefficients $[g_{i,j}]_{i=1...n, j=1...k}$, where n is the number of payloads received by any node v is called the **coding matrix**. Consider the example depicted in Equation 1. The vector $P = [p_1, p_2, \ldots, p_n]$ represents the original payloads generated by the source. Node v has the vector $y_1^v, \ldots, y_n^v$ of coded payloads. This vector is obtained by multiplying the matrix G by the vector P.

$$\begin{pmatrix} y_1^v \\ y_2^v \\ \cdots \\ \cdots \\ y_n^v \end{pmatrix} = \begin{pmatrix} g_{1,1} & g_{1,2} & \cdots & g_{1,k} \\ g_{2,1} & g_{2,2} & \cdots & g_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n,1} & g_{n,2} & \cdots & g_{n,k} \end{pmatrix} \times \begin{pmatrix} p_1 \\ p_2 \\ \cdots \\ \cdots \\ p_n \end{pmatrix} \quad (1)$$

$$Y = G \times P$$

To generate a coded payload with linear coding, any node should select coefficients. Whereas centralized deterministic methods exist, Ho and al. [4] presented a novel coding algorithm, which does not require any central coordination. This algorithm is called **random linear coding**: when a node transmits a payload, it computes a linear combination of all data it possesses with randomly selected coefficients and sends the result of the linear combination. In practice, a special header containing the coding vector of the transmitted payload may be added as proposed by Chou et al. [11].

Thinking in terms of coding vectors, at any point of time, it is possible to associate any node v with **the vector space**, Π_v spawned by its coding vectors, and which is identified with its coding matrix. **The dimension of that vector space, denoted D_v , $D_v \triangleq dim(\Pi_v)$, is also the rank of the matrix**. By

abuse of language, we call **rank of any node** v the rank of its coding matrix.

There are many network coding methods. We speak for instance about **block coding** when the original payload sequence is divided into blocks, called **coding blocks** (as known as **generations**). Coding is performed only over payloads within the same block.

There is also, the **sliding encoding window** when the coding blocks are selected based on a sliding window. For instance, for a window size = $[5, 19]$, the node should code payloads having indices between 5 and 19 (from p_5 to p_{19}). In this case, coding blocks of nodes may be partially overlapping, and, over time, moving to higher original payload sequence numbers. This method has the advantage to allow a real-time decoding; any node is not obliged to wait for the reception of a whole coding block to be able to decode payloads, as in the block coding method. The solution we propose for the network coding is based on a sliding window as we will see in the remaining of this article.

2) Decoding and Rank: The rank of a node is a direct metric for the amount of useful received payloads, and a received payload is called **innovative** when it increases the rank of the receiving node. Ultimately, a node can decode all source payloads when its rank is equal to the total number of source payloads. In this case, we say that the decoding matrix **has full rank**. Decoding is done by **inverting the coding matrix** formed by the coding coefficients. As seen in the example of Equation 1, $Y = G \times P$. Hence, the original payloads $(p_i, i = 1 \dots n)$ can be determined by inverting the matrix G: $P = G^{-1} \times Y$ (see Equation 2). Matrix inversion can be performed by **Gauss Elimination**.

$$G^{-1} \times \begin{pmatrix} y_1^v \\ y_2^v \\ \cdots \\ \cdots \\ y_n^v \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \times \begin{pmatrix} p_1 \\ p_2 \\ \cdots \\ \cdots \\ p_n \end{pmatrix} \quad (2)$$

$$G^{-1} \times Y = I \times P$$

C. Benefits of Network Coding

The efficiency provided by network coding in multicast and broadcast communications has been studied for instance, by Lun et al. [12], and Wu et al. [13]. In particular, they provide methods for determining optimal network coding parameters for a given network with specific model assumptions. The work of Fragouli et al. [14] gives insights for all-to-all broadcast and illustrates how gains could be obtained compared to classical routing.

It has been argued that the network coding is best suited to multi-hop wireless networks. Indeed, compared to wired networks, they have specific properties, see for instance [15], including:

- Wireless 'neighborcast': one wireless transmission by a node may reach several receivers. This property may be used to optimize broadcast.
- Time-variation: the visibility between two nodes may evolve with time, due to node mobility, physical changes in the propagation environment or other reasons.
- Unreliability of wireless communications: due to wireless channel conditions or properties, transmission losses (packet erasures) potentially occur.

