HAND: HEADER-ASSISTED NETWORK DECODING

Qiuyi Wang[1]*, Yang Xiao*[1]*, Michel Kieffer*[1]*, and Cédric Adjih*[2]

[1] L2S, CNRS–CentraleSupelec–Univ Paris-Sud, Univ Paris-Saclay,
3 rue Joliot-Curie, 91192 Gif-sur-Yvette, France.
[2]Inria, Université Paris-Saclay, France.

ABSTRACT

This paper considers the problem of data collection in a sensor network using network coding (NC). It proposes a decoding approach, called HAND, which does not require source packets to be supplemented with NC headers (with encoding vectors), classically used to decode the network-coded packets. HAND exploits the structure imposed by the communication protocol on the packet headers to estimate the original source packets from the received network-coded packets. Network-decoded packets are obtained as the solution of systems of linear equations. The decoding complexity is only one order of magnitude larger than that of classical network decoding.

1. INTRODUCTION

Data collection using sensor networks [1] or via crowdsensing [2, 3] consists in gathering data collected by sensing devices at some central data processing node or sink. With long range radio technologies [4], data collection is relatively easy; nevertheless, many uses cases rely on shorter range radio technologies (for instance, standards [5, 6, 7] based on IEEE 802.15.4 [8] and 6LoWPAN [9]). In such networks, data packets have usually to be forwarded by intermediate nodes using multi-hop communication. Several approaches may be considered, such as routing [10, 11, 12], or network coding (NC) [13, 14, 15, 16, 17]. In general, in wireless networks, the inherent broadcast capacity of the channel provides added benefits to NC and improves the communication efficiency [18, 19, 20].

In case of NC, most of the protocols transmit, along with the coded payload, a description of the associated linear combination: coefficients and involved packets. As example, the header of COPE [19] includes the list of 32-bits identifiers of XOR-ed packets. *Practical NC* (PNC) [21] proposes a general solution: source packets are grouped into generations of fixed size, and are numbered. The header of a network coded packet is a fixed size *encoding vector* corresponding to the coefficient of each packet of the generation, *e.g.* $(0\ 7\ 5\ \ldots)$ for $7\mathbf{x}_2 + 5\mathbf{x}_3$. Only packets of the same generation are mixed together. For data collection, once enough independent linear combinations from one generation are received by the sink, the latter can perform network decoding via simple Gaussian elimination.

When the number of nodes of the network, or the size of generations is large, the overhead due to the PNC headers may become costly. To address this issue, compressed forms of NC headers have been proposed [19, 22, 23]. Nevertheless, even if headers are compressed, nodes have still to agree on a global indexing of packets within a generation, or have to use sufficiently long packet identifiers within a generation to ensure decodability of the network-coded packets. Such global agreement may be reached in a fully controlled static wireless sensor network, but it is much more complex to obtain in a network with high churn rate, especially with crowdsensing.

Our aim in this paper is to show that in data collection applications based on NC, the PNC header may be removed, while still being able to perform network decoding. This can be obtained by implementing NC as a *shim* between IP and the MAC layer [24, 19]: the NC shim recognizes the structure of headers of upper layer packets (e.g. IPv6), before coding them together. Network-decoding is guided by the fact that decoded packets should verify the constraints of the structure of the headers of input packets. This is the main idea of the proposed Header-Assisted Network Decoding (HAND) approach. Headers introduced by upper layers in the protocol stack contain enough redundancy to infer the NC operations applied on the received packets by observing the combined headers, and to recover the original packets by inverting these operations.

A closely related idea is independent component analysis over finite fields [25, 26] applied in the context of network decoding [27]. In this approach, network-coded packets without NC headers are decoded using entropy minimization. In Protocol-Assisted Network Decoding (PANDA) [28], the redundancy introduced by communication protocols is exploited to assist the receiver in decoding network-coded packets via maximum *a posteriori* estimation. Nevertheless, both approaches have an exponential complexity in the size of the generation considered. HAND exploits efficiently the structure and *a priori* information about packet headers to obtain network decoded packets as the solution of systems of linear equations. This significantly reduces the computational complexity.

The rest of this paper is organized as follows. Section 2 states the network decoding problem in absence of NC headers. Section 3 describes how known information in packets headers can be exploited and presents the proposed decoding method. A comparison of HAND with PANDA and with classical network decoding is provided in Section 4. Finally, Section 5 draws some conclusions.

