A new algorithm and routing protocol based on convolutional codes using TCNet: Trellis Coded Network

Diogo F. Lima Filho and José R. Amazonas

Escola Politécnica of the University of São Paulo, Brazil
dioferlima.usp@gmail.com,jra@lcs.poli.usp.br

Abstract. Currently emerging technologies have taken advantage of the possibility of communication with the *World Wide Web* to expand to all applications of this technology, among them the Internet of Things – IoT. This research, studies to implement a new algorithm and protocol that allows routing of data collected in micro sensors in ad hoc networks scenarios with randomly distributed sensors in adverse areas. This work proposes the implementation of new packets forwarding algorithms using the concept of convolutional codes. The results obtained by means of extensive simulations show gains in terms of latency and energy consumption reduction compared to the AODV protocol. The implementation complexity is extremely low and compatible with the few hardware resources usually available in the elements of a wireless sensor network (WSN). This research uses Finite State Machine (FSM) to get the networks nodes, obtained from settings with Mealy machine – (MM) due to the low complexity of ("XOR" gates and "shift registers"), eliminating the use of any routing tables by means of Trellis decoding. The sequence of states of the FSM will be interpreted as a network route to be traveled by a transmission frame. The route can be chosen based on criteria of Quality of Service (Qos) – aware routing protocols.

Keywords: Sensor Network. Convolutional codes. Wireless. Trellis Decoder. Protocols. Encoded networks. Finite State Machine.

1 Introduction

Wireless sensor networks (WSNs) have evolved from the idea that small wireless sensors can be used to collect information from the physical environment in a large number of situations. Early work in WSNs were developed by Defense Advanced Research Projects Agency (DARPA), so called Smart Dust, based on microelectromechanical systems (MEMS), devices able to detect light, temperature, vibration, magnetism or chemicals, with embedded processing and capable of transmitting wireless data. Currently emerging technologies have taken advantage of the possibility of communication with the World Wide Web to expand to all applications of this technology, among them the Internet of Things – IoT.

Although they were developed several hardware devices by the research community on WSN, there is an effort led by Internet Engineering Task Force (IETF) represented by the Working Group: Routing Over Low-Power and Lossy Networks (ROLL) [1], responsible for standardization and specifying the IP protocol, proposed several criteria to satisfy specific routing requirements. The world of WSNs is no exception: the use of an open standard such as IP is crucial and is necessary to build a scalable architecture for the next Internet and other IP networks [2]. This work proposes the implementation of new packets forwarding algorithms using the concept of convolutional codes. The results obtained by means of extensive simulations show gains in terms of latency and energy consumption reduction compared to the AODV protocol. The implementation complexity is extremely low and compatible with the few hardware resources usually available in the elements of a wireless sensor network (WSN).

1.1 Related work

The constant searches to obtain protocols that support the WSNs networks are challenging due to the dynamic characteristics of these networks [3]. The attempts to adapt the routing protocols of the infrastructure networks [4], [5], [6] to the cases of ad hoc networks are often inconsistent to satisfy the such aspects: frequent changes in topologies, poor links, restricted bandwidth, and restrictions on power sources.

Several criteria in protocol specification have been studied by the ROLL Working Group and a specific protocol called attention in this research, because it is one of the most used protocols in WSNs: Ad-hoc On Demand Vector Routing (AODV) [7]. As conclusion, the Working Group defined the following set of requirements for emerging protocols to meet WSNs needs: scalability; routing metrics; network stability; degree of constraints and application aware routing as challenging, because of the high degree of network constraints.

The MPLS concept are also required for providing service to the network (IP) layer, could be a good option to implement QoS, in which routing algorithms are not scalable. Although the MPLS employs label tables that are added to datagrams. These tables are used by intermediate nodes to identify and switch the packets. However, this procedure is not compatible with the resources available in WSN nodes.

1.2 Proposal and contribution

This work proposes a new algorithm and routing protocol based on convolutional codes to enable QoS-aware protocols in WSNs using the concept of a “Trellis Coded Network”- (TCNet) [8], [9], [10] and [11] that will be used in determining the routing datagrams in environment of WSNs with limited resources, resulting the following advantages:

- Elimination extensive routing tables;
- Reduced latency by eliminating signaling packets: the route request (RREQ) and route reply (RReply) employed, for example, in AODV;
- Implicit self-recovery mechanism in case of failure;
- Making it inherent in energy savings.

2 Fundamentals of the TCNet model

The situation shown in this section represents a typical problem encountered in applications in WSNs.

