

SR-Code: Smart Relay Network Coding for Data Collection for Wireless Sensor Networks

Zaher Merhi*, Oussama Tahan*, Bakri Abdul-Hay**, Rabih Rammal***
and Samih AbdulNabi*

*(*Department of Computer and Communication Engineering, Lebanese International University*)

** (*Department of Mechanical Engineering, Lebanese International University*)

*** (*Department of Electrical Engineering, Lebanese International University*)

ABSTRACT

Reliability in data collection for wireless sensor networks is one of the major problems in IoT applications. Sensor nodes are usually placed in harsh conditions where data communication is at risk of losing packets. Retransmissions are considered costly in terms of delay and power consumptions, especially that wireless sensor nodes are battery operated. In this context we introduce SR-Code, a novel network coding algorithm that achieves reliability in harsh conditions. SR-Code utilizes the XOR operator to code overheard packets. The targeted network topology is a 2-tier network where data loss can occur in all tiers. SR-Code utilizes bit addresses where each node is identified by a single bit in an address bit vector. Identifying packets and computing the cardinality of coded messages can be easily done using address bit vectors. SR-Code realizes redundancy as a function of overheard packet. SR-Code achieved a reliability factor of 75% when the number of packets lost was 100% of the original (un-coded) packet sent.

Keywords: IoT, Network Coding, Reliability, Wireless Sensor Networks

I. INTRODUCTION

With the evolution of technology, the internet of things (IoT) [1] is becoming a reality. Wireless sensor networks (WSN) play an integral part in most IoT applications [2] where a group of sensor nodes collaborate with each other to perform a specific task and report the outcome, through the cloud, to interested users. For example, in IoT based healthcare applications [3-5], a group of sensors collect vital signals from the patient and collaborate with each other to determine the health levels of its user. If a risk is detected, the collected data is sent to the patient's doctor to notify him accordingly. Reliability is a major concern in any wireless sensor network application. Typically, wireless sensor nodes are deployed in noisy environment where data is often corrupted or dropped due to weak signal strength or high congestion levels. For example, the IEEE 802.15.4 standard [6] which is the standard for WSN, operates in the unlicensed band which is usually crowded with appliances sharing the same band. Given that wireless sensor nodes are battery operated, retransmission can be costly in terms of network lifetime. However, one way of insuring reliability is to inject redundancy. Redundancy can be accomplished by retransmissions. However, basic retransmissions are too expensive in terms of latency and power consumption. Thus, a compromise must be taken in order to ensure high reliability taking into consideration limited power consumption.

Network coding [7] is a technique used to combine packets together into a single packet. The combination is usually done in two ways: either by performing a linear combination of the packets or by performing the XOR-operation. Benefits for network coding can be in reliability, storage, and security. Improved reliability can be clearly demonstrated in Fig. 1 which presents network coding using the XOR operator. We have 2 sources S1 and S2 and four receivers D1, D2, D3 and D4. By noting packet P1 as packet produced by S1 and P2 packet produced by S2 and receivers have to receive all packets, an intermediate node, R, can XOR both packets to create a coded packet and send it again to receivers that are capable to extract packet P2 having P1 and capable to extract packet P1 having packet P2. This will lead to an increase in reliability when there are packet losses.