2. PROBLEM FORMULATION

Consider a network $\mathcal{N}$ of N wireless sensor nodes taking measurements of some physical quantities that have to be transmitted to some sink s. Each node i can only communicate with a subset $\mathcal{N}_i \subset \mathcal{N}$ of neighbors. The graph $\mathcal{G} = \overline{\mathcal{N}} \times \mathcal{E}$, where $\overline{\mathcal{N}} = \mathcal{N} \cup \{s\}$ is the set of vertices and $\mathcal{E}$ is the set of edges, represents the possible communications between nodes and with the sink. $\mathcal{G}$ is assumed to be connected.

Time is slotted with a constant period T, and the clocks of all nodes are assumed to be perfectly synchronized. During the k-th time slot, only a subset $\mathcal{N}_{\mathrm{c},k}$ of nodes generates data packets to be transmitted to the sink. The number of data packets generated during the k-th time slot is a realization g_k of a Poisson random variable G_k. Each node in $\mathcal{N}_{\mathrm{c},k}$ may generate one or more data packets during the k-th time slot. A random linear network coding (RLNC [29])

protocol is put at work to deliver the g_k data packets to the sink s. All nodes participate to the packet delivery. Only data packets generated during the same time slot are network coded together. In what follows, one focuses on one specific time slot k and the subscript k is omitted to lighten notations.

The upper layer packets generated by the nodes in $\mathcal{N}_c$ are first intercepted by the NC shim yielding packets denoted $\mathbf{x}_1, \ldots, \mathbf{x}_g$. Each packet satisfies the same format. It consists of a constant-size header $\mathbf{h}_i$, followed by a payload $\mathbf{p}_i$ and a packet check sequence (checksum or hash) $\mathbf{c}_i = \mathbf{f}(\mathbf{h}_i, \mathbf{p}_i)$, where $\mathbf{f}$ is some known deterministic function. Each packet can be viewed as a row vector of ℓ_i elements of a Galois field $\mathbb{F}$, in which NC operations are performed. Prior to NC, packets are zero padded, so that they may be represented by vectors of $\ell = \max\{\ell_1, \ldots, \ell_g\}$ elements of $\mathbb{F}$. Thus, after zero-padding, the data packets can be stacked in a matrix $\mathbf{X} \in \mathbb{F}^{g \times \ell}$. Contrary to classical RLNC, no NC header is added to the data packets.

Assume that the sink s collects $g' \geqslant g$ packets containing random linear combinations of the source packets $\mathbf{x}_1, \ldots, \mathbf{x}_g$. These g' packets may be stacked to get a matrix $\mathbf{Y}' \in \mathbb{F}^{g' \times \ell}$ of received mixed data packets. The NC operations performed within the network may be represented by a matrix $\mathbf{A}' \in \mathbb{F}^{g' \times g}$, unknown to the sink s such that

$$\mathbf{Y}' = \mathbf{A}'\mathbf{X}. \tag{1}$$

One assumes that $\mathbf{Y}'$ is of full row rank g so that it is possible to extract sub-matrices $\mathbf{Y} \in \mathbb{F}^{g \times \ell}$ and $\mathbf{A} \in \mathbb{F}^{g \times g}$ of $\mathbf{Y}'$ and $\mathbf{A}'$, both of full rank g. Thus, (1) may be rewritten as

$$\mathbf{Y} = \mathbf{A}\mathbf{X}. \tag{2}$$

At the sink s, $\mathbf{Y}$ may be easily obtained by selecting g linearly independent rows of $\mathbf{Y}'$ in $\mathbb{F}$. Nevertheless, both $\mathbf{A}$ and $\mathbf{X}$ are unknown.

The aim of this paper is to propose a network decoding technique able to recover $\mathbf{X}$ from $\mathbf{Y}$ using the structure of the data packets in $\mathbf{X}$, without the help of NC headers. A way to solve this problem is to estimate a network decoding matrix $\mathbf{W} \in \mathbb{F}^{g \times g}$ such that

$$\widehat{\mathbf{X}} = \mathbf{W}\mathbf{Y}, \tag{3}$$

where $\widehat{\mathbf{X}}$ is equal to $\mathbf{X}$, possible up to a permutation of the rows of $\mathbf{X}$. For that purpose, one will use the fact that the packet headers, even if they do not contain the NC coefficients, may be used to estimate $\mathbf{A}$.

3. HAND

The main idea of HAND, inspired by PANDA [28], is to build a set of *admissible* candidate rows $\mathbf{w}$ of $\mathbf{W}$. Such rows should lead to estimated source packets with headers *compliant* with the structure of the protocols (as maintained by the NC shim). More specifically, some parts of the header may only take a small finite set of values which may be known in advance by the sink s. A row $\mathbf{w}$ of the network decoding matrix is admissible only if the part of the header of the decoded packet $\mathbf{w}\mathbf{Y}$ belongs to the set of admissible values.