2.1 Example of scenario found in WSNs

The Fig. 1 shows a scenario of a WSN in which the sink node acts as Access Point to IP infra-structured networks, allowing information to be obtained from the other sensors in the network.

In this multicast scenario, the message has to be forwarded by several intermediate nodes before arriving at the desired destination. In the conventional protocols, the intermediate nodes have stored route tables, where the hop to be executed are planned to be carried out.

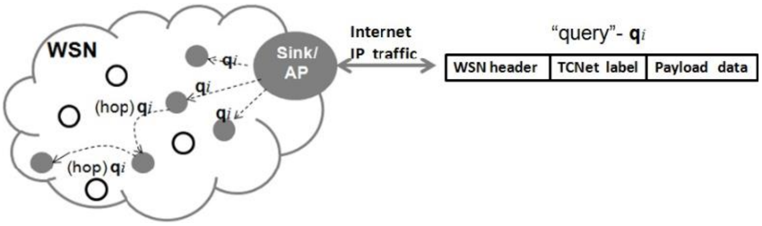


Fig. 1. Scenario of a sink node querying a set of sensor nodes in a WSN and the frame structure used by the TCNet model.

In the TCNet scenario the sink node sends a query-q_i in a specific order defined by a QoS criterion, and each node of the network performs the following procedures:

- verifies if it is the called node;
- if it concludes that it is a route node, it updates the payload in the frame with the sensed value and rebroadcasts the message.

The configuration of the frame structure consists of the fields (WSN header - TCNet label - Payload data) as follows:

- The WSN header (a code word is loaded as the metric to be used by the routing algorithm);
- The TCNet label (a sequence defined by a QoS criterion is loaded, allowing each node to recognize its position in the route);
- The payload (the field that stores the updated data sensed by the sensor).

The process is concluded when all nodes of interest have been visited.

2.2 Analogy with convolutional codes

Considering that one of the possible ways of representing and analyzing a network is through a modeling as a digraph D = (V, E) where:

- V is the set of vertices v_i, representing the network's nodes or equipments;
- E is the set of edges (v_i, v_j), representing the physical or logical connections between the vertices v_i and v_j.

Convolutional Codes use concept of finite automata or Finite State Machines (FSM) [12] defined by a "cross" function (k_n/out_n) as shown in Fig. 2. It is also assumed that a transition occurs at the FSM from state *i* to state *j*. In TCNet each state represents a network node and the transition state indicates that the frame information must be sent from node *i* to node *j*.

Based on this analogy, this work applied the same concepts of convolutional codes in the solution of WSNs network structures due to the similarity of graph topology and the transition of states to behave as a transmission of data packets.

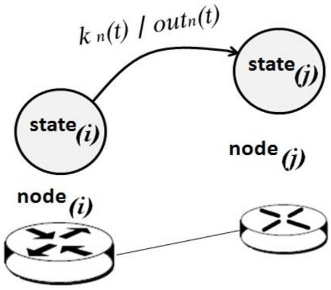


Fig. 2 . Network nodes using as model the states of a Finite State Machine - FSM

.

One method of describing the convolutional code is to consider the operation of a Finite State Machine (FSM). In the context of this work will be used the Mealy machine (MM), as shown in the example of Fig. 3, where the input sequence $k_n(t)$ = 1 1 0 1 ... moving through the MM registers (starting with LSB), changes the states of the MM, in the instants corresponding to $(\cdots t_i,\ t_{i+1}\cdots)$ in the shift register stages representing the states ($\cdots \mathbf{q_i}, \mathbf{q_{i+1}}\cdots$). The result of this operation at each instant *t* is a sequence $out_n(t) = (c_1, c_2)$ of the MM, representing the modulo-2 sum of connections existing between the shift register stages.

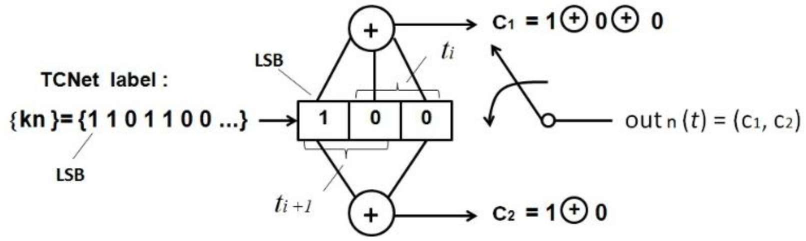


Fig. 3. Example of a MM with the input sequence $\{k_n\}$ generating an output sequence $out_n(t) = (c_1,c_2)$.