Intuitively, by combining the received packets, a coded packet sent by an intermediate node could benefit multiple receivers simultaneously, thus improving the bandwidth efficiency. In [1], it was shown that the multicast capacity (that is the maximum number of packets that can be sent from the source to a set of terminals per time unit) can be achieved by performing network coding at the intermediate nodes. A few years later, in [3], it was shown that for multicast networks, linear coding suffices to achieve the capacity limit, which is the max-flow from the source to each receiving node.

The authors of [5] show that random linear coding technique performs asymptotically as efficiently as any other network coding method in terms of capacity, for the case of single source multicast [5], and its performance is determined entirely by the average rate of nodes [6].

By reducing the number of transmissions required to transmit some amount of information, network coding achieves energy efficiency. In [16], authors analyse the performance of network coding in torus networks. Results show that the capacity in a torus grid is equal to the number of neighbors of a node. Moreover, network coding in such networks is "near optimal" in terms of energy efficiency, in the sense that each transmission will provide innovative information (outside the vicinity of the source).

D. DRAGONCAST

DRAGONCAST [8], [9] is a protocol for broadcasting a set of packets from one source to the entire network with network coding. The base functioning is simple: the broadcast is initiated by transmissions from the source. Every node in the network retransmits coded payloads with a changing interval between transmissions. At the same time, every node collects received coded payloads and performs decoding as they are received. Finally, termination is automatically detected when all the nodes have successfully received all data.

In this work, we have developed a solution called GardiNet which specifies and completes the different modules of DRAGONCAST by the information base and the signaling required to perform network coding in a wireless network.

III. GardiNet: Generic Solution for Network Coding

A. Overview

GardiNet is a generic framework for network coding in wireless networks. It is based on intra-flow coding where the source divides the flow in a sequence of payloads of equal size (padding may be used). The design keys of GardiNet are simplicity and universality; GardiNet does not use explicit or implicit knowledge about the topology (such as the direction or distance to the source, the loss rate of the links, etc). Hence,

it is perfectly suited to the most dynamic wireless networks. The protocol is distributed and requires minimal coordination. GardiNet architecture is modular, it is based on 5 building blocks (LIB, SIG, Protocol, SEW and DRAGON). Each block is almost independent. This makes GardiNet generic and hence adaptable to many application scenarios.

GardiNet derives from an existing protocol called DRAGONCAST. Indeed, GardiNet shares the same principles and theoretical overview of DRAGONCAST. It enriches DRAGONCAST by the information base and signaling required to perform broadcast in wireless networks and in wireless sensor networks in particular. Furthermore, GardiNet specifies the different blocks of DRAGONCAST. The IRTF draft [9] is a detailed description of GardiNet and the material of this paper derives from this draft. Notice however, that in this draft, the same acronym of DRAGONCAST is kept. For clarity reasons, we replace it by GardiNet for the remaining of this paper. Also, some acronyms are changed compared to this draft.

B. The Building Blocks of GardiNet

Figure 1 illustrates the different building blocks of GardiNet.

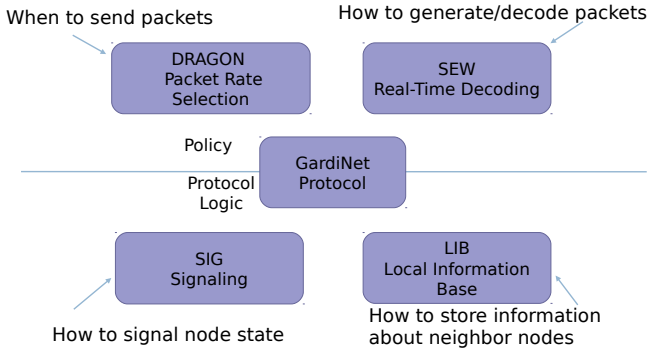

Fig. 1. Organization of the different building blocks of GardiNet.

To make GardiNet the most universal possible, we separate the design of its building blocks into two categories: the protocol and the policy. The protocol category defines the general protocol aspects themselves and includes the information base LIB, the signalisation SIG and the Protocol itself. The policy category defines the higher protocol behavior and it includes the sliding encoding window SEW and the dynamic rate adaptation DRAGON.
In the following, we give a brief description of all these building blocks.

1) LIB: Local Information Base: This module is responsible for maintaining all information required for the functioning of the protocol. This information base maintains information about the flows, the decoding process, and the state of the neighbors.

2) SIG: SIGnalisation: This module provides the signaling for the control plane for GardiNet. The signaling consists mainly on the specification of a header for each coded payload (e.g piggybacking). It includes information relative to the state of the node, in addition to the packet encoding information. This allows each node to maintain information about the state of its neighbors.

3) Protocol: This module is the protocol itself with message generation and message processing.