3.1. Set of known potential subheaders

Let ℓ_h be the number of components in $\mathbb{F}$ of packet headers. The header $\mathbf{h}_i$ of the packet $\mathbf{x}_i$ belongs thus to $\mathbb{F}^{\ell_h}$. Nevertheless, in general, due to the constraints imposed by the protocol used to deliver data packets, only a very small subset of values in $\mathbb{F}^{\ell_h}$ may be taken by $\mathbf{h}_i$. Assume that the components of $\mathbf{h}_i$ may be partitioned into two sub-vectors $\mathbf{k}_i$ of ℓ_k *potentially known* components and $\mathbf{o}_i$ of ℓ_o *other* components. The sub-vectors $\mathbf{k}_i$, $i = 1, \ldots, g$ contain components of the header that

1. are *specific* to each packet generated during a given time slot: if one considers two different packets $\mathbf{x}_i$ and $\mathbf{x}_j$ generated during the same time slot, then $\mathbf{k}_i \neq \mathbf{k}_j$;
2. belong to a *small* subset $\mathcal{K} \subset \mathbb{F}^{\ell_k}$ of n_k possible values, all elements in $\mathcal{K}$ being *different* and *known* at the sink.

The set $\mathcal{K}$ is called the set of known potential subheaders. All elements of $\mathcal{K}$ are stacked in a matrix $\mathbf{K} \in \mathbb{F}^{n_k \times \ell_k}$.

Example 1. Assume that the application is generating data in UDP datagrams in an IPv6 network (using the socket API), at well defined ports. The NC shim will be able to identify and intercept such packets. If nodes are only allowed to generate a single data packet during a time slot, then each packet may be uniquely identified by the IPv6 source address field (bytes at positions 8 to 24) taken to be $\mathbf{k}_i$ in this case. The set of known potential subheaders contains the IPv6 addresses of all nodes in the network, known in advance by the sink. IPv6 addresses include 64-bit interface identifiers (EUI-64) [9], but these can be pseudo-random with privacy extensions [30, 31].

If nodes are allowed to generate several data packets per time slot, they would typically include also a counter along with the data. The set of known potential subheaders would then contain pairs of IPv6 addresses and possible values of the counters. Notice also, that when using IPv6 over IEEE 802.15.4 [9, 32], headers can be modified and compressed (for instance, network prefix removal, etc.). In our case, the NC shim would operate in conjunction with IPv6 header compression, and could perform any reversible transform as well (adding hash $\mathbf{c}_i$, whitening of some parts, ...).

3.2. Decoding method

Consider a matrix $\mathbf{P} \in \mathbb{F}^{\ell \times \ell_k}$ extracting the potentially known subheader from a data packets $\mathbf{x}_i$. One has

$$\mathbf{k}_i = \mathbf{x}_i\mathbf{P}. \tag{4}$$

$\mathbf{P}$ can also be used to get the g network-coded known potential subheaders from $\mathbf{Y}$ as

$$\mathbf{Z} = \mathbf{Y}\mathbf{P}, \tag{5}$$

where $\mathbf{Z} \in \mathbb{F}^{g \times \ell_k}$ is a sub-matrix of $\mathbf{Y}$ containing g linear combinations of vectors $\mathbf{k}_i \in \mathcal{K}$, $i = 1, \ldots, g$.

An admissible network decoding vector $\mathbf{w}$ has to be such that the estimated known potential subheader of the network-decoded packet $\mathbf{w}\mathbf{Y}$ must be one of the row vectors of $\mathbf{K}$, *i.e.*, there exists $j = 1, \ldots, n_k$ such that

$$(\mathbf{w}\mathbf{Y})\,\mathbf{P} = \mathbf{k}_j \tag{6}$$

or using (5), one should have

$$\mathbf{w}\mathbf{Z} = \mathbf{k}_j. \tag{7}$$

Let r be the rank of $\mathbf{Z}$. Several cases have now to be considered, depending on the values of g, ℓ_k, and r.

3.2.1. When $\ell_k > g \geqslant r$ or when $g \geqslant \ell_k > r$

In this case, using complete pivoting on the rows and columns of $\mathbf{Z}$, one may find two invertible matrices $\mathbf{U} \in \mathbb{F}^{g \times g}$ and $\mathbf{V} \in \mathbb{F}^{\ell_k \times \ell_k}$ such that

$$\mathbf{Z} = \mathbf{U} \begin{bmatrix} \mathbf{I}_{r \times r} & \mathbf{0}_{r \times (\ell_k - r)} \\ \mathbf{0}_{(g-r) \times r} & \mathbf{0}_{(g-r) \times (\ell_k - r)} \end{bmatrix} \mathbf{V}^{-1} \tag{8}$$

where

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ \mathbf{U}_{21} & \mathbf{U}_{22} \end{pmatrix} \text{ and } \mathbf{V}^{-1} = \begin{pmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{pmatrix}^{-1}.$$