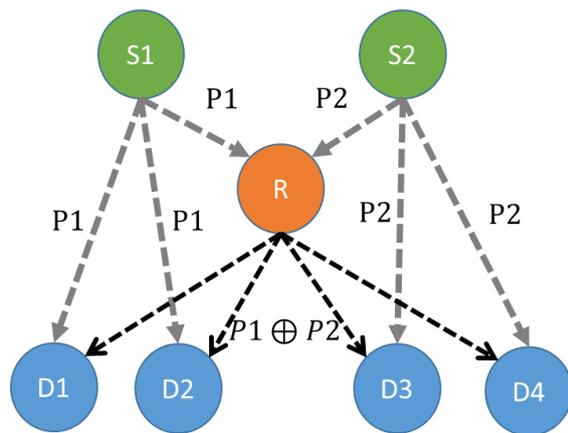


Figure 1: Network Coding using XOR

Network coding becomes challenging due to its complexity and the difficulty of integration with existing infrastructure. Adding a linear combination or logical combination such as XOR requires every node in the network to have the capability to generate such combinations or to solve linear equations by Gaussian elimination for example. Recent work suggest that network coding becomes too complex when dealing with Gaussian elimination causing a computational time in the TinyNode [8] which lead to additional memory requirements.

The way network coding is used can also be classified into three main categories depending on the type of combinations; those classifications are: Inter-session Network Coding, Intra-session Network coding and Joint Inter and Intra- Session Network Coding [9].

Inter-session Network Coding can reduce the number of transmissions when the packets from different source are coded together as shown in Fig. 2. When we code the packets from the same source, we are performing intra-session network coding. If the wireless medium have lossy links it is preferable to have inter-session and intra-session network coding together as joint process.

Reliability can also be achieved using linear combinations instead of the XOR operator however it is more costly in terms of complexity. Usually sensor nodes have limited computational capabilities which hinder them from performing complicated arithmetic operations. In this paper, we are proposing a network coding scheme called SR-Code for data collection protocols. Typically, in these applications, the sink node initiates round of collection where sensor nodes broadcasts sensed/processed information towards the sink. Without loss of generality, we assume a two-tier network with static routes. In networks where the sink is located several hops away from the sensor nodes and with dynamic routes, a collection based

routing protocol such as CTP [8] can be efficiently used to creates the routes. The main issue with using network coding for achieving reliability is that coding a large number of packets will lead to a large enough overhead that overcomes its advantages. SR-Code achieves reliability at low cost by utilizing the XOR-Operator and using bit vectors for addressing. The paper is organized as follows: Section II discusses the literature review, section III presents the system model. In Section IV we discuss SR-Code algorithm in details. Section V presents all experimental results. Finally Section VI is the conclusion and future work.

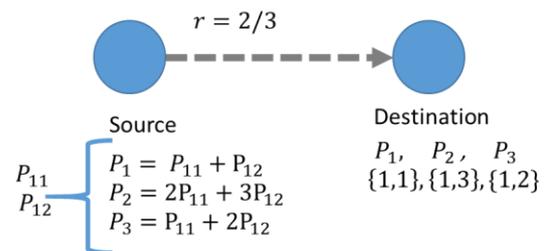


Figure 2: Network Coding using Linear Combinations (Inter Session)

II. LITERATURE REVIEW

The importance of packets movement in wireless sensor networks was a basic subject for most of the researchers in the past few years. The main focus of network design has been based on simplicity of operation and scalability. However, recent works shows also that effective power saving and physical resource management could be done by broadcasting with the appropriate coding schemes. The movement of data over the network was either considered in the data dissemination scheme or the data collection scheme, Hui and Culler in [10] introduced Deluge a reliable data dissemination protocol to overcome the propagation of large data objects using advertisement process of the most recent version of data object to neighbors that can hear it. Deluge is a well-known algorithm that solved many issues in data dissemination; however a considerable new protocol was introduced in [11] where 40% less packets were used in large networks than Deluge and load balancing was also taken into consideration even though it was not clearly verified in the algorithm. AdapCode used to disseminate data in the network using network coding [12] but in an adaptive way according to link quality while decoding was based on Gaussian elimination. To focus on data collection algorithms, Wang et al. in [13] introduce a new scheme in network coding called PNC (Partial Network Coding) trying to overcome the challenge of continuous data collection from sensors to server where data removal was available without decoding with same overhead

and performance as the conventional network coding and the focus was on obsolete data removal. SenseCode [8] was introduced as a collection protocol in WSN using NC (Network Coding) as many-to-one communication protocol in which the time was divided into rounds where sensor is responsible to send message to the sink. The main idea behind SenseCode was to adopt a systematic communication – a technique that reduces the coding vector overhead by applying coding on half transmitted packets.