4) SEW: Sliding Encoding Window: SEW is a real-time decoding method. This method relies on implicit cooperation between neighbor nodes, in order to allow recovery of some source payloads without requiring to decode all source payloads at once. Technically, as described in [8], it ensures the existence of a low triangle in the coding matrix during the online Gauss elimination process. The method SEW relies on two principles:

- SEW coding rule: generates only coded payloads that are linear combinations within a given window. The determination of this window is a policy for SEW.
- SEW decoding rule: when decoding, performs a Gaussian elimination in such a way that one coded payload is only used to eliminate the source payload with the highest possible index (i.e. the latest source payload).

5) DRAGON: Dynamic Rate Adaptation from Gap with Other Nodes: DRAGON is a dynamic payload rate adjustment policy. Every node transmits coded payloads with a specific payload rate. With DRAGON, this rate is adjusted dynamically. Essentially, the rate of the node increases if it detects that some nodes in the current neighborhood are "falling behind" in the decoding process. This is called a "dimension gap". DRAGON provides a heuristic to avoid this gap.

C. General Functioning

Figure 2 illustrates the general functioning of GardiNet.

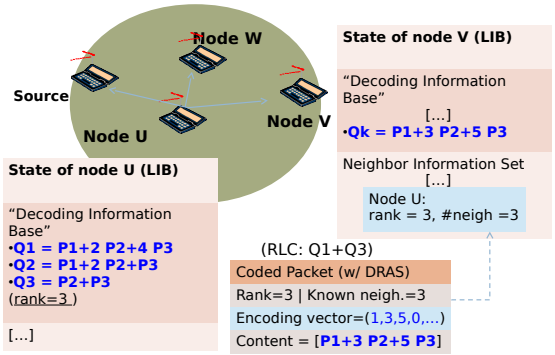

Fig. 2. The general functioning of GardiNet.

The source initiates broadcasting by sending its original data payloads. These payloads have a predefined, constant

size; padding can be used if necessary. Other nodes initiate transmission of encoded payloads upon receiving the first coded payload. As an example, we observe that the node u in Figure 2 has received payloads Q_1, Q_2 and Q_3 and has rank=3. These payloads are stored locally in the decoding information base as we will specify later. In addition to payloads, any node stores information relative to its neighbors. As an example, the node v has information about its neighbor u: (the rank and the number of neighbors). Notice that received coded payloads are stored only when they are innovative. Then, nodes transmit payloads periodically. The transmission periodicity is decided by payloads rate selection algorithms. Precisely, when intermediate nodes receive a data payload that is a source payload or a coded payload, they start scheduling encoded data transmission. The scheduling interval is decided by the policy DRAGON. Payloads transmitted by intermediate nodes are coded payloads generated using random linear coding with a specified header. This header contains information needed for decoding. For instance, we see in Figure 2 that node u piggybacks its message by adding its rank, the number of its neighbors, the encoding vector and the coded payload. Data transmission continues until nodes detect the termination condition, i.e. when themselves and all their neighbors have successfully decoded the data stream. However, a node may re-enter the transmission state. This happens when it receives a notification indicating that one neighboring node requires more coding vectors to recover some source payloads.

D. GardiNet in Practice: Heuristics

In this section, we propose two heuristics for GardiNet. First, we explain how SEW encodes and decodes payloads. Second, we describe how DRAGON adjusts rates.

1) ***Coding Rule of SEW****:*

We introduce the following definitions.

*Definition 1 (****highest (resp. lowest) index of a coded payload****):* The highest (resp. lowest) index of a coded payload, is the maximum (resp. minimum) index of original encoded payloads (payloads generated by the source).

Example:

For the payload $Q = P_3 + P_5 + P_7 + P_8$, the highest index is 8 and the lowest index is 3.

Because all coded payloads have their own highest index and lowest index, we can also compute the maximum of the highest indices of all undecoded payloads at any node, as well as the minimum of the lowest indices. Hence, we define:

*Definition 2 (****The high (resp. low) index of any node****):* The high (resp. low) index of any node is the maximum (minimum) index of all its undecoded payloads.

Notice that a node will generally decode the source payloads from 1 up to its low index.

To ensure real-time decoding, SEW uses knowledge about the state of neighbors of one node, namely their low index. Any node restricts the generated payloads to a subset of payloads of the source such that its perceived neighbors are able to decode nearly all of them, up to a margin K. Notice that once all these neighbors may decode up to the first L-K payloads, it is unnecessary for the node to include payloads $P_1, \dots P_L$ in its generated combinations.

