Robustness Situations in Cases of Node Failure and Packet Collision Enabled by TCNet: Trellis Coded Network - A New Algorithm and Routing Protocol

Diogo F. Lima Filho[1(✉)] and José R. Amazonas[2]

[1] Universidade Paulista - UNIP, Rua Dr Bacelar, 1212, Vila Clementino, São Paulo, Brazil
dioferlima.usp@gmail.com

[2] Escola Politécnica, da Universidade de São Paulo, Av. Prof. Luciano Gualberto - Tr 3, 158, São Paulo, Brazil
jra@lcs.poli.usp.br

Abstract. This research exploits the new concept of route discovery using TCNet - Trellis Coded Networks an algorithm and routing protocol based on convolutional codes to be used in WSNs an important infrastructure of the Internet of Things (IoT) architecture. This work shows the robustness of the TCNet algorithm in making decisions in cases of nodes failure and packages collisions, taking advantage of the regeneration capacity of the trellis. This proposal innovates in making decisions on the node itself, without the need of signaling messages such as “Route Request”, “Route Reply” or the RTS and CTS. TCNet uses low complexity Finite State Machine (FSM) network nodes (“XOR” gates and shift registers), eliminating the use of any routing tables by means of Trellis decoding, where the sequence of states of the FSM, corresponds to a network route, and can be chosen based on different optimization criteria.

Keywords: Wireless sensor networks · Finite state machine · Nodes failure Packages collisions · Trellis decoder · Trellis regeneration

1 Introduction

The constant searches to obtain protocols that support the WSNs networks are challenging due to the dynamic characteristics of these networks. Attempts to adapt the routing protocols of infrastructure networks to *ad hoc* networks are often inconsistent to identify issues such as: frequent changes in topologies, poor link quality, restricted bandwidth and power limitation.

The WSNs configure an *ad hoc* network scenario [1], where the nodes are the routers themselves due to the lack of network structure. In addition, the difficulty of links in covering large areas, suggests a distributed management of resources with intelligent protocols. *Ad hoc* devices can also be subjected to adversities that may render them inoperative. For these reasons, it is necessary that *ad hoc* networks must be

A.-S. K. Pathan et al. (Eds.): SGIoT 2018, LNICST 256, pp. 100–110, 2019.
https://doi.org/10.1007/978-3-030-05928-6_10

robust to failures so that important parameters such as latency, packet loss and energy consumption are not drastically affected.

This work proposes a new routing protocol based on convolutional codes addressed to *ad hoc* networks used to implement WSNs. It has been evaluated by means of efficiency analysis compared to the currently used traditional AODV [2] protocol that depends on the construction of routing tables, similar to RPL [3], both differing from TCNet that proposes a novel and different paradigm. This evaluation takes into account: latency and robustness in the presence of failures. To meet these characteristics, the protocol proposed in this research offers the following advantages that are compatible with the limited resources of WSNs:

- Elimination of routing tables;
- Reduced latency by eliminating the route request (RREQ) and route reply (RReply) signaling packets employed in protocols that use routing tables;
- Implicit self-recovery mechanism in case of failure.

1.1 Related Work

Over the past 15 years, routing in IP networks has been a topic of great interest and has led to the emergence of several routing protocols. The main function of the routing protocol is to determine the "best" path to reach a destination according to various metrics and objective functions.

Routing tables are populated in routers and indicate the best next hop for each reachable destination. Several routing protocols have been developed for intra-domain (e.g., AODV [2], RIP [4], IS-IS [5], OSPF [6], OLSR [7]) and inter-domain routing (e.g., BGP [8]).

Quality of Service (QoS) [9] is the network's ability to meet certain performance criteria such as network delay, jitter or packet drop probability and to perform a number of tasks in the network as packets are forwarded from the source to the destination.

