Probabilistic Network Coding for Reliable Wireless Sensor Networks

Eman AL-Hawri[(⊠)], Faroq AL-Tam, Noelia Correia,
and Alvaro Barradas

CEOT, University of Algarve, 8005-139 Faro, Portugal
{esalhawri, ftam, ncorreia, abarra}@ualg.pt

Abstract. Energy consumption and packet loss are common concerns in wireless sensor networks. However, designing protocols to minimize both metrics becomes very challenging because these are competing goals. In this paper, a data dissemination approach is proposed that uses probabilistic network coding to achieve a balance between energy consumption and packet loss. A subset of the network nodes, forming a directed acyclic graph, is selected to perform the encoding process under a certain probability. These nodes are the ones responsible for data delivery towards the gateways. The encoding probability is based on the kind of packet received: the more innovative the packet is, the higher the probability of going through the encoding process. The results show that random linear network coding, under such conditions, is able to adequately minimize energy consumption and packet drops, while being a very practical solution.

Keywords: Wireless sensor network · Network coding · Reliability · Energy saving

1 Introduction

Wireless sensor networks (WSNs) are used in many applications and are now a key element in the increasingly growing Internet of Things. A WSN is a network of spatially distributed sensing devices, usually of low cost, that may have self-organizing ability, or not, and may vary in their features and capabilities. These are able to sense their surrounding, communicate with their neighbors, and send their observations to a sink/gateway where the observed data is analyzed, processed and sent to the end user.

WSNs are usually considered constrained networks because nodes are equipped with restricted processing, memory, and energy capabilities [1]. Such resource limitations make the design of high Quality of Service (QoS) and energy-efficient applications a very challenging task. In fact, most methods and routing protocols try to solve this tradeoff between achieving QoS while using as little energy as possible. To develop applications with reasonable long lifetime, energy-saving policies should be followed [2]. Here in this article a data dissemination protocol is proposed that improves reliability, through packet loss minimization, while having energy-saving concerns for long application lifetimes.

The reminder of this article is organized as follows: Sect. 2 provides the related work while Sect. 3 presents the contribution of this article to life improvement. Section 4 addresses the network coding principle. Section 5 details the proposed method

while Sect. 6 discusses the results analysis. Finally, Sect. 7 provides the conclusions and future work.

2   Related Work

Network coding (NC) has been explored for different types of networks, including wired, wireless, sensor, and mobile. Each type of network applies NC to achieve a certain goal, which may relate to throughput increase, security improvement, reliability (decrease of packet drops), and energy saving.

In [3], dynamic network coding is used in multi-source systems that organize nodes as peers to form a P2P network. In these systems, a coordinator server is used to assign the roles for peers, ensuring that the NC process is accomplished successfully. The paper employs dynamic NC, where all peers in the P2P distributed system participate in the encoding/decoding process. The NetCoder in [4] mixes transmit diversity and random linear network coding techniques for WSN reliability purposes. Inter-flow Network Coding based Opportunistic Routing (INCOR) in [5] incorporates both opportunistic routing and inter-flow network coding to increase the performance of WSNs.

CodeDrip, proposed in [6], is a data dissemination protocol that utilizes network coding to achieve reliability, energy saving, and to increase the data dissemination speed in WSNs. CodeDrip uses a probability to decide whether to send the packet or to combine it with other messages for sending. To combine packets, binary coding with finite field $\mathbb{F}_2$ (XOR) is used to reduce the overhead induced by the random linear coding. In [7], the adequate number and location of encoding nodes is planned, considering predefined failure scenarios. This means that the number and location of critical links must be previously identified, which may not be possible in dynamic environments.

The method proposed here can achieve reliability while reducing the energy consumed, and can be applied in dynamic networks.

3   Contribution to Life Improvement

Sensor nodes can now be deployed anywhere and in any number to cover any intended area, and WSNs have now many different applications [8, 9]. Sensor networks have mainly been used in:

- *Military applications*: This was one of the first applications of WSNs. Nowadays sensor networks play a significant role in military commands, communication, monitoring of either friend or enemy force, intelligence, and surveillance.
- *Health applications*: Sensor networks became an essential part of health care systems. Hospitals are equipped with numerous sensor networks that monitor patients and return their state and location. This can be critical for patients requiring constant monitoring. Additionally, life expectancy will increase in the coming years, and healthcare systems can be equipped with technology distributed in the surrounding area of aged people, for monitoring of their state without affecting their daily activities [10].

- *Industrial applications*: One of the fields getting most benefits from WSNs is the industry. There are applications in buildings, constructions, and bridge condition monitoring, and other. Furthermore, sensor nodes can monitor the condition of machines, providing in some cases automatic maintenance, very important in production processes. Another vital role of sensor networks is to monitor production performance [11].