Hence, the general idea of SEW is to encode payloads with indices within a given window a fixed size K. In other words, the i-th generated payload $q(v,i)$ of any node v is given by: i-th generated payload $q(v,i) = a(v,i,k)P_k + ... + a(v,i,k+K)P_{k+K}$, where $(P_j, j = k, \dots k+K)$ is the set of payloads generated by the source, The sequence of coefficients for $q(v,i)$ is the following coding vector: $[0, 0, ..., a(v,i,k), a(v,i,k+1), ..., a(v,i,k+K), ..., 0, 0]$.

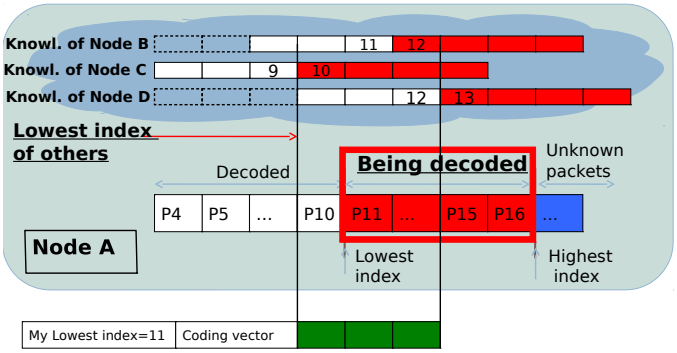

Fig. 3. Illustration of coding rule of SEW

As an example, we consider the state of node A in Figure 3. Node A has decoded payloads until index 10. Its neighbors B, C and D have lowest indices equal to 12, 10 and 13 respectively. Hence, node A codes payloads starting from index 10 which is the minimum low index of its neighbors.

A node will repeat transmissions of new random combinations within the same window, until its neighbors progress in the decoding process.

2) ***Decoding Rule of SEW****:*

The intent of the SEW decoding rule, is to guarantee proper functioning of the Gaussian elimination. An example of SEW decoding rule is the following: assume that node v has received payloads q_1 and q_2, for instance $q_1 = P_1 + P_9$ and $q_2 = P_1 + P_2 + P_3$. Then q_1 would be used to eliminate P_9 for newly incoming payloads (the highest possible index is 9), and q_2 would be used to eliminate P_3 from further incoming payloads. On the contrary, if the SEW decoding rule was not applied and if q_1 were used to eliminate P_1, then it would be used to eliminate it in q_2, and would result into the computation of $q_2 - q_1 = P_2 + P_3 - P_9$; this quantity now requires elimination of P_9, a higher index than the initial one in q_2. In contrast, the SEW decoding rule guarantees the following invariant: during the Gaussian elimination process, the highest index of every currently undecoded payload will always stay identical or decrease.

Provided that the neighbor state is properly exchanged and known, the combination of the SEW coding rule and the SEW

decoding rule, guarantees that ultimately every node will be able to decode the payloads in the window starting from its lowest index; that is, they guarantee early decoding.

Notice that improper knowledge of neighbor state might impact the performance of the method but not its correctness: if a node detects a previously unknown neighbor (for instance due to mobility), it will properly adjust its encoding window. Similarly, in GardiNet, obsolete neighbor information, for instance about disappeared neighbors, will ultimately expire.

*3) **DRAGON: Packet Rate Selection**:*
The heuristic DRAGON has proposed and analyzed in [8] is inspired by Fragouli et al. [17]. We briefly summarize it in this section. The starting point of our heuristic DRAGON is the following observation. Indeed, for real-time decoding, the rank of nodes inside the network should be close to the index of the last source payload, and that in any case, they should at least evolve in parallel. Thus, one would expect the rank of any node to grow at the same pace as the source transmission, as in the example of optimal rate selections for static networks. Decreasing the rates of intermediate nodes by a too large factor, would not permit the proper propagation of source payloads in real time. On the contrary, increasing excessively their rates, would not increase the rate of the decoded payloads (naturally bounded by the source rate) while it would decrease energy-efficiency (by increasing the amount of redundant transmissions). The idea of the proposed rate selection is to find a balance between these two inefficient states. As we have seen, ideally the rank of a node would be comparable to the lastly sent source payload. Since we wish to have a simple decentralized algorithm, instead of comparing with the source, we compare indirectly the rank of a node with the rank of all its perceived neighbors. The key idea is to perform a control so that the rank of neighbor nodes would tend to be equalized: if a node detects that one neighbor had a rank which is too low compared to its own, it would tend to increase its rate. Conversely, if all its neighbors have greater ranks than itself, the node does not need to send payloads in fact.