III. SYSTEM MODEL

SR-Code is a network coding protocol that achieves reliability for data collection application for wireless sensor networks. We assume a two-tier network with static routes where each node communicates its packets to a relay. In turn, this relay will deliver received packets to the sink. We assume sparse networks where relays, in most cases, are far enough that they do not overhear each other as shown in Fig.4. This will add more complexity when achieving reliability since overheard packets by relays are major in recovering packets. We also assume that sensor nodes can communicate to one or more relays. In SR-Code, network coding is performed using the XOR operator. Simply, when a packet is received, the nodes will XOR the content with one or more packets that is available in its buffer. The resultant packet is then forwarded to the next hop. In order to identify the packets that were XOR-ed, the address of each packet is included in the header of the coded message. However, this poses a large overhead on the size of the coded packets which in some cases can exceed the size of the payload. To overcome this issue, SR-Code utilizes bit addresses where the bit location refers to the identity of the originating packets.

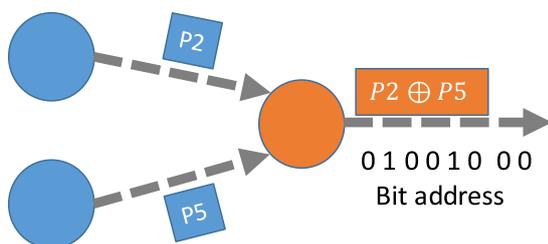


Figure 3: Binary Coding of Addresses

For an example, assume that the relay node received P2 and P5 packets and needs to code them together. Assume for the sake of simplicity, there are only 8 nodes in the network where a bit vector of size 8 will suffice as shown in Fig.3.

When coding is performed, SR-Code will set the corresponding bits that refer to the address of the originating packets. In this case, bit 2 and bit 5 will be set to indicate that the coded packet contains

information from node 2 and node 5. In larger networks a larger bit vector will be used. The advantage of using bit vectors is that it reduces the size of the header of coded packets which is a major concern when network coding is performed. To see the advantage of the address bit vector method, consider a network where a relay needs to send packets consisting of 32 coded packets. To identify these packets the addresses which are usually 16-bits are sent along the packet. Thus, the total overhead will be 64 bytes. In our case, and using a 64-bit vector which can identify up to 64 coded packets, an overhead of 4 bytes is introduced.

IV. SR-CODE DESIGN

The main idea behind SR-Code is to inject enough redundancy to achieve reliability without depleting the power reserve of the wireless sensor nodes. SR-Code specifies three roles for nodes, each with a different task. As shown in Fig 4, sensor nodes at the first tier will send recorded data and are able to code packets. Relay nodes are responsible for coding, decoding and forwarding packets. The sink node is responsible for un-coding packets.

Collection starts when the sink initiates a round for data collection. When sensor nodes receive the request from the sink, nodes will initiate data transfer by sending the un-coded packets, for example, Node 2 will send packet P2 to the relay node. Second, it waits for a pre-determined time to overhear neighboring packets. For each new un-coded packet received, the nodes will XOR its content with its list of un-coded packets one at a time.

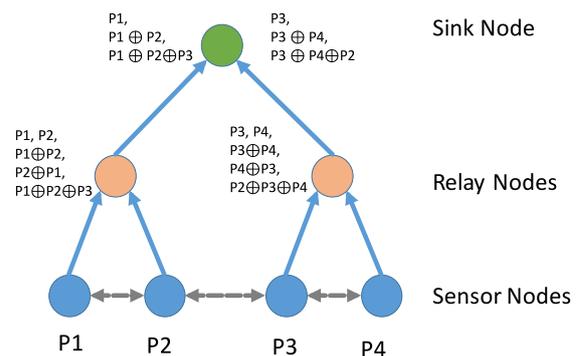


Figure 4: A 4-2-1 Wireless Sensor Network

Only un-coded packets are coded, as indicated above where sensor nodes do not perform decoding. For example, if node 2 overhears packets P3 and P1, it will subsequently send, to the relay node, P1 XOR P2 (if P2 is received first) and then P1 XOR P2 XOR P3. Moreover, it will set the corresponding bit addresses in the headers of each packet, as explained earlier. Typically, in the first packet the node will set bit location 1 and 2, and in