In this paper we aim to achieve two goals: (*i*) help constrained networks to have longer life; (*ii*) provide reliable information to the interested users. Achieving these goals will contribute to improve the lives of those who depend on these networks.

4   Network Coding

Network coding allows network nodes to perform linear combinations on received packets, and route one or more combined packets. In linear network coding, packets are represented as symbols over finite field $\mathbb{F}(2^m)$ of size m and an encoding node performs a linear combination per symbol position as follows:

$$Y_k = \sum_{i=1}^{n} \alpha_i \times X_k^i \tag{1}$$

where the $\alpha_1, \alpha_2, \ldots, \alpha_n$ is a set of random coefficients, called encoding vector, and n is the number of packets to be encoded. For decoding, the receiver needs to wait until it has enough independent coded packets to be able to perform the decoding and retrieve the original packets. Technically speaking, if a receiver node has received sufficient coded symbols and encoding vectors (α_m, Y_m), it can solve the linear system of equations $X = \beta^{-1} \times Y$ to retrieve the original symbols, where β^{-1} is the encoding matrix that contains the coefficients used in the encoding process, [12].

a. Flows towards G1.   b. Flows towards G2.   c. Transmission with NC.
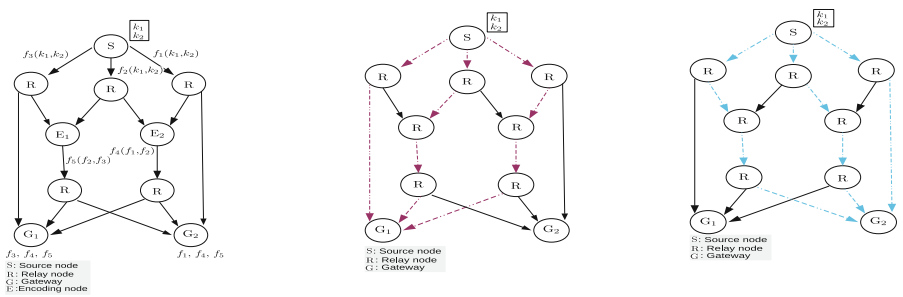
Fig. 1. Network coding robustness in multicast network

Network coding can achieve robustness against packet drops resulting from link failures, with no need for retransmission or rerouting. Figure 1 shows an example of a

multicast network with one source and two gateways. As shown in Fig. 1a and b, the maximum flow from source s to any gateway is 3, assuming the link capacity is one unit. Regardless of the output port through which packets k_1 and k_2 go out, in a link failure scenario one of the gateways may not receive both packets. In case of network coding, illustrated in Fig. 1c, the source and nodes E_1 and E_2 can linearly combine the received data. Consequently, when a link fails, regardless of its location, both gateways can decode and recover the lost data. This is so because each gateway will receive at least two out of three independent coded data units [13]. Thus, with network coding, a multicast rate of 2 is always achieved no matter which link fails. In this work, random linear network coding is used with finite field $\mathbb{F}_{2^8}$.

5   Proposed Method

The idea behind the proposed method is to: (*i*) reduce the number of transmissions by giving each packet an encoding probability; (*ii*) reduce the number of encoding nodes by dividing network nodes into three sets, each with a different role, as illustrated in Fig. 2:

- *Source nodes*: These nodes can send their data, and what they hear from their neighbors, to the collector nodes to which they are associated.
- *Collector nodes* (CN): Smallest possible set of nodes chosen to serve the other source nodes. Collectors receive data from their sources, encode them, and forward the coded packets to another CN or to the nearest gateway.
- *Gateways*: These nodes participate in the P2P overlay and their responsibility is to store the packets, coming from the wireless sensor network section, in the P2P where the decoding process can take place to retrieve the original data.
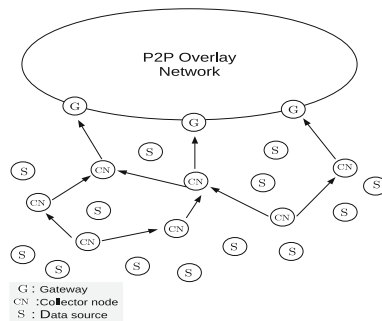

Fig. 2. Proposed architecture

This way, the problem can formally be described as follows: *given a network topology represented as a graph $\mathcal{G} = \{N, L\}$, where N and L are the set of nodes and links, respectively, and given a set of gateways $G \subset N$, the objective is to select the set of collectors C, sources S, and subsets of sources $Z_c \subset S, \forall c \in C$ (each subset connected to a $c \in C$) such that $|C|$ is minimized and reliability is ensured when one or more links fail.*