Hence, DRAGON uses the following heuristic. Let present first these definitions:

- $D(v,t)$ denotes the rank of any node v at time t.
- $N(v,t)$ denotes the number of neighbors of the node v at time t.
- $g(v,t)$ denotes the maximum rank gap of v compared to its neighbors, normalized by the number of these neighbors. Then $g(v,t)$ is evaluated as:

$$g(v,t) = Max_{for\ all\ u\ neighbor\ of\ v} \frac{D(v,t) - D(u,t)}{N(u,t)}$$

- We determine $C(v,t)$: the payload rate of any node v at time t as follows:
  - if $g(v,t) > 0$ then: $C(v,t) = A\ g(v,t)$ where A is some constant.
  - Otherwise, the node stops sending encoded payloads until $g(v,t)$ becomes larger than 0.

When computing the payload rate selection, the node uses information about its neighbors stored in the Neighbor Information Base. Indeed, any node needs the rank of each of its neighbors as well as their total number. This information is deduced from the last received payload. Although this payload might not necessarily reflect the exact values at the computation time, they provide an estimate.

IV. Experimentation of GardiNet in FIT IoT-LAB

We evaluate the performance of GardiNet, we run experiments on a real WSN testbed. we implemented SEW heuristic defined in Section III-D1. The rate adaptation DRAGON is not activated in this version.

A. WSN platform FTT IoT-LAB

IoT-LAB [13] is a very large scale testbed, remotely accessible, and includes a total number of 2728 nodes (in 6 sites); the nodes are mostly of type "wireless sensor nodes" with a wireless radio, and are well suited to perform wireless protocol experiments.

B. Setting of the Experiment

We run experiment on the Euratech testbed in the region of Lille using 20 nodes arranged in a line. Figure 4 shows a part of this testbed. A source generates periodically payloads. Nodes receiving this payload discard it if it is not innovative and try to decode payloads. Then, they generate a random linear combination and broadcast it.

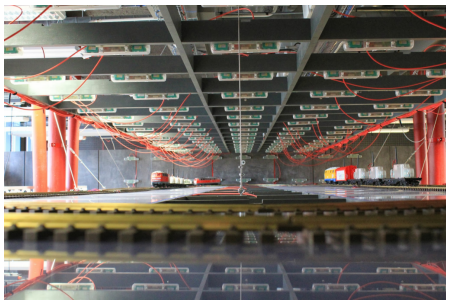

Fig. 4. Euratech testbed in Inria Lille.

The nodes are equipped with MSP430F1611 microcontroller (16-bit, 48kB flash and 10kB RAM) and CC2420 radio.

In the configuration tested, the cumulative distribution of the loss rate is illustrated in Figure 5. Notice that the network is highly lossy. Less than 20% of the links have less than 18% loss rate, and in the same spirit, more than 40% have a loss rate greater than 40%.

C. Results

We first study the evolution of the network in terms of rank and number of decoded packets. These results are depicted in Figures 6, 7 and 8. We set the window size to 15.

From these figures, we notice the following remarks.

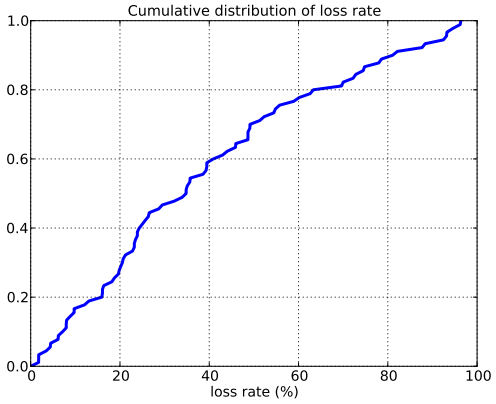

Fig. 5. Cumulative distribution of the loss rate.

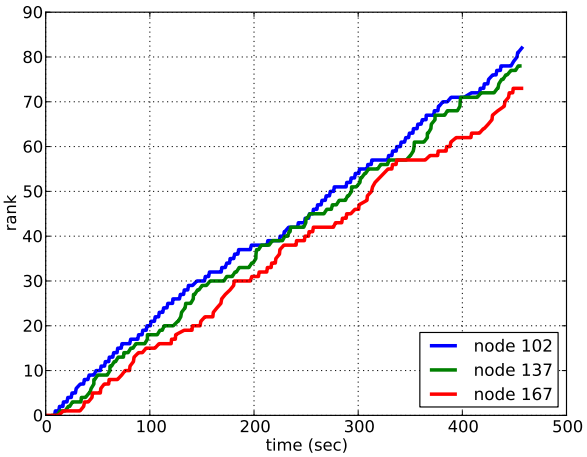

Fig. 6. The rank evolution of nodes 102, 137 and 167.