For a given $\mathbf{k}_j \in \mathcal{K}$, one may write $\mathbf{w} = \begin{pmatrix} \mathbf{w}^{(1)} & \mathbf{w}^{(2)} \end{pmatrix}$ and $\mathbf{k}_j = \begin{pmatrix} \mathbf{k}_j^{(1)} & \mathbf{k}_j^{(2)} \end{pmatrix}$, where $\mathbf{w}^{(1)} \in \mathbb{F}^r$, $\mathbf{w}^{(2)} \in \mathbb{F}^{g-r}$, $\mathbf{k}_j^{(1)} \in \mathbb{F}^r$ and $\mathbf{k}_j^{(2)} \in \mathbb{F}^{\ell_\mathrm{k}-r}$. Then, combining (8) and (7) leads to the system of equations

$$\begin{cases} \mathbf{k}_j^{(1)}\mathbf{V}_{11} + \mathbf{k}_j^{(2)}\mathbf{V}_{21} &= \mathbf{w}^{(1)}\mathbf{U}_{11} + \mathbf{w}^{(2)}\mathbf{U}_{21}. \\ \mathbf{k}_j^{(1)}\mathbf{V}_{12} + \mathbf{k}_j^{(2)}\mathbf{V}_{22} &= \mathbf{0}_{1\times(\ell_\mathrm{k}-r)}. \end{cases} \quad (9)$$

If $\mathbf{k}_j$ is such that $\mathbf{k}_j^{(1)}\mathbf{V}_{12} + \mathbf{k}_j^{(2)}\mathbf{V}_{22} \neq \mathbf{0}$, then the system (9) cannot admit a solution and there is no $\mathbf{w}$ such that (7) is satisfied for that $\mathbf{k}_j$. If $\mathbf{k}_j$ is such that $\mathbf{k}_j^{(1)}\mathbf{V}_{12} + \mathbf{k}_j^{(2)}\mathbf{V}_{22} = \mathbf{0}$, then, since $\mathbf{U}_{11}$ is invertible, one gets

$$\mathbf{w}^{(1)} = \left(\mathbf{k}_j^{(1)}\mathbf{V}_{11} + \mathbf{k}_j^{(2)}\mathbf{V}_{21} - \mathbf{w}^{(2)}\mathbf{U}_{21}\right)\mathbf{U}_{11}^{-1}. \quad (10)$$

When $r = g$, (10) boils down to

$$\mathbf{w} = \left(\mathbf{k}_j^{(1)}\mathbf{V}_{11} + \mathbf{k}_j^{(2)}\mathbf{V}_{21}\right)\mathbf{U}_{11}^{-1}. \quad (11)$$

For a given row $\mathbf{k}_j$ of $\mathbf{K}$, satisfying $\mathbf{k}_j^{(1)}\mathbf{V}_{12} + \mathbf{k}_j^{(2)}\mathbf{V}_{22} = \mathbf{0}$, one is thus able to deduce the expression of one or several associated candidate decoding vectors $\mathbf{w}$. When $\mathrm{rank}\,(\mathbf{Z}) = g$, a unique network decoding vector is deduced for that $\mathbf{k}_j$ using (11). When $\mathrm{rank}\,(\mathbf{Z}) = r < g$, $|\mathbb{F}|^{g-r}$ network decoding vectors may be deduced for that $\mathbf{k}_j$ using (10).

Candidate decoding vectors are then selected when the resulting decoded packet check sequence is valid.

3.2.2. When $g \geqslant \ell_k = r$

In this case, $\mathbf{Z}$ may be written as

$$\mathbf{Z} = \mathbf{U}\begin{bmatrix} \mathbf{I}_{r\times r} \\ \mathbf{0}_{(g-r)\times r} \end{bmatrix}\mathbf{V}^{-1} \quad (12)$$

where

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ \mathbf{U}_{21} & \mathbf{U}_{22} \end{pmatrix} \text{ and } \mathbf{V}^{-1} = (\mathbf{V}_{11})^{-1}.$$