2.3 Decoding Protocol using relationships: State Diagram, Tree Diagram, and Trellis Diagram

Thus, as explained in more detail in [8], [9], [11], the tree diagram can be used to describe the evolution of the states, in this work, will be considered as nodes on the tree. As a result, it is possible to trace a path on the tree determined by the input sequence $k_n(t)$ in the MM, as shown in Fig. 4-b where the symbol (1) of the sequence $k_n(t\ )$ specifies a lower branch of a bifurcation in the tree, while symbol (0) specifies the upper branch. Considering that the tree diagram structure is not the best representation for routing analysis, the simplifications obtained due to the elimination of tree diagram redundancies make the trellis diagram one of the tools used to analyze the configurations generated by the finite state machines [13], [14]. As in the state diagram and the tree diagram, the same components can be identified in the trellis diagram, such as: states, branches and routes, as described below, according to Fig. 4-c.

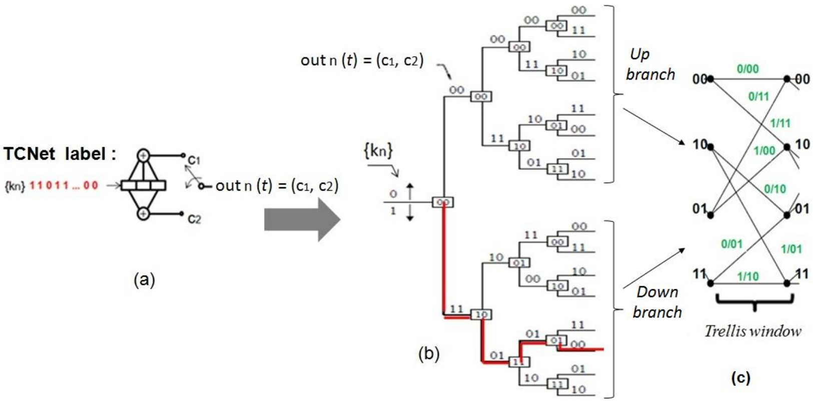


Fig. 4. (a) MM - generator with the input sequence $k_n(t)$ and output word code $out_n(t) = (c_1, c_2)$; (b) Tree diagram based on the MM - generator and detail of the path in the "full line" onthe tree, associated to the input sequence kn (t) in the generating MM; (c) Trellis diagram.

In this new proposal for the decoding of routes in a network, by the similarity of the analysis of branches on a trellis diagram as a function of a received data sequence, this work uses the properties of the algorithm of Viterbi [14].

The principle of the Viterbi algorithm consists of decoding a received sequence, estimating the shortest Hamming distance, in the case of hard decision, between a sequence of symbols received and the weight of the *branch or metric* in the section considered in the trellis, configuring a decision case by *maximum likelihood*, explained in further detail in Section 3.

The Fig. 5 shows an example of a decision on a trellis branch, where the state-10 decides the origin of the sent sequence to establish a *surviving branch*. Using the concepts of the Viterbi algorithm, the state in question analyzes the adjacent branches, calculating the *Hamming distance* between the received sequence and the respective weights of the (metric) branches, deciding for the branch that has the shortest

Hamming distance. This example shows a simple decision with no ambiguity. In other cases where ambiguities exist require the introduction of logical distance based on the *Lattice theory* [15].

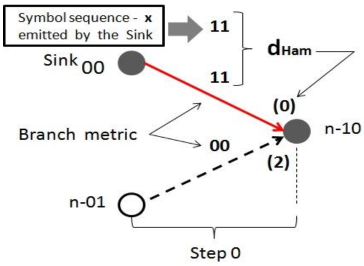


Fig. 5. Example of hard decision used by state (n – 10) to decide on a trellis branch, based on received sequence {11}.

3 Typical application scenario of the TCNet algorithm

The application of the TCNet algorithm for decision making on the trellis, can be seen in a typical broadcast scenario where the network nodes are initially in the same coverage area and one node may be receiving a signal from all others.

Fig. 6 (c) shows a simple situation, where each node receives information only from two other nodes. As an example, the node 10 receives a request with the information *(c_1, c_2) = (11)* starting a query to the trellis decoder resident at the node, Fig. 6 (b), and being informed with the following reception possibilities: node (00) can generate the code (c1, c2) = (11) and node (01) can generate the code (c1, c2) = (00).