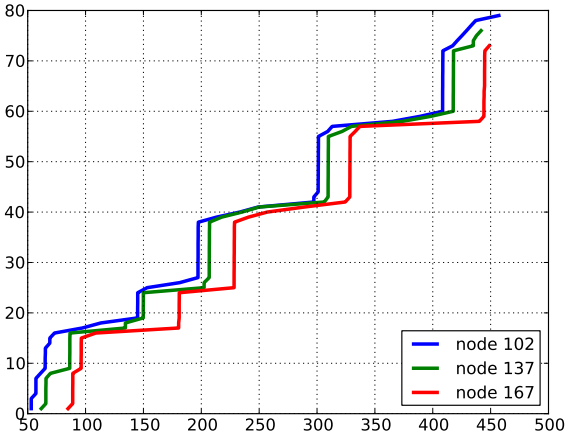

Fig. 7. The number of decoded packets at nodes 102, 137 and 167.

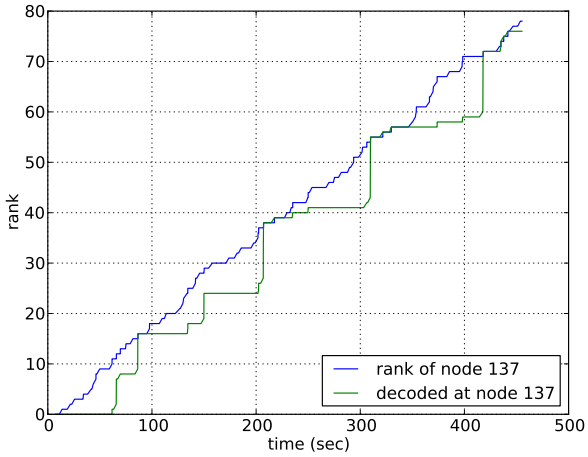

Fig. 8. The evolution of the rank and the number of decoded packets at node 137.

- The rank and the decoding process evolve progressively in time, which means that the GardiNet nodes are not blocked.
- The rank and the number of decoded payloads evolve very closely (see Figure 8). This is the advantage of the real time decoding of GardiNet. Whereas, with classical random linear coding, decoding might occur at the end.
- However, we still notice some plateau in the number of decoded packets plot. This is expected and corresponds to the instants where the neighbors of selected nodes advance their encoding window. Indeed, nodes code packets from their low_index to $low\_index+K$, where K is the window size. When this index in incremented, the receiver nodes cannot decode the received packets immediately because these packets have not been seen previously. This is confirmed by the result in Figure 9 that illustrates the evolution of the lowest index at nodes 97, 102 and 137.
- Notice that, globally, the progress of the rank at the selected nodes is not very different. Same remark applies for the number of decoded packets. This means that the network nodes evolve in parallel. Of course, this result is positive when the nodes progress is good. It is the case here. The window mechanism of SEW allows nodes to have near progress speed which is beneficial mainly for real time applications.
  However, we notice that the progress of the rank of node 167 is the slowest one (see Figure 6). This is because this node is the farthest node from the source compared to other nodes (see Figure 10 where the source is node 97). Indeed, the probability to receive innovative packets is higher at nodes close to the source. Hence, these nodes would decode earlier.

Now, we set the window size to 25 and run the same experiment. We want to determine the impact of the window size.

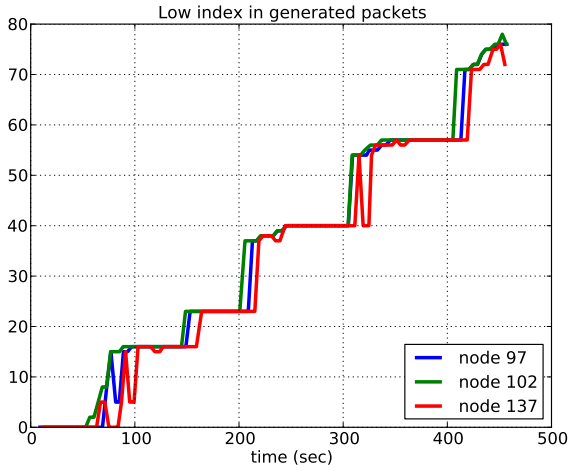

Fig. 9. Low index of generated packets at nodes 97, 102 and 137.