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 Future Internet and other IP networks. The Internet Engineering Task Force (IETF) represented by the Routing Over Low-Power and Lossy Networks (ROLL) Working Group [10] discussed a series of existing protocols, and to that end, defined the following set of requirements for WSN's network: routing metrics; scalability; network stability; degree of constraints and application aware routing as extremely challenging because of the high degree of network constraints.

1.2 Contributions and Proposal of the Paper

This work exploits the new concept of a "Trellis Coded Network" (TCNet) introduced by the authors in [11] for discovering the route to be followed by datagrams arriving at each of the network's nodes. The TCNet model explores the forward mechanism of routing protocols in analogy to Forward Error Correction (FEC) codes and its association with the states of a convolutional code. A trajectory in the convolutional code trellis representation corresponds to a path in the WSN and can be discovered by means of a Viterbi-based algorithm [12] proposed in 1967 for decoding convolutional codes based on the trellis diagram. TCNet implementation is on the level of proof-of-concept, and it can be easily adapted to work with IoT OSs like ContikiOS, TinyOS, FreeRTOS and others emerging OSs for the IoT [10].

This paper adds contributions to the proposal introduced in the previous work [11], where it will be shown the robustness of the process in case of nodes failures, and proposing solutions in cases of hidden and exposed nodes in WSNs. Briefly, such problems may be described as follows [13]:

- The hidden node problem refers to the collision of packets at a receiving node due to the simultaneous transmission of those nodes that are not within the direct transmission range of the sender, but are within the transmission range of the receiver, so both nodes transmit packets at the same time without knowing about the transmission of each other.
- The exposed node problem refers to the inability of the node that is blocked due to transmission by a nearby transmitting node to transmit to another node.

2 TCNet Algorithm Implementation Scenario

For a better understanding of the TCNet decoding mechanism, consider the example shown in the previous work [11]. The sink node initializes the frame loading the WSN header field with the information generated by the MM generator $(out_n(t) = (c1, c2))$ and transfers the input sequence *({kn})* to the TCNet label field. Initially all nodes in the sink's node coverage area receive the request from it, Fig. 1a. Figure 1b shows the decoding trellis in which we realize that each node can only receive information from two other nodes. For example: node *(10)* receives information from node *(00)* when node *(00)* generates the code *(c1, c2) = (11)* and receives information from node *(01)* when node *(01)* generates the code *(c1, c2) = (00).*

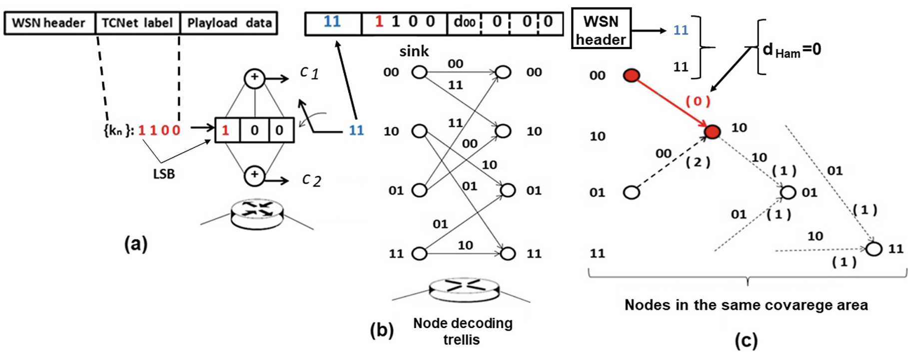

Fig. 1. (a) Initialization of the TCNet frame by the sink node: the input sequence *{kn}* is loaded on the TCNet label field and the output sequence $out_n(t) = (c1, c2)$ is loaded on the WSN header field; (b) Decoding trellis; (c) Nodes in the sink's coverage area receive the query.

Figure 1c shows the operations performed by each node when they receive the frame sent by the sink. For instance, node (10) evaluates d_{Ham} between the code received in the WSN header and the codes of the trellis branches, obtaining:

- from node $(00) \Rightarrow d_{Ham} = 0$
- from node $(01) \Rightarrow d_{Ham} = 2$.