For a given $\mathbf{k}_j \in \mathcal{K}$, one may write $\mathbf{w} = \begin{pmatrix} \mathbf{w}^{(1)} & \mathbf{w}^{(2)} \end{pmatrix}$, where $\mathbf{w}^{(1)} \in \mathbb{F}^r$ and $\mathbf{w}^{(2)} \in \mathbb{F}^{g-r}$. Then, combining (12) and (7) one obtains only the first equation of (9), with $\mathbf{V}_{21} = 0$ and $\mathbf{k}_j^{(1)} = \mathbf{k}_j$. For each $\mathbf{k}_j$, one is now able to deduce the expression of one or several associated candidate decoding vectors $\mathbf{w}$. When $\mathrm{rank}\,(\mathbf{Z}) = g$, a unique network decoding vector is deduced for each $\mathbf{k}_j$ using (11). When $\mathrm{rank}\,(\mathbf{Z}) = r < g$, $|\mathbb{F}|^{g-r}$ network decoding vectors are deduced for each $\mathbf{k}_j$ using (10), with $\mathbf{V}_{21} = 0$ and $\mathbf{k}_j^{(1)} = \mathbf{k}_j$.

Candidate decoding vectors are again only selected when the resulting decoded packet check sequence is valid.

3.3. Decoding algorithm

HAND is summarized in Algorithm 1.

First, the matrices $\mathbf{U}$ and $\mathbf{V}$ are obtained from $\mathbf{Z}$ using a complete pivoting. Candidate decoding vectors $\mathbf{w}$ are partitioned into $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$ when $g > \mathrm{rank}(\mathbf{Z})$.

Then, all $\mathbf{k}_j \in \mathbf{K}$ are considered. When $\ell_\mathrm{k} > \mathrm{rank}(\mathbf{Z})$, some $\mathbf{k}_j$ may be eliminated using the second set of equations in (9).

In the second loop, which complexity depends on the size of the Galois field and on the difference between g and r, for each possible value of $\mathbf{w}^{(2)}$, $\mathbf{w}^{(1)}$ is evaluated using (10). Then $\mathbf{w}$ is obtained by combining $\mathbf{w}^{(1)}$ and $\mathbf{w}^{(2)}$. Using the known potential subheaders, the search space for $\mathbf{w}$ is thus decreased from q^g to $\overline{n}_\mathrm{K} \times q^{g-r}$, where $\overline{n}_\mathrm{K}$ is the number of valid $\mathbf{k}_j$, and q is the size of the considered Galois field. Thus, an enormous reduction of the search space is reached, especially when $g = r$.

Finally, the estimated data packet $\widehat{\mathbf{x}}$ is given by $\widehat{\mathbf{x}} = \mathbf{w}\mathbf{Y}$, and the check sequence is used to verify the validity of $\widehat{\mathbf{x}}$. All validated estimated packets are then kept in the set $\mathcal{X}$. The decoding is error-free when $\mathcal{X}$ contains only g packets.

Algorithm 1 Header-Assisted Network Decoding

function $\mathcal{X} = \mathrm{HAND}(\mathbf{Y}, \mathbf{K})$
  $\mathcal{X} \leftarrow \varnothing$; $\mathbf{Z} \leftarrow \mathbf{YP}$;
  $(\mathbf{U}, \mathbf{V}, \mathbf{r}) \leftarrow \mathrm{pivot}(\mathbf{Z})$;
  for each $\mathbf{k}_j \in \mathbf{K}$ **do**
    if $\ell_\mathrm{k} > r$ **and** $k_j^{(1)}\mathbf{V}_{12} + k_j^{(2)}\mathbf{V}_{22} \neq \mathbf{0}$ **then**
      continue;
    end if
    if $g > r$ **then**
      for all $\mathbf{w}^{(2)} \in \mathbb{F}_q^{g-r}$ **do**
        $\mathbf{w}^{(1)} \leftarrow (\mathbf{k}_j^{(1)}\mathbf{V}_{11} + \mathbf{k}_j^{(2)}\mathbf{V}_{21} - \mathbf{w}^{(2)}\mathbf{U}_{21})\mathbf{U}_{11}^{-1}$;
        $\widehat{\mathbf{x}} \leftarrow \mathbf{wY}$; $(\widehat{\mathbf{h}}, \widehat{\mathbf{p}}, \widehat{\mathbf{c}}) \leftarrow \widehat{\mathbf{x}}$;
        if $\widehat{\mathbf{c}} = \mathbf{f}(\widehat{\mathbf{h}}, \widehat{\mathbf{p}})$ **then** ▷ check sequence
          $\mathcal{X} \leftarrow \mathcal{X} \cup \{\widehat{\mathbf{x}}\}$;
        end if
      end for
    else
      $\widehat{\mathbf{x}} \leftarrow (\mathbf{k}_j^{(1)}\mathbf{V}_{11} + \mathbf{k}_j^{(2)}\mathbf{V}_{21})\mathbf{U}_{11}^{-1}\mathbf{Y}$; $(\widehat{\mathbf{h}}, \widehat{\mathbf{p}}, \widehat{\mathbf{c}}) \leftarrow \widehat{\mathbf{x}}$;
      if $\widehat{\mathbf{c}} = \mathbf{f}(\widehat{\mathbf{h}}, \widehat{\mathbf{p}})$ **then** ▷ check sequence
        $\mathcal{X} \leftarrow \mathcal{X} \cup \{\widehat{\mathbf{x}}\}$;
      end if
    end if
  end for
end function