the second packet it will set bit location 1, 2, and 3. Coding is only done once at the sensor node level. The coding technique is presented in Algorithm 1.

Algorithm1 : Coding at Sensor Nodes

```

1: if !Round_Started() then
2:   wait_for_sinkpacket()
3: else
4:   startRound()
5:   RoundTimer.start()
6:   Address = 1<<i
7:   Pi.CODED = 0
8:   sendPacket(Pi)
9: end if
10: if ReceivedPacket(Pj) then
11:   if Packet_IsUncoded(Pj) then
12:     Pi = Pi XOR Pj
13:     Pi.CODED=1
14:     Address = 1<<i | 1<<j
15:     Forward(Pi)
16:   end if
17: if RoundTimer.fired() then
18:   endRound()
19: end if
    
```

Upon the reception of packets at the relay nodes, packets are categorized whether they are coded or un-coded, and stored accordingly in each buffer. The relay node will decode these packets in preparation for forwarding as shown in Algorithm 2. The aim of the decoding algorithm is to reach all packets in their un-coded original format. The decoding algorithm works as follows: The first step of decoding starts with un-coded packets. It will iterate over all un-coded packets one at a time. The algorithm checks if the un-coded packets exist in coded packets. This is done by a checking if the bit address at the location of the address of the un-coded packet is set to 1. If so, the packet is XOR-ed with the coded packets that succeeded the existence test. Moreover, it will clear the bit location in the bit address vector thus eliminating its existence in the coded packet. After the XOR operation is performed, the algorithm checks if the performed operation yielded a packet that is a singleton by checking its cardinality i.e. does not contain coded information.

Algorithm 2: Decoding at Relay Nodes

```

1: waitforRoundStart();
2: if ReceivedPacket(P) then
3:   if P.uncoded == 1 then
4:     UncodedBuffer.add(P)
5:   else
6:     CodedBuffer.add(P)
7:   endif
8: endif
9: while(AllPacketsUncoded() | k<n)
10: for i=1:size.UncodedBuffer
    
```

```

11: for j=1:size.CodedBuffer
12:   P = CodedBuffer(j)>
13:   if Cardinality(P) >1 then
14:     if UncodedBuffer(i).exists then
15:       P = P XOR UncodedBuffer(i)
16:       P.Address = P.Address XOR
17:         UncodedBuffer(i). Address
18:       if Cardinality(P) ==1 then
19:         UncodedBuffer.add(P)
20:       endif
21:     end if
22:   end for
23: end for
24: end while
    
```

The cardinality test is done by summing up the bits in the bit address vector. If the result is one, then we have reached the original un-coded packet and it is appended at the end of the un-coded buffer. Moreover, the number of un-coded packets is increased by one so that the algorithm uses the new un-coded packet in searching for the rest of the un-coded packets. In case of no packet loss, and after exhausting all options with un-coded packets, the algorithm checks packets with cardinality 2. SR-Code locates these packets by summing up the bits in the bit address vector. Furthermore, SR-Code also fetches packets with cardinality greater than two and tries to search for them in other packets. If a message is un-decodable at this point, it will be placed in a separate buffer awaiting further packets if any is received afterwards. Once decoding finishes, the relay will prepare packets to be forwarded. The relay will choose the first packet in the un-coded buffer and will forward it in un-coded format. The relay will then send only coded packets. The coding process will be as follows: The first coded packet sent will be the first un-coded packet XOR-ed with the second un-Coded packet in the un-coded buffer. The second packet will be the first packet XOR-ed with the second XOR-ed with the third, and so forth. To illustrate, consider a relay with 4 un-coded packets P1 to P4. The relay will send one un-coded packet and 3 coded packets as shown in Fig.5.