The value $d_{Ham} = 0$ indicates that the received frame is for node (10) and it has come from the sink. Node (10) loads its information (value of the sensed variable) in the payload. Using the *{kn}* sequence received in the TCNet label, it determines that the next code *(c1, c2)* is (01) and updates the WSN header with that value.

For every other node that receives the frame from the sink, the procedure is performed, but none of them produce a value of $d_{Ham} = 0$. Therefore, these other nodes will not transmit and will wait to receive the next frame. This is a simple example to prove the validity of the concept proposed in this work.

3 TCNet Approach to Solve Node Failure and Package Collisions

Before introducing the TCNet approach to solve node failure and package collisions, this section starts by introducing the capability of trellis regeneration.

3.1 Network Recovery Capability Using Trellis Regeneration

The ability of TCNet to establish routes on a trellis is associated with the complexity of the Mealy machine (MM). The initial scenarios presented in this work use MM configuration with rate k/n = 1/2 and *hard decision* based on *Hamming distance* (d_{Ham}) [12] for the purpose of hardware simplifications, with implications in the reduction of the branches that leave the nodes, limiting the ability of decision of the trellis, as will be shown later.

An alternative to extend the options of connections between the nodes of a network is to use smoothing of the *Hamming distance*, through the concept of *soft decision* decoding, where the soft decision metric uses the concept of *free distance* $\left(d_{free}\right)$ [12]. This is possible by changing the inputs and outputs configurations of the MMs combined with changes in the quantities of symbols $\{v\}$ in the input sequence $k_n(t)$ (which indicates the route on the trellis), also corresponding to the quantities of connections leaving a node (e.g. $2^v = 2$ output, used with d_{Ham} in cases of *hard decision*). In cases of *soft decision* the node configurations are extended to $2^v = 4$ and $2^v = 8$ output or settings with more outputs, resulting in more branch options leaving the node, as the example in Fig. 2.

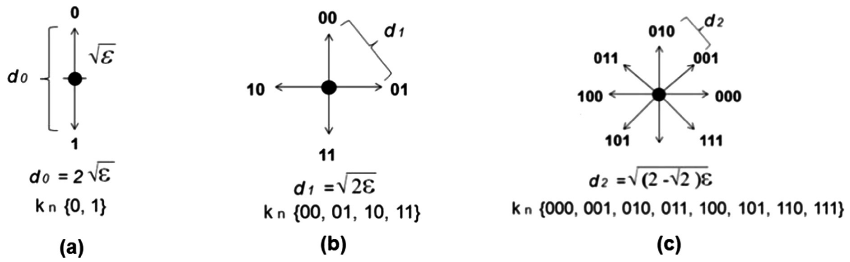

Fig. 2. Configurations of nodes and their respective d_{free} between the symbols of the sequence $k_n(t)$, with $d_0 > d_1 > d_2$. (a) Trellis node corresponding to the sequence set $k_n(t) = \{0, 1\}$; (b) Trellis node corresponding to the sequence set $k_n(t) = \{00, 01, 10, 11\}$; (c) Trellis node corresponding to the sequence set $k_n(t) = \{000, 001, 010, 011, 100, 101, 110, 111\}$.

An example of MM configuration proposed by Ungerboek [14] as shown in Fig. 3(a), represents a MM with rate k/n = 2/3, where the sequence $k_n(t)$ is composed of words with v = 2 symbols: $\{u_1 u_2\}$ moving in the registers of MM with outputs $\{n_1 n_2 n_3\}$, resulting in a increase in the nodes connection capacity as shown by the equivalent trellis of the MM considered, Fig. 3(b). An important result in this configuration is the increase in the ability to establish routes on the trellis and the consequent time reduction of the trellis stabilization, which means to reach the *steady state* in fewer steps.