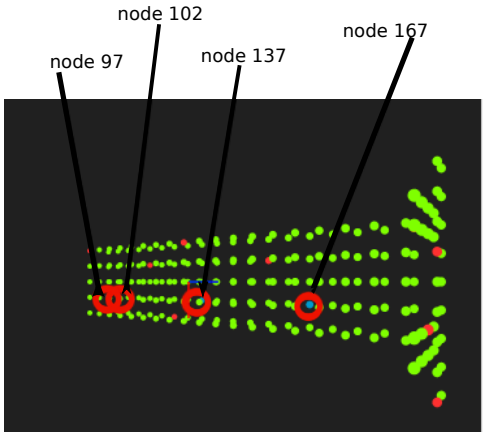

Fig. 10. Nodes whose statistics are depicted in the plots.

We observe that the rank evolution is almost not impacted by the window size (see Figures 11 and 12). The number of decoded packets is slightly impacted (see Figures 13 and 14). When the window size increases, the decoding is relayed. For instance, node 102 decodes almost 41 packets at instant 300 seconds if the window size is 15, and decodes the same number of packets at 350 seconds if the window size is 25. This is because, when the window size increases, the number of undecoded packets at any node increases (new column in the coding matrix), and hence more time is needed for the node to decode them.

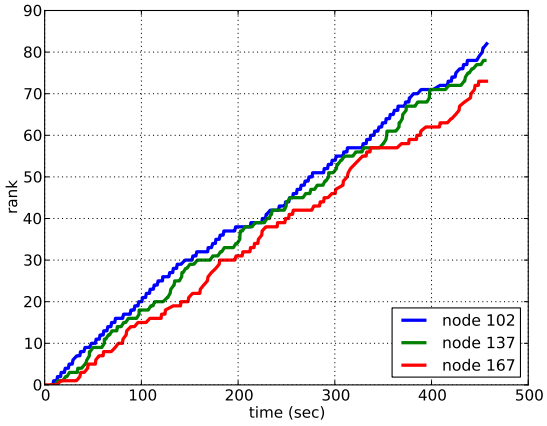

Fig. 11. The rank of nodes 102, 137 and 167 for window size = 15.

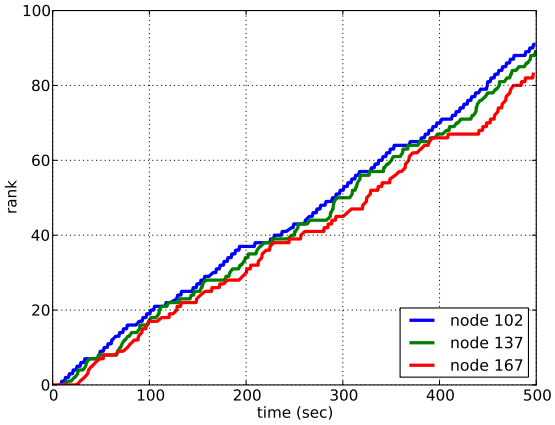

Fig. 12. The rank of nodes 102, 137 and 167 for window size = 25.

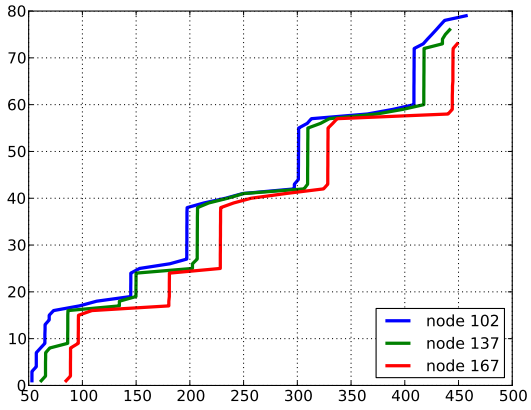

Fig. 13. The number of decoded packets at nodes 102, 137 and 167 for window size = 15.

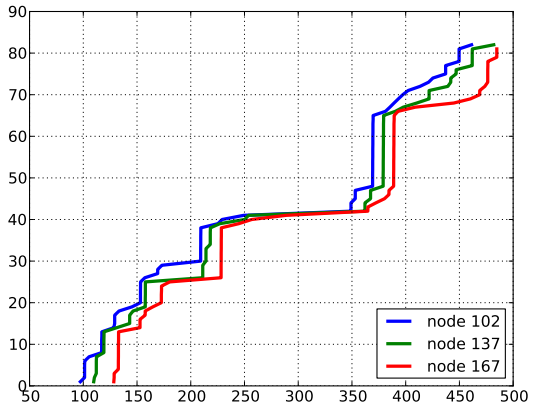

Fig. 14. The number of decoded packets at nodes 102, 137 and 167 for window size = 25.