4. SIMULATION RESULTS

To illustrate the performance of HAND, one considers synthetic source packets assumed to be NC between the IP and MAC layer. The fields of a typical packet exploited for decoding by HAND are
- a one-byte known field constant for all packets,
- a four-byte randomly generated IP-address, different for each node,
- a two-byte field containing the packet counter of each node,
- a four-byte checksum,
- $\ell - 11$ uniformly distributed bytes representing the other components of the header and the payload.

In each scenario, a random connected network of $N = 100$ nodes and one sink is generated. Then, g packets are randomly generated by a subset $\mathcal{N}_\mathrm{c}$ source nodes in such a way that no source node generates more than $\overline{n}$ packet during the same time slot. Transmission of network-coded packets is then performed through the network until the sink is able to collect g' network-coded packets gath-

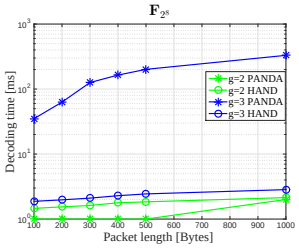

Fig. 1. Decoding time as a function of the packet length considering PANDA and HAND; NC is in $\mathbb{F}_{2^8}$ with $g = 2$ and $g = 3$

ered in a matrix $\mathbf{Y}'$ such that rank $(\mathbf{Y}') = g$, from which a submatrix $\mathbf{Y}$ of rank g can be extracted.

One assumes that the sink has been able to perfectly decode the packets up to the considered time slot. It is thus able to build the matrix $\mathbf{K} \in \mathbb{F}^{n_k \times \ell_k}$, where $n_k = N\overline{n}$ and ℓ_k depends on the size of Galois field used to perform NC operations. Decoding using HAND may then be performed from $\mathbf{Y}$ and $\mathbf{K}$.

HAND has been implemented in C++ using NTL[33] and Visual Studio 2010 on a SONY VAIO with Intel Core i3-2350M at 2.3 GHz with 4 GB RAM running Windows 7. Results are averaged over 100 random network and packet realizations.

HAND is first compared to the PANDA [28] decoding algorithm, in which (*i*) all possible decoding vectors $\mathbf{w}$ are generated, and (*ii*) only those $\mathbf{w}$ leading to a decoded packet compliant with the protocol are kept. Figure 1 shows the decoding time as a function of the packet length when NC operations are performed in $\mathbb{F}_{2^8}$. PANDA performs better than HAND when $g = 2$, but much worse than HAND when $g = 3$. This result is mainly due to the exponential complexity of PANDA. More than $g = 3$ network-coded packets are difficult to decode using PANDA in a reasonable amount of time.

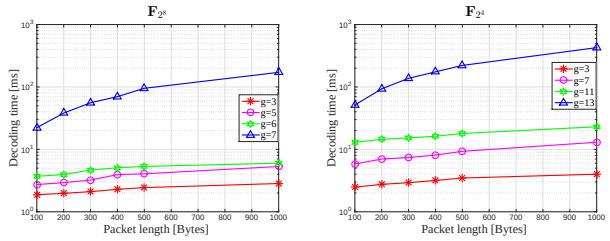

Fig. 2. Decoding time with HAND as a function of the packet length when NC is in $\mathbb{F}_{2^8}$ (left) and $\mathbb{F}_{2^4}$ (right)

Figure 2 (left) illustrates the decoding time with HAND when NC is in $\mathbb{F}_{2^8}$. Considering the header fields that may be exploited by HAND, one has $\ell_k = 7$. For $g < \ell_k$, rank $(\mathbf{Z})$ has a high probability to be g. Thus, many $\mathbf{k}_j$ can be eliminated using the second line of (9) and the decoding vectors $\mathbf{w}$ are obtained directly from (11). In all simulations, no decoding error is observed. Moreover, as seen in Figure 2, all source packets are correctly reconstructed in less than 10 ms even for packets of 1000 bytes. When $g = \ell_k$, rank $(\mathbf{Z})$ has a high probability to be g, but all $\mathbf{k}_j$ have to be considered, and in some cases, several $\widehat{\mathbf{w}}$ have to be considered for each $\mathbf{k}_j$. The verification of the check sequence has to be performed for each candidate $\widehat{\mathbf{w}}$. Again no decoding error is observed. The computing time has increased, but remains less than 100 ms in most of the cases.