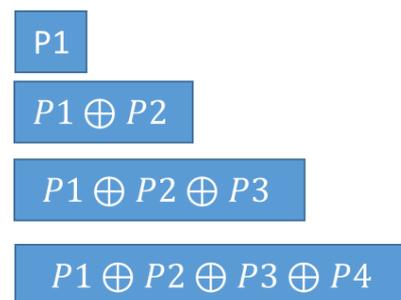


Figure 5: Coding process at the Relay

Finally the sink will receive the packets from all relays and will perform the decoding algorithm described earlier.

V. EXPERIMENTAL RESULTS

To verify the correctness of the proposed work, SR-CODE was simulated on three different networks. The networks consist of a single sink, 5 relays and we varied the number of sensor nodes from 10 to 20 as shown Figs. 6-8. Sensors belonging to the same relay are assumed to communicate with each other. Sensor nodes on the edges of relays are assumed to communicate with nodes belonging to neighboring relays. Only relays can communicate directly with the sink. Furthermore, relays do not overhear each other. This assumption adds more complexity since it will reduce redundancy and lowers reliability.

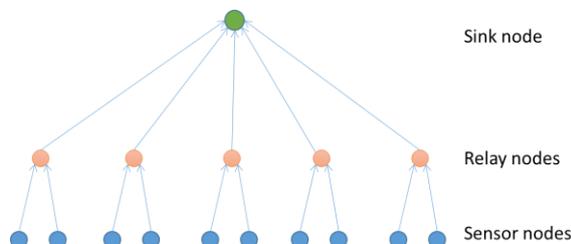


Figure 6: Network consisting of 10 sensor nodes

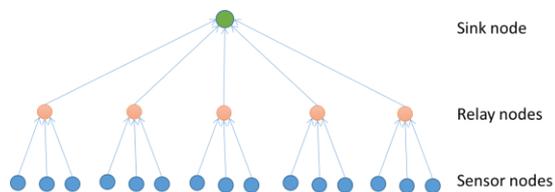


Figure 7: Network consisting of 15 sensor nodes

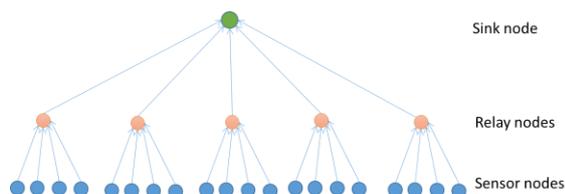


Figure 8: Network consisting of 20 sensor nodes

The simulation for all three networks was repeated 20 times. For each iteration, we varied the number of packet loss from 0% to 100% of the original packets sent by sensor nodes. The packet loss were randomized and not uniformed. That is, the packets that were eliminated can be belonging to any node and/or relay. As shown in Figs 9-11, SR-CODE could recover most packets even in harsh conditions where the error introduced 100%. The success ratio was up to 75% percent when the network consists of 10 nodes as shown in Fig. 9.

In Figs. 10 and 11, the performance deteriorates because the number of packets sent is less since the number of nodes per relay decreased. The reason behind the reduction of sent packets is that SR-CODE send packets based on overheard packets. Thus, when the number of nodes increase in the same hop, the number of overheard packets is reduced and in turn the number of sent packets. SR-Code only sends the original packet and coded packets from overheard neighbors. In our networks two one hop networks at the sensor node level are connected by only one node.