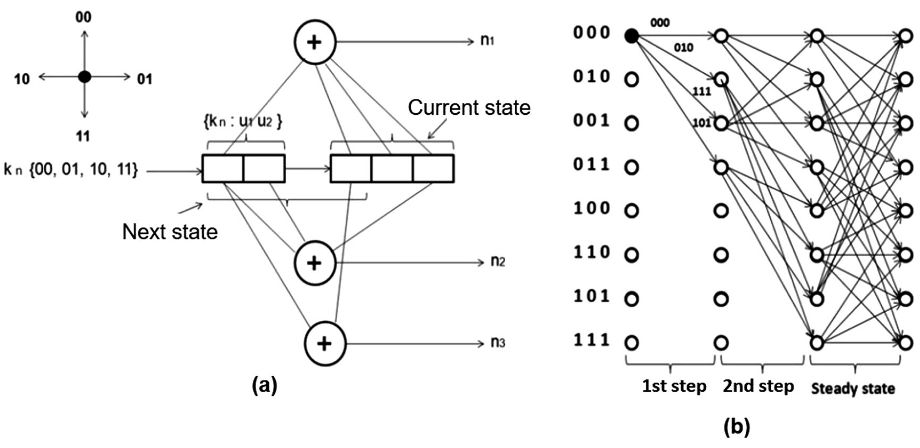

Fig. 3. (a) Configuration of the MM with rate k/n = 2/3, showing detail of node connection capacity; (b) The resulting trellis diagram with $2^v = 4$ branches connecting the nodes of the trellis and the instant that the trellis reaches the *steady state*.

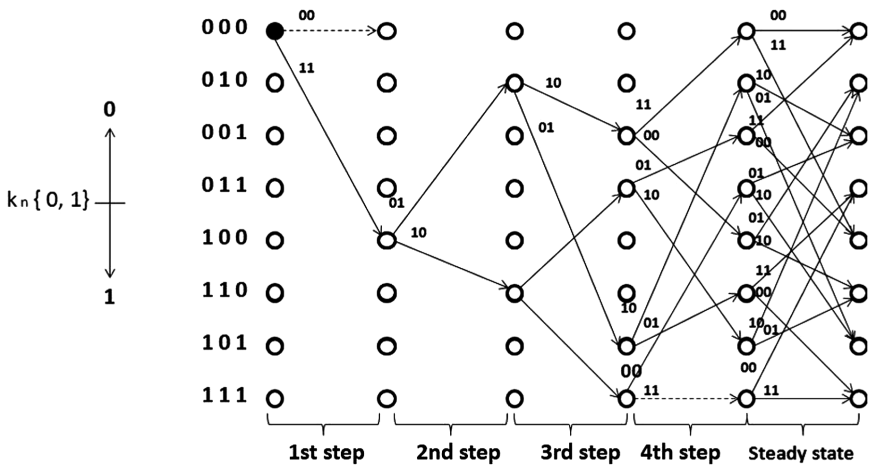

Fig. 4. Configuration of the trellis generated by the MM with rate k/n = 1/2 showing *steady state* after 4 th step.

3.2 Simulation Scenario with Network Nodes Failures Considering the Same Coverage Area

The initial scenario uses a MM with rate k/n = 1/2 and a sequence $k_n(t) = \{1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\}$ defined by a desired QoS. Shifting the input sequence k_n *(t)* in the MM enables the reachability in a standard 8 node network, so that comparisons can be made in situations of nodes failures. In this simulation will be considered the nodes failures before the trellis reaches the *steady state*, due to the reasons presented in the first steps of trellis construction shown in Fig. 4:

- 1st step: Node *(100)* is required for the route initialization;
- 2nd step: Nodes *(010)* and *(110)* are used for trellis initialization, and failure may occur only in one of the nodes;
- 3rd step: Nodes *(001), (011), (101)* and *(111)* are also used for trellis initialization and, only two nodes can fail at the same time.