The performance of HAND when NC is in $\mathbb{F}_{2^4}$ and $\mathbb{F}_{2^2}$ are given

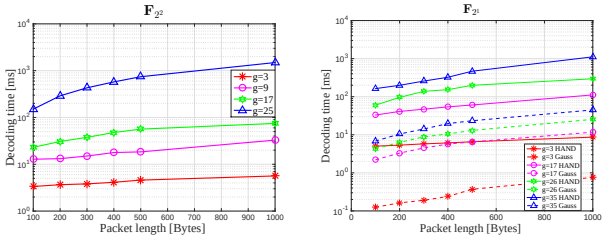

Fig. 3. Decoding time with HAND as a function of the packet length when network coding is in $\mathbb{F}_{2^2}$ (left) and $\mathbb{F}_2$ (right); decoding times with classical Gaussian elimination are also presented in the case $\mathbb{F}_2$

respectively in Figure 2 (right) and Figure 3 (left). In those cases, $\ell_k = 14$ and $\ell_k = 28$ respectively. Compared to the results shown in Figure 2, much larger values of g can be considered. When NC operations are in $\mathbb{F}_{2^2}$, $g = 17$ network-coded packets can be decoded without error using HAND in less than 100 ms. This is mainly due to the fact that the rank of $\mathbf{Z}$, upper-bounded by ℓ_k, remains close to g for larger values of g when small Galois fields are considered. For example, when $g = 17$, the average rank of $\mathbf{Z}$ is 15.3. As a consequence, HAND is able to eliminate candidate $\mathbf{k}_j$ efficiently, and avoids the enumeration steps in its second loop.

Finally, the performance of HAND is compared to that of a standard network decoding algorithm by Gaussian elimination when NC operations are performed in $\mathbb{F}_2$. In the latter case, NC headers have been supplemented to each packet to identify uniquely each packet generated by the sources, see Figure 3 (right). When considering packets of 500 bytes, 26 packets are decoded in less than 200 ms, while classical network decoding requires about 15 ms.

More generally, with HAND, the decoding time is about one order of magnitude larger than that of classical network decoding. Nevertheless, the absence of NC header, and of the need for an agreement among nodes for a global numbering (e.g. as necessary for encoding vectors of PNC) makes HAND a viable alternative approach to classical NC for data collection and recovery in sensor networks.

5. CONCLUSIONS

This paper considers the problem of data collection in a sensor network using NC. It proposes a decoding approach, called HAND, which does not require source packets to be supplemented with NC headers, classically used to decode the network-coded packets. HAND exploits the structure imposed by the communication protocol on the packet headers to estimate the original source packets from the received packets. This idea, previously proposed in PANDA, has been refined in HAND, where the structure of the received matrix is better exploited, to avoid the combinatorial search performed by PANDA, and reduces significantly the search space for candidate decoding vectors.

As a consequence, considering NC operations in $\mathbb{F}_2$, generations of 26 packets of 100 bytes can be network decoded in less than 60 ms. Such generation sizes are out of reach of alternative techniques based on finite field independent component analysis or based on PANDA. Avoiding the introduction of a NC header reduces the packet length, but more importantly, avoids the coordination step among nodes in order to ensure that different NC headers for each packet in the same generation are generated. This is particularly important in application where the churn rate is high, such as mobile crowdsensing.

6. REFERENCES

[1] Chunsheng Zhu, Lei Shu, Takahiro Hara, Lei Wang, Shojiro Nishio, and Laurence T. Yang, "A survey on communication and data management issues in mobile sensor networks," *Wireless Communications and Mobile Computing*, vol. 14, no. 1, pp. 19–36, 2014.

[2] C. Borcea, M. Talasila, and R. Curtmola, *Mobile Crowdsensing*, Chapman and Hall/CRC, 2016.

[3] W. Zamora, C. T. Calafate, J.-C. Cano, and P. Manzoni, "A survey on smartphone-based crowdsensing solutions," *Mobile Information Systems*, vol. 2016, pp. 26, 2016.

[4] Anum Ali, Ghalib A. Shah, Muhammad Omer Farooq, and Usman Ghani, "Technologies and challenges in developing machine-to-machine applications: A survey," *Journal of Network and Computer Applications*, vol. 83, no. Supplement C, pp. 124 – 139, 2017.

[5] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6tisch: deterministic ip-enabled industrial internet (of things)," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 36–41, December 2014.

[6] Thread Group, "Thread Specification 1.1.1," Tech. Rep., Feb. 2017.