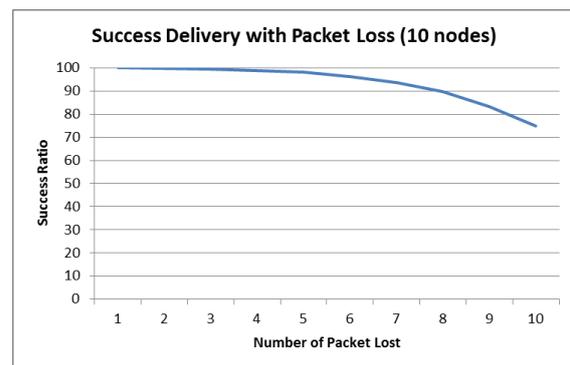


Figure 9: Simulation Results for a 10 nodes network

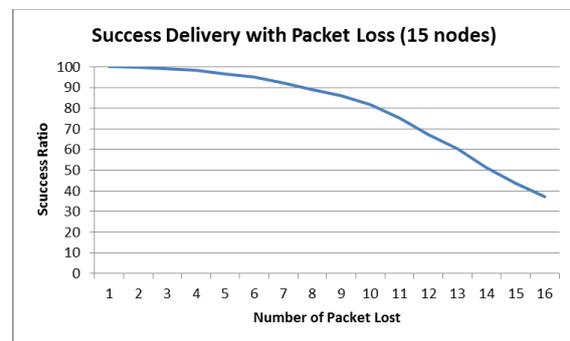


Figure 10: Simulation Results for a 10 nodes network

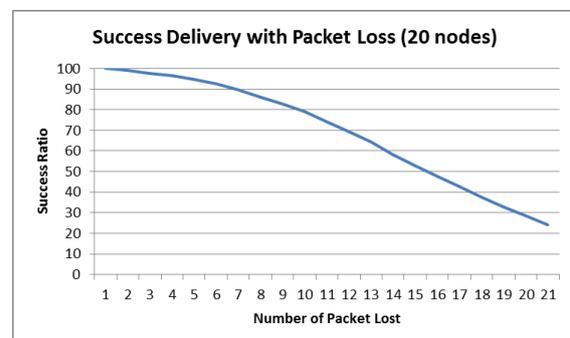


Figure 11: Simulation Results for a 10 nodes network

To demonstrate the effect of adding more nodes per one-hop neighborhood while keeping the degree of connectivity between neighbors to one, we

simulated a network of 8 nodes and 4 relays. In this simulation we measured the number of packet sent and the percentage of redundancy. The redundancy is calculated with respect to original packets sent. For an example, a redundancy of 200% means that the number of packets sent is twice the number of original packets. Consider Figs. 12 and 13, if the number of nodes is set to 20 while keeping connectivity with neighboring hops to one, the redundancy is 1.4. In this case if 100% of the packets are lost and the system will not be able to recover all packets. Thus, in harsh conditions, it is assumed that the number of nodes should be increased and more importantly the connectivity between neighboring nodes should be increased, thus achieving better reliability.