The proposed solution for this problem is detailed in Algorithm 1. The idea is to create a directed acyclic graph (DAG), from C and connections between its elements, to

ensure the flow of packets towards the gateways. This requires the definition of a topological order for C. A topological order of C is any τ such that if $(c_i, c_j) \in L$, then c_i precedes c_j in τ. That is, $\tau_{c_i} \leq \tau_{c_j}$. This DAG is used to forward coded packets toward gateways. More specifically, and as shown in Algorithm 1, the tuple $\{N, L, G\}$ is received and failing links, denoted by F, are selected according to a predefined failure probability. Then the following steps are followed:

- *Initialization*: Choose the smallest possible set C taking into account the connectivity of nodes. This criteria reduces the overhead caused by network coding. Build the DAG, towards the multiple gateways, using the nodes in C.
- *Dissemination*: Each source sends its packets, and what it hears from neighbors, to the CN to which it is associated. A CN calculates the occurrence of each packet in order to define its encoding probability, and accordingly applies the encoding process. Thus, the probability of encoding depends on the number of occurrences of each packet received by a CN. The rarest packets are more likely to experience an encoding process. The coded packets, generated from the encoding, are then sent through the DAG.
- *Regeneration*: Gateways store the received coded packets in the P2P overlay network, where the decoding can be performed to retrieve the original packets.

```
DAG Based Probabilistic network coding (DAG-PNC)
1. INPUT: N, L, G, F
2. Begin

      // Initialization
3.    select C
4.    build the DAG by defining a topology order τ for C
5.
      // Dissemination
6.    ∀s ∈ S:
7.       collect its own packet and packets heard from
              neighbors
8.    ∀s ∈ S, ∀l: source(l) = s ∧ l ∉ F:
9.       send collected packets to c ∈ C to which s is
      associated to
10.     ∀c ∈ C:
11.       receive packets from Z_c, k = |Z_c|
12.       calculate the occurrence of each packet
13.       define the probability of encoding
14.       generate k coded packets
15.       send coded packets to neighbors of c ∈ C
                (considering the DAG) or g ∈ G

      // Regeneration
16.  ∀g ∈ G: store received coded packets in P2P
```

Algorithm 1: The proposed method

6   Experimental Results

A set of random dense and sparse physical topologies of 30 nodes were generated and used to compare the performance of the proposed method with CodeDrip [6]. The proposed algorithm and network coding methods were all implemented in Matlab[1]. The parameters adopted in the tests are: (*i*) 4 gateways; (*ii*) link failure probability ranges from 0.05 to 0.3; (*iii*) network connectivity degree of 0.2 for sparse and 0.3 for dense network topologies.

   To evaluate both methods, random failure scenarios are generated. In each scenario a link failure probability is assumed, such that one or more links become inoperable. For a specific link failure probability, 20 folds are performed (failing links change randomly at each fold) and the average is used for reporting the results.

6.1   Analysis of Results

We consider two performance metrics to evaluate the performance of the methods: **reliability**, which is measured through the number of recovered packets, and the **number of transmissions**, which has impact on energy consumption.
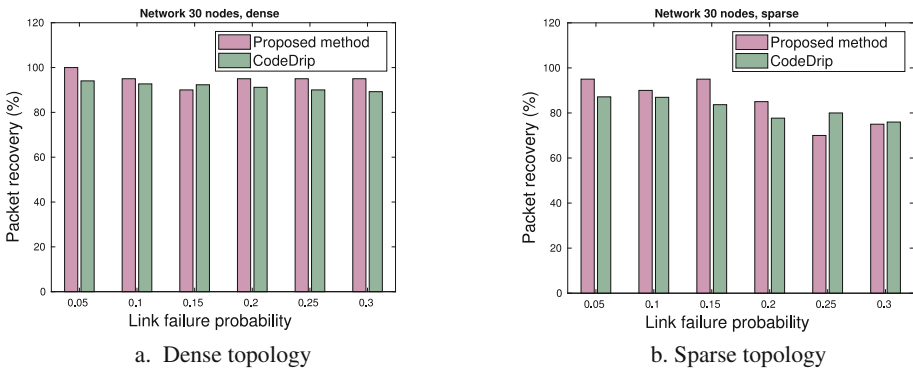

a. Dense topology   b. Sparse topology

Fig. 3. Reliability

   Figure 3 shows the results of packet recovery for both CodeDrip and the proposed method. Figure 3a shows the results for 30-node dense topologies while Fig. 3b shows the results for 30-node sparse topologies. For dense topologies we can see that the proposed method outperforms the CodeDrip and can retrieve more original packets using the decoding process. In sparse topologies, the proposed method also shows better performance, except for two cases of failure rates. This is because CodeDrip also sends original packets towards the gateways. On the other hand, in the proposed method if the decoding process is not accomplished, because enough linearly independent packets cannot reach the P2P successfully, there will be no recovered packet.

---

[1] MathWorks, Inc.