[7] K. Mochizuki, K. Obata, K. Mizutani, and H. Harada, "Development and field experiment of wide area Wi-SUN system based on IEEE 802.15.4g," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Dec 2016, pp. 76–81.

[8] IEEE, "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," Tech. Rep., 2011.

[9] Gabriel Montenegro, Jonathan Hui, David Culler, and Nandakishore Kushalnagar, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944, Sept. 2007.

[10] J. N. Al-Karaki and A. E. Kamal, "Routing in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6–28, 2004.

[11] M. H. Anisi, A. H. Abdullah, S. A. Razak, and M. A. Ngadi, "An overview of data routing approaches for wireless sensor networks," *Sensors*, vol. 12, no. 4, pp. 3964–3996, 2014.

[12] Roger Alexander, Anders Brandt, JP Vasseur, Jonathan Hui, Kris Pister, Pascal Thubert, P Levis, Rene Struik, Richard Kelsey, and Tim Winter, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, Mar. 2012.

[13] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," in *Proc. IEEE INFOCOM*, April 2008.

[14] F. Bassi, L. Chao, L. Iwaza, and M. Kieffer, "Compressive linear network coding for efficient data collection in wireless sensor networks," in *Proc. EUSIPCO*, Bucharest, Romania, 2012, pp. 1–5.

[15] R. R. Rout and S. K. Ghosh, "Enhancement of lifetime using duty cycle and network coding in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 656–667, February 2013.

[16] Shabbir Ahmed and Salil S. Kanhere, "Hubcode: hub-based forwarding using network coding in delay tolerant networks," *Wireless Communications and Mobile Computing*, vol. 13, no. 9, pp. 828–846, 2013.

[17] Lorenzo Keller, Emre Atsan, Katerina Argyraki, and Christina Fragouli, "Sensecode: Network coding for reliable sensor networks," *ACM Trans. Sen. Netw.*, vol. 9, no. 2, pp. 25:1–25:20, Apr. 2013.

[18] Christina Fragouli, Jörg Widmer, and Jean-Yves Le Boudec, "Efficient broadcasting using network coding," *IEEE/ACM Trans. Netw.*, vol. 16, no. 2, pp. 450–463, Apr. 2008.

[19] S. Katti, H. Rahul, Wenjun Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 3, pp. 497 –510, june 2008.

[20] Szymon Chachulski, Michael Jennings, Sachin Katti, and Dina Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, New York, NY, USA, 2007, SIGCOMM '07, pp. 169–180, ACM.

[21] P.A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. of Allerton Conf. on Commun. Control and Comput.*, Oct. 2003.

[22] M. Jafari, L. Keller, C. Fragouli, and K. Argyraki, "Compressed network coding vectors," in *Proc. of IEEE Int. Symp. Inf. Theory*, June 2009, pp. 109–113.

[23] N. Thomos and P. Frossard, "Toward one symbol network coding vectors," *IEEE Commun. Lett.*, vol. 16, no. 11, pp. 1860–1863, 2012.

[24] J. Hansen, J. Krigslund, D. E. Lucani, and F. H. P. Fitzek, "Sub-transport layer coding: A simple network coding shim for ip traffic," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, Sept 2014, pp. 1–5.

[25] A. Yeredor, "Independent component analysis over galois fields of prime order," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5342–5359, Aug. 2011.

[26] H. W. Gutch, P. Gruber, A. Yeredor, and F. J. Theis, "ICA over finite field—separability and algorithms," *Signal Proc. (Elsevier Science)*, vol. 92, no. 8, pp. 1796–1808, 2012.

[27] I.-D. Nemoianu, C. Greco, M. Cagnazzo, and B. Pesquet-Popescu, "On a hashing-based enhancement of source separation algorithms over finite fields for network coding applications," *IEEE Trans. Multimedia*, (Accepted for publication).

[28] C. Greco, M. Kieffer, C. Adjih, and B. Pesquet-Popescu, "Panda: a protocol-assisted network decoding algorithm," in *Proc. of IEEE International Symposium on Network Coding (NETCOD)*, 2014.

[29] T. Ho, M. Médard, J. Shi, M. Effros, and D.R. Karger, "On randomized network coding," in *Proc. of IEEE Int. Symp. Inf. Theory*, June 2003, pp. 11–20.

[30] T. Narten, R. Draves, and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6," RFC 4941 (Draft Standard), Sept. 2007.

[31] F. Gont, "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)," RFC 7217 (Proposed Standard), Apr. 2014.

[32] Pascal Thubert and Jonathan Hui, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks," RFC 6282, Sept. 2011.

[33] Victor Shoup, "NTL: Number Theory library," 2001.