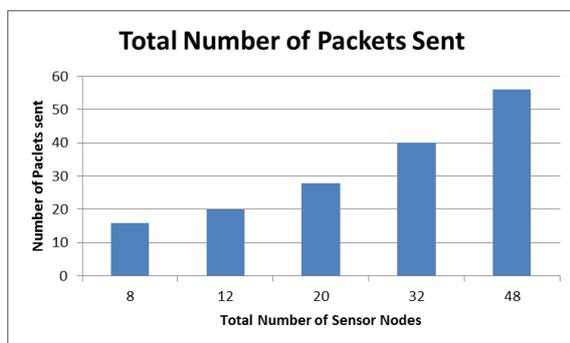


Figure 12: Total number of packets sent as a function of number of nodes

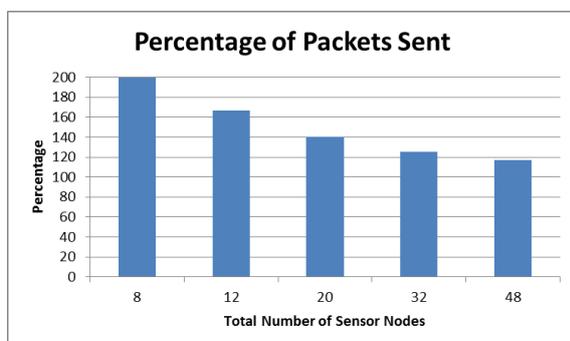


Figure 13: Percentage of packet sent with respect to the number of nodes

VI. CONCLUSION

In this work we introduced a novel network coding algorithm called SR-Code. The main purpose of this work is to achieve reliability in face of packet loss. The algorithm utilizes the XOR operator to code packets which have a low computational overhead. Furthermore, SR-Code uses bit addresses which lowers the amount of bytes sent per packet. Moreover, using bit addresses aids in identifying the number of packets included in each coded packet and allows to easily identifying which packets are included in the coded message. SR-Code achieved redundancy as a function of neighboring nodes that

are one hop away from them. Thus, it is common to have a redundancy factor less than 2. Simulations were conducted on three different networks where the number of sensor nodes is increased. The algorithm presented robustness even in harsh conditions where packet losses were 100% of the total number of original packets sent.

In our future work, we are planning on implementing SR-Code on actual hardware, where power consumption and reliability can be addressed in more details. Furthermore, we will be testing SR-Code on different scenarios and topologies.

REFERENCES

- [1] A S Abdul-Qawy et al., The Internet of Things (IoT): An Overview, *Int. Journal of Engineering Research and Applications*, ISSN: 2248-9622, Vol. 5, Issue 12, (Part - 2) December 2015, pp.71-82
- [2] Liu, Vincent Y., and Zhao Yu., Wireless Sensor Networks for Internet of things: a systematic review and classification. *Information Technology Journal* 12.16 (2013): 3581.
- [3] Islam, SM Riazul, et al. "The internet of things for health care: a comprehensive survey." *IEEE Access* 3 (2015): 678-708.
- [4] Rohokale, Vandana Milind, Neeli Rashmi Prasad, and Ramjee Prasad. "A cooperative Internet of Things (IoT) for rural healthcare monitoring and control." *Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*. IEEE, 2011.
- [5] Doukas, Charalampos, and Ilias Maglogiannis. "Bringing IoT and cloud computing towards pervasive healthcare." *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*. IEEE, 2012.
- [6] IEEE P802.15.4/D18, Draft Standard: Low Rate Wireless Personal Area Networks, Feb. 2003
- [7] Christina Fragouli, J.-Y. L. Network Coding: An Instant Primer CM SIGCOMM Computer Communication Review. *ACM SIGCOMM Computer Communication Review*, 6. (2006).
- [8] Keller, Lorenzo, et al., SenseCode: Network coding for reliable sensor networks, *ACM Transactions on Sensor Networks (TOSN)* 9.2 (2013): 25.
- [9] Ostovari, Pouya, Jie Wu, and Abdallah Khreishah. "Network coding techniques for wireless and sensor networks." *The Art of*

- Wireless Sensor Networks*. Springer Berlin Heidelberg, 2014. 129-162.
- [10] Gnawali, Omprakash, et al., CTP: An efficient, robust, and reliable collection tree protocol for wireless sensor networks, *ACM Transactions on Sensor Networks (TOSN)* 10.1 (2013): 16.
- [11] Jonathan W. Hui, D. C. (2004). The Dynamic Behavior of a Data Dissemination Protocol for Network Programming at Scale. *SenSys'04, November 3-5, 2004, Baltimore, Maryland, USA.*, 14.
- [12] Hou, I-H., et al. "Adapcode: Adaptive network coding for code updates in wireless sensor networks." *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE. IEEE*, 2008.
- [13] Wang, Dan, Qian Zhang, and Jiangchuan Liu. "Partial network coding: Theory and application for continuous sensor data collection." *200614th IEEE International Workshop on Quality of Service. IEEE*, 2006.