In general, the superior performance of the proposed method over CodeDrip can be related with: (*i*) better design of encoding nodes' location; (*ii*) overhearing performed by sources allows collector nodes to receive packets not only from their sources but also from sources associated with the neighbor collectors; (*iii*) performing the decoding process at the P2P overlay network and not at each gateway.

Even though CodeDrip nodes broadcast their original or coded packets through all paths, this does not ensure better results. The reason is that CodeDrip uses a probability to decide whether to send the packet or to combine it with other packet for sending. Also, XOR is used to combine packets, which means that some packets may not go through the coding process and some lost packets will not be recovered.
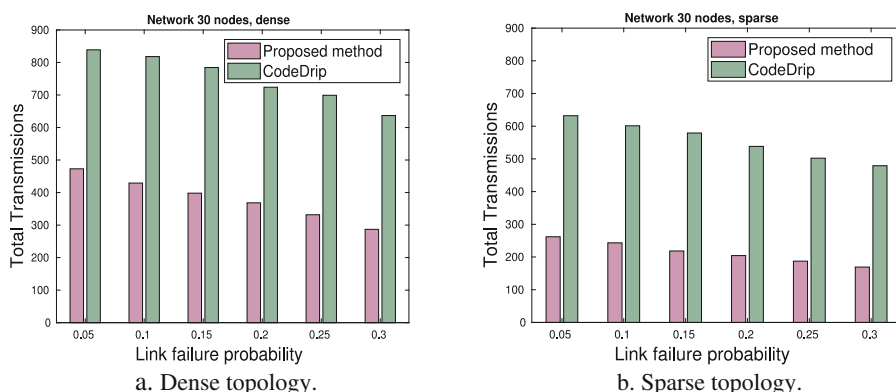

Fig. 4. Total transmissions.

Figure 4 shows the total number of transmissions generated by each method. This can help to decide which method consumes more energy. Both Fig. 4a and b show the large number of transmissions generated by CodeDrip. This is because all nodes in CodeDrip broadcast packets through all outgoing links, which results in a huge number of transmissions. In the proposed method, on the other hand, hearing is performed only by sources while collector nodes send a small number of coded packets. More specifically, the number of coded packets generated by a CN is the number of the sources associated with this CN.

7   Conclusions and Future Work

In this paper, a probabilistic network coding based WSN is proposed. This includes selecting which packets will be encoded and which nodes will be encoding nodes. The proposed method could recover the packet drops by applying decoding at the P2P overlay network. This means that using network coding in certain nodes and applying the encoding process on the more innovative packets can help in recovering lost packets with no need of a massive number of transmissions. This balancing between

reliability and energy consumption can be achieved by sending the adequate number of coded packets to overlay networks. Results show the effectiveness of the proposed approach when compared with the state of art. In the future, we will try to implement binary coding due to the low overhead resulted from such technique when compared to random linear coding.

References

1. Yick, J.A.: Wireless sensor network survey. Comput. Netw. **52**, 2292–2330 (2008)
2. Matin, M.A.: Overview of wireless sensor network. In: Wireless Sensor Networks-Technology and Protocols (2012)
3. de Asis Lopez-Fuentes, F.A.-A.: Dynamic network coding for collaborative multisource system. In: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 378–382. IEEE (2018)
4. Valle, O.T.: NetCoDer: a retransmission mechanism for WSNs based on cooperative relays and network coding. Sensors **16**, 766 (2016)
5. Hu, D.A.: INCOR: inter-flow network coding based opportunistic routing in wireless mesh networks. In: 2015 International Conference on Communications (ICC), pp. 3666–3671 (2015)
6. Junior, N.D.: CodeDrip: improving data dissemination for wireless sensor networks with network coding. Ad Hoc Netw. **54**, 42–52 (2017)
7. Al-Hawri, E.A.: Design of network coding based reliable sensor networks. Ad Hoc Netw. **91**, 101870 (2019)
8. Stojmenovic, I.: Handbook of Sensor Networks: Algorithms and Architectures, vol. 49. Wiley, Hoboken (2005)
9. Akyildiz, I.F.: Wireless sensor networks: a survey. Comput. Netw. **38**, 393–422 (2002)
10. Tennina, S.A.-V.: WSN4QoL: Wireless Sensor Networks for quality of life. In: 2013 IEEE 15th International Conference on e-Health Networking, pp. 277–279 (2013)
11. Erdelj, M.A.: Applications of industrial wireless sensor networks. In: Industrial Wireless Sensor Networks: Applications, Protocols, and Standards, pp. 1–22 (2013)
12. Goncalves, D.A.: Random linear network coding on programmable switches. In: 2019 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS), pp. 1–6 (2019)
13. Kamal, A.E.: Network coding-based protection. Opt. Switching Netw. **11**, 189–201 (2014)