The operations performed by the nodes in the same coverage area of the node (sink-00), are shown in Fig. 6 (c). The operation performed by node 10 consists of obtaining d_{Ham} between the code word carried in the header of the FRAME, Fig. 6 (a) and the information of the code word registered in the branch of the trellis resulting in:

- branch as origin node (00) ⇒ $d_{Ham} = 0$;
- branch as origin node (01) ⇒ $\mathrm{d}_{\mathrm{Ham}} = 2$.

The meaning of the value $d_{Ham} = 0$ indicates that the FRAME received by the node (10) originated from the node (sink-00), thus starting other operations by node (10), resulting from the request received: refresh the payload and determine the next code word *(c1, c2) = (01)* using the *{kn}* sequence received in the FRAME label, to be updated in the FRAME header. The other nodes located in the same coverage area perform the same procedure of node (10), not having the result of $d_{Ham} = 0$. In this case, the nodes in the same coverage area remain waiting for a new request.

However, this is a simple example where ambiguity is absent from the decision-making process, only served to validate the concept proposed in this paper. In [11] there are shown other cases that demonstrate the robustness of the process.

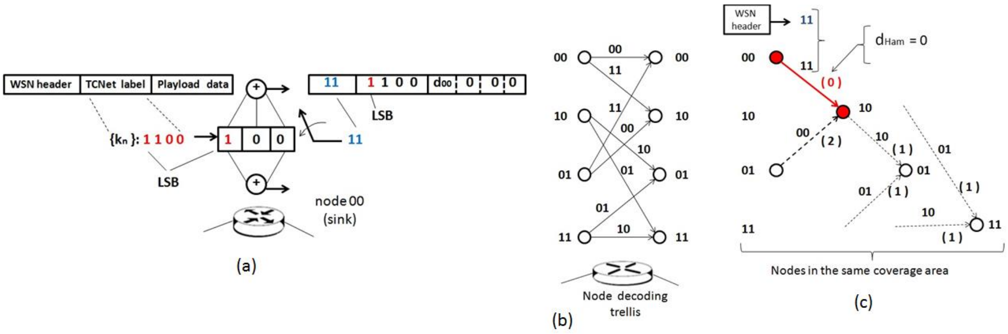


Fig. 6. (a) TCNet FRAME initialization by the sink node; (b)Trellis decoder; (c) Nodes in the same coverage area.

4 Simulation and experimental results

This work uses as simulation environment the OMNeT++ based on C ++ and object oriented. Currently it has been widely used in research for being open software with applications in simulations and modeling of traffic networks, used as reference for comparisons among other techniques due to available frameworks.

4.1 TCNet algorithm simulation scenario

The tests were initially done with an 8 nodes network scenario, considering an ideal case without nodes failure, where the SINK node sends a query with a CBR traffic to verify the reachability of the nodes. Fig. 7-a shows the node model used, configured by a Mealy Machine (MM) with k/n=1/2 and the respective trellis decoder shown in Fig. 7-b. The theoretical result of the route to the chosen scenario, it can be observed in Fig. 7-c, where the resulting route defined on a trellis represent the route from SINK, passing through all the nodes of the network and returning to SINK itself after completing a query.

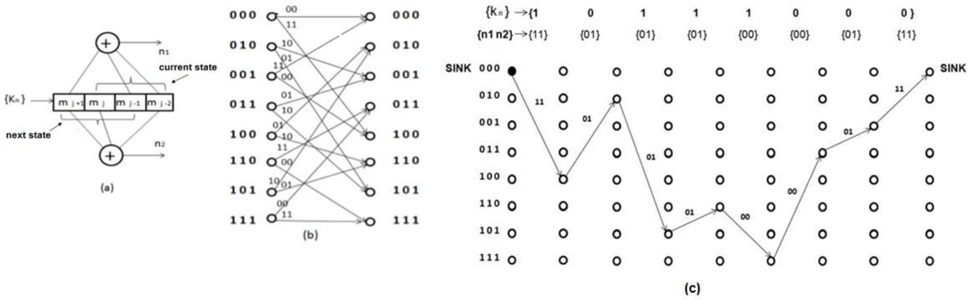


Fig. 7. (a) Mealey machine (MM) with k/n=1/2, resulting output words *(n1, n2);*
(b) Trellis diagram corresponding to MM; (c) Theoretical result of the route over the trellis due to a query.