This work uses as simulation environment the OMNeT++ based on C++ and object oriented [15]. It has been chosen because it is an open software with applications in simulations and modeling of traffic networks, used as reference for comparisons among other techniques due to available frameworks.

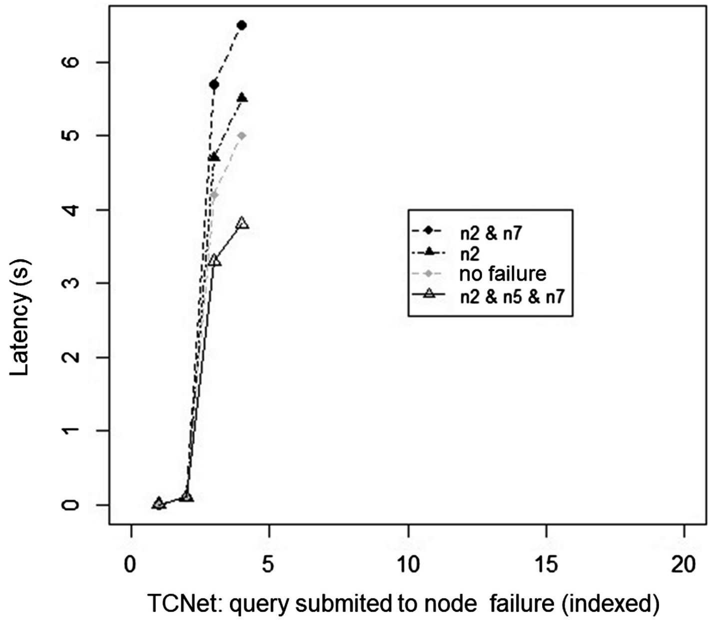

Fig. 5. Network latency resulting during route recovery, in failure cases: 1 node, 2 nodes and 3 nodes.

3.3 Cases of Nodes Failure in a 8 Nodes Network During a Query

The methodology used in this work to measure the efficiency of a network in the presence of node failures will be shown through latency during route recovery. In order to obtain the network latency during a query, the following data were normalized:

- Processing time $(t_p n)$, considered during data 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;
- Time out (t_{out}), time the network waits for the response of the requested node to decide to replace it by another network node.

Considering a theoretical scenario, with random distribution of the nodes in a coverage area with the maximum radius distance (dmax = 1000 m), Eq. (1) was used to evaluate the total latency of the network for the case of node failure:

$$\Sigma T_{Lf} = t_{pn} + t_c + t_g + t_{out} \quad (1)$$

In the considered scenario of nodes failures, the nodes to fail were chosen among those that compromise the trellis recovery. The nodes located in the first steps of the trellis, region of instability, were the chosen nodes in simulation environment below, as shown in Fig. 4:

- One node failure: node – *(010)* or **{2}** in 2 nd step of the trellis path;
- Two simultaneous nodes failures: node – *(010)* or **{2}** and node – *(111)* or **{7}** located in 3rd and 4th steps of the trellis path;
- Three simultaneous nodes failures: node – *(010)* or **{2}**, node – *(101)* or **{5}** and node – *(111)* or **{7}** in 2nd, 3rd and 4th steps of the trellis path;

Figure 5 shows the latency results to recover routes in cases of node failures, taking as reference a route without node failure as commented below:

- In case of one node failure, it results in a 10% increase in latency in relation to the network without node failure, due mainly to the (t_{out}) considered in the simulation;
- In case of 2 - nodes failures, it results in a 25% increase in latency in relation to the network without node failure;
- In case of 3 - nodes failures, it is important to realize that there was a reduction of the network by 50%, and even so the network recovers and completes the query with 70% of the latency of the network without any node failure.