V. Conclusion

In this paper we described GardiNet, a generic architecture for network coding applied to broadcast in wireless networks. We conducted an experimental study on a real testbed of wireless sensor networks. Results prove that GardiNet works well under real conditions even on wireless sensor nodes with limited resources. Also, we noticed that the network coding operates well even with highly lossy and unreliable links found in our testbed. This confirms that GardiNet meets the known benefits of network coding in wireless networks. We also illustrate the sliding encoding window mechanism SEW. Recall that according to SEW, nodes encode payloads regarding only the first undecoded payload at neighbors. However, in the network, nodes may have heterogeneous encoding rates. Hence, we may find late nodes, that is nodes whose first undecoded payload is very smaller than the first undecoded index of all their neighbors. Such late nodes may prevent their neighbors from increasing their sliding window and then slow their decoding progress. As a future work, we plan to enhance SEW to handle this heterogeneity issue. Also, we will consider the buffer overflow problem.

ACKNOWLEDGMENT

This work has been partly supported by DGA/ASTRID/ANR-11-ASTR-0033 under the GETRF project.

References

[1] Ahlswede R., Ning Cai, Li S.-Y.R., Yeung R.W., *Network information Flow*, Information Theory, IEEE Transactions on, 46(4):1204-1216, July 2000.

[2] Firoiu V., Adamson B., Roca V., Adjih C., Bilbao J., Fitzek F., Masucci A., M. Montpetit, *N*etwork Coding Taxonomy, draft-firoiu-nwcrg-network-coding-taxonomy-01 (work in progress), March 2014. http://tools.ietf.org/html/draft-firoiu-nwcrg-network-coding-taxonomy-01

[3] Li S-Y R., Yeung R. W., Cai N., *Linear network coding*, IEEE Transactions on Information Theory, vol. 49, no. 2, pp. 371-381, Feb. 2003.

[4] Ho T., Koetter R., Médard M., Karger D.R., Effros M., *The benefits of coding over routing in a randomized setting*, In Proceedings of 2003 IEEE International Symposium on Information Theory, 2003.

[5] Lun D. S., Médard M., Koetter R., Effros M., *Further results on coding for reliable communication over packet networks*, CoRR, abs/cs/0508047, 2005.

[6] Lun D. S., Médard M., Koetter R., Effros M., *On coding for reliable communication over packet networks*, Physical Communication, 1(1):3-20, 2008.

[7] Koetter R., Medard M., *An algebraic approach to network coding*, IEEE/ACM Trans. Networking, vol. 11, no. 5, pp. 782-95, Oct. 2003.

[8] Cho S-Y., Adjih C., *W*ireless Broadcast with Network Coding: Dynamic Rate Selection, MedHocNet 2008, June 2008.

[9] Adjih C., Cho S-Y., Baccelli E., *Broadcast With Network Coding: DRAGONCAST*, draft-adjih-dragoncast-00 (work in progress), July 2013. http://tools.ietf.org/html/draft-adjih-dragoncast-00

[10] http://getrf.gforge.inria.fr/

[11] Chou P. A., Wu Y., Jain K., *Practical Network Coding*, 43^{th} Annual Allerton Conference on Communication, Control, and Computing, Monticello, IL, Oct. 2003.

[12] Lun D.S., Ratnakar N., Medard M., Koetter R., Karger D.R., Ho T., Ahmed E., Zhao F., *Minimum-cost multicast over coded packet networks*, IEEE Transactions on Information Theory, 52(6):2608-2623, June 2006.

[13] Wu Y., Chou P. A., Sun-Yuan Kung, *Minimum-energy multicast in mobile ad hoc networks using network coding*, IEEE Transactions on Communications, 53(11):1906-1918, 2005.

[14] Fragouli C., Widmer J., Le Boudec J-Y., *Efficient Broadcasting Using Network Coding*, IEEE/ACM Transactions on Networking, 16(2):450-463, 2008.

[15] Baccelli E., Perkins C., *Multi-hop Ad Hoc Wireless Communication*, draft-baccelli- manet-multihop-communication-02 (work in progress), July 2013.

[16] Masucci A., Adjih C., *Capacité de Diffusion avec Codage Réseau dans les Grilles Torique*, ALGOTEL 2014, 16‘emes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, pp. 1-4, June 2014.

[17] Fragouli C., Widmer J., J. Le Boudec, *A Network Coding Approach to Energy Efficient Broadcasting*, INFOCOM 2006, April 2006.