4.2 Latency of the TCNet algorithm during a query

Although the processing times of the FRAMES *(t_pn)* executed by the MM are different, due to the positions in the displacements of the input sequence in the MM, all nodes in the network will be busy executing the same sequence received during *Multicast*, so collisions will be avoided. Even so, guard bands *(t_g)* were considered to guard against possible collisions due to channel delays.

To obtain network latency during a query, the following data were normalized:

- Processing time *(t_pn)*, considered during the displacement in the MM;
- Channel Delay *(t_c)*, propagation time in the wireless communications channel considered;
- Guard band *(t_g)*, time interval considered by the nodes between the Multicasts of the FRAMES

Considering a theoretical scenario, with random distribution of the nodes in a coverage area with the maximum radius distance *($d_{max} = 1000\ m$)*, (3) was used to define the total latency of the network representing the contributions of the nodes to the considered network:

$$\Sigma T_L = t_p + t_c + t_g \tag{1}$$

Initially, values were plotted showing the cumulative sum of latency individually from the nodes, organized in the same sequence as the route: *ΣT_{L0} (node 0), ΣT_{L4} (node 4), ΣT_{L2} (node 2), ΣT_{L5} (node 5), ΣT_{L6} (node 6), ΣT_{L7} (node 7), ΣT_{L3} (node 3) and ΣT_{L1} (node 1).*

Table 1 shows a progressive latency variation due to sequence processing in the respective MMs configured at the network nodes.

Table 1. Contribution of latency individually by network nodes.

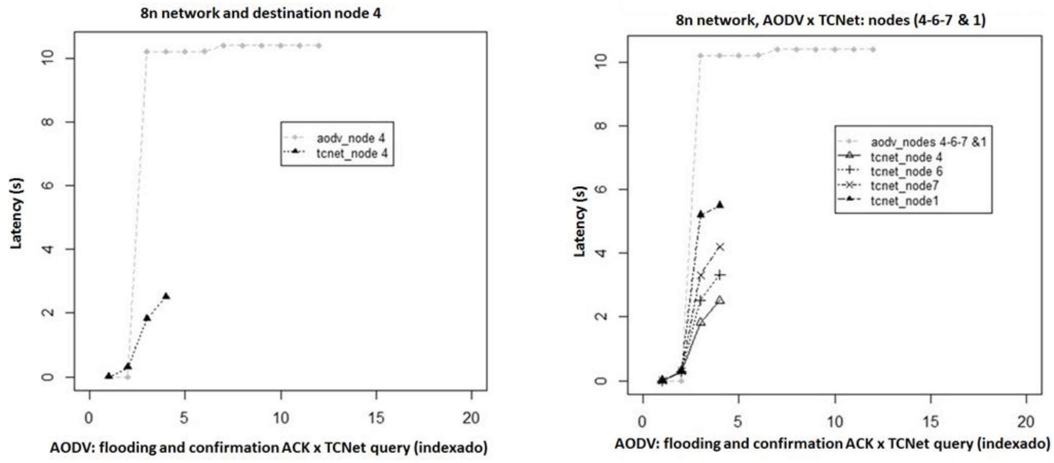
ΣT_{L0} = 0,1 + 0,000003 + 0,2 = 0,300003 s
ΣT_{L4} = ΣT_{L0} + (0,2 + 0,000003 + 0,2) = 0,700006 s
ΣT_{L2} = ΣT_{L4} + (0,3 + 0,000003 + 0,2) = 1,20001 s
ΣT_{L5} = ΣT_{L2} + (0,4 + 0,000003 + 0,2) = 1,80001 s
ΣT_{L6} = ΣT_{L5} + (0,5 + 0,000003 + 0,2) = 2,50002 s
ΣT_{L7} = ΣT_{L6} + (0,6 + 0,000003 + 0,2) = 3,30002 s
ΣT_{L3} = ΣT_{L7} + (0,7 + 0,000003 + 0,2) = 4,20002 s
ΣT_{L1} = ΣT_{L3} + (0,8 + 0,000003 + 0,2) = 5,20002 s

4.3 Performance of the TCNet algorithm in relation to the AODV algorithm

It will be considered an initial scenario of a network with 8 nodes and the same criteria for specific nodes (Source and Destination). In this work will be considered only the latency in a scenario defined: fixed; nodes randomly positioned in (area of 1000 m x 1000 m); the SINK that initializes the communication with the other nodes through *queries* in the network (TCNet); the host (0) is the source node and attempts to communicate with the other nodes of the network using alternate routes through successive *floodings* (AODV).