3.4 Cases of Nodes Failure in Extended Networks During a Query

Considering the basic configuration of MM used in this work (Mealy machine with rate k/n = 1/2, Fig. 1(a)) and progressively adding shift registers and connections in its "XOR" gates, it is possible to generate higher densities networks which can be used to compare TCNet efficiency measures.

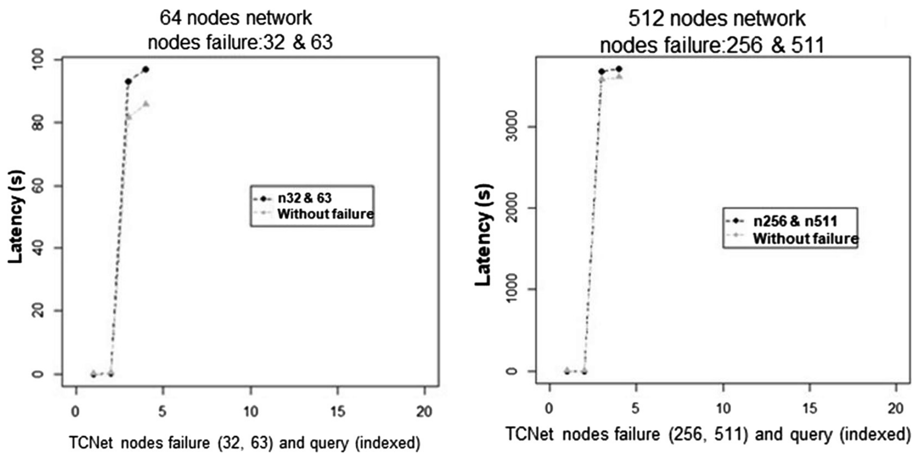

Fig. 6. Latency comparison in route recovery for networks with higher node density, with respect to the routes without failure considering two cases of increased networks.

Table 1. Latencies comparison in route recovery, as the nodes density increases.

Network	Nodes failure	Latency increase
16 nodes	n8 & n15	20%
32 nodes	n16 & n31	18.8%
64 nodes	n32 & n63	12.5%
128 nodes	n64 & n127	7.3%
256 nodes	n128 & n255	5.3%
512 nodes	n256 & n511	2.7%

The nodes selected for the case of failure in the higher density networks were chosen in the first steps of the trellis, (instability region of trellis). The results shown in Fig. 6 refer to the latencies comparison in route recovery, as the nodes density increases, with respect to the routes without failure, considering two cases of increasing network density: 64 nodes network and 512 nodes network.

Table 1 shows a progressive reduction of latency during route recovery, using the simulation environment [15], as the number of nodes increases in the network, considering more cases, making easier the trellis recovery.

3.5 Simulation Scenarios with Packet Collision

The main challenges in designing a routing algorithm for ad hoc networks according to [13], besides the problems related to: randomness of nodes and resource containment they are the hidden and exposed terminal. The hidden terminal contributes to the degradation of the data rate transfer in the network due to collisions caused and the terminal exposed is the existence of nodes belonging to the network but that are outside the coverage area. The practice adopted by the conventional protocols to overcome the problem consists of the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) procedure [16] which is summarized with the signaling messages using the controls mechanism: Request to Send, Clear to Send, Data, Acknowledgment (RTS, CTS, Data and ACK), representing a complex solution to the limited capacity of WSNs.

The contribution proposed by TCNet [17] consists of the decision made by the node itself, in being part of the route, using finite state machines, without the need for network signaling messages as: Route Request or Route Reply. There is still the possibility of using codes based on diversity as Code Division Multiple Access (CDMA), thus allowing channel sharing by the nodes. Associating the CDMA technique in allowing diversity at the physical layer with the capacity of the TCNet algorithm in using strategies that allow the node to decide whether it belongs to a particular route, results in a combination that contributes to a query at a given time. The Fig. 7(a) shows a classic terminal scenario using CSMA/CA, where nodes A and C transmit at the same time to node B, occurring packet collision, because nodes A and C are in the hidden state of each other. The channel sharing solution involves the use of diversity at the physical layer level using Code Division Multiple Access (CDMA) as shown in Fig. 7(b). The solution adopted by TCNet uses the decision concept taken by the node