Fig. 8-a compares the latency of the AODV and TCNet scenarios during the query to node 4, belonging to an 8-node network randomly distributed in areas with the same propagation coverage. It is noted that AODV latency is 75% higher than TCNet latency due to the simplicity of TCNet in performing 1 query. Fig. 8-b compares the

TCNet queries for the other nodes of the network with the AODV route-setting mechanism for the same nodes: 4, 6, 7, and 1: in TCNet the latency increases as the nodes correspond to the last positions of the sequence k_n *(t),* resulting in longer MM processing time during decision making of the target node and in TCNet the latency increases as the nodes correspond to the last positions of the sequence k_n *(t),* resulting in longer MM processing time during decision making of the target node;



(a)

(b)

Fig. 8. (a) Comparison of 1 TCNet query and AODV signaling mechanism to establish a route to node 4; (b) Scenario of comparing TCNet queries for an 8-node network with respect to AODV.

In cases of scenarios where the network nodes density increases, they are shown in [11], which compares the latencies for the most difficult access nodes.

5 Conclusion and future works

The main objective of this work is the proposal of a new algorithm and packet forwarding protocol which is specific for networks with limited processing, communication and energy resources such as WSNs. As solution, a new paradigm based on convolutional codes was proposed called the TCNet model. In addition to this work will be presented study cases of MM with rates k / n = 2/3 and k / n = 3/3, in order to evaluate improvements in latency and energy consumption. So, study other routing protocols for comparison with TCNet and route optimization for cases of node failure.

References

1. IETF Roll Working Group routing over low-power and lossy networks, http://www.ietf.org/dyn/wg/charter/roll-charter.html
2. Vasseur, J.P., Dunkels, A., Interconnecting Smart Objects with IP The Next internet, 1st ed. Morgan Kaufmann (2010)
3. Kurose, J.F., Ross, K.W., Computer Networking A Top-Down Approach Featuring the internet, 5th ed. England: Addison-Wesley, (2009)
4. ROST, P.; FETTWEIS, G. Green communications in cellular networks with fixed relay nodes. In: Cooperative Cellular Wireless Networks. Cambridge, Reino Unido: Cambridge University Press, 2011. p. 300–323. ISBN 978-0-521-76712-5.
5. MOY, J. Open Shortest Path First. 1988. Available in <http:www.rfceditor.org/rfc/rfc2328.txt. Access in: May 2014>.
6. REKHTER, Y.; LI, T.; HARES, S. Border Gateway Protocol. 2006. Available in:<http:www.rfc-editor.org/rfc/rfc4271.txt. Access in: May 2014>.
7. Perkins, C., BeldingRoyer, E., and Das, S., "Ad hoc on-demand distance vector-(AODV) routing," RFC 3561, july 2003, http:www.rfc-editor.org/rfc/rfc3561.txt
8. Lima, D. F., Amazonas, J.R., "TCNet: Trellis Coded Network - implementation of QoS-aware routing protocols in WSNs," IEEE Latincom 106445, november 2012, http:www.ieeelatincom.org/2012/
9. Lima, D. F., Amazonas, J.R., "TCNet: Trellis Coded Network - implementation of QoS-aware routing protocols in WSNs," IEEE Latin America Transctions Vol 11, may 2013, http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6495636
10. Lima, D. F., Amazonas, J.R., "A Trellis Coded Networks-based approach to solve the hidden and exposed nodes problems in WSN," SICOTEL2014 Seminario Taller Latinoamericano de Instrumentación, Control y Telecomunicaciones, April 2014, http://www.gituq.edu.co/memorias%20sicotel2014.pdf
11. Lima, D. F., "Design and implementation of a new algorithm and routing protocol based on convolutional codes using TCNet: Trellis Coded Network", Thesis (Doctorate) - Polytechnic School of the University of Sao Paulo. Department of Telecommunications Engineering and Control, 2015, http://www.teses.usp.br/teses/disponiveis/3/3142/tde-29012016-085422/pt-br.php
12. Hopcroft, J.E., Ulman, J.D., "Introduction to Automata Theory, Languages and Computation", 1st ed. Addison Wesley, 1955
13. Haykin, S., Moher, M., Communication Systems, 5th ed. John Wiley & Sons, Inc, 2009
14. Proakis, J.G., Salehi, M., Digital Communications, 5th ed. Mc Graw Hill, 2008
15. Gratzer, G., General Lattice Theory, 1st ed. Academic Press, Inc, 1978