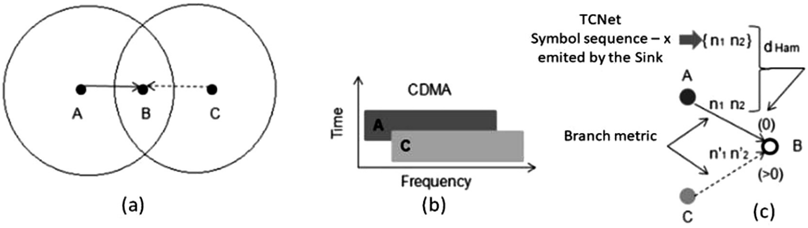

Fig. 7. (a) Scenario of hidden terminal using decision (CSMA/CA); (b) CDMA enabling them to share the same channel for packages A and C; (c) Mechanism of TCNet decision.

itself, based on the algorithm of Viterbi explained in [11], decoding the received sequence and estimating the minimum Hamming distance between the symbols of the sequence sent by Sink and the weight of the path branches as shown in Fig. 7(c).

Another possible collision scenario in ad hoc networks is the existence of nodes belonging to the network, but outside the coverage area, thus configuring the case of the exposed terminal. The TCNet scenarios are usually applied to hundreds of nodes distributed over large areas, suggesting in these cases the subdivision of these areas into *clusters*. In light of this, we propose in future works, studies of cases where scenarios with node *clusters* will be presented in order to solve packet collision.

4 Conclusions and Future Works

In this work it was reviewed the innovative approach of TCNet model introduced by the authors in [11] which exploits the new concept of route discovery quite advantageous for limited resources networks as is the case of WSNs.

It was shown the robustness of the process in case of nodes failures, and proposing solutions in cases of hidden and exposed nodes due to the packet collision problem in WSNs. In future works we intend to: (i) present results of efficiency of the process using *cluster* of nodes to solve packet collision and; (ii) to verify which extensions should be incorporated into RPL so that it can work according to the TCNet paradigm.

References

1. Sivalingam, K.M.: Tutorial on wireless sensor network protocols. In: International Conference on High Performance Computing 2002, Bangalore, India, December 2002
2. Perkins, C., Belding Royer, E., Das, S.: Ad hoc on - demand distance vector - (AODV) routing, RFC 3561, July 2003. http://www.rfc-editor.org/rfc/rfc3561.txt
3. RPL: IPv6 Routing Protocol for Low - Power and Lossy Networks. https://tools.ietf.org/html/rfc6550. Accessed 8 May 2018
4. Hendric, C.L.: Routing Information Protocol, RFC 1058, June 1988. http://www.rfc-editororg/rfc/rfc1058.txt
5. Oran, D.: OSI IS-IS Intra domain Routing Protocol, RFC 1141, February 1990. http://tools.Ietf.org/html/rfc1142

6. Moy, J.: Open Shortest Path First, RFC 2328, April 1988. http://www.rfc-editor.org/rfc/rfc.2328.txt
7. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR), RFC 3626, October 2003. http://www.rfc-editor.org/rfc/rfc3626.txt
8. Rekhter, Y., Li, T., Hares, S.: Border Gateway Protocol, RFC 4271, January 2006. http://www.rfceditor.org/rfc/rfc4271.txt
9. Chen, S., Nahrsted, K.: Distributed quality of service routing in Ad Hoc networks. IEEE J. Sel. Areas Commun. **17**(8), 1488–1504 (1999)
10. Vasseur, J.P., Dunkels, A.: Interconnecting Smart Objects with IP The Next internet, 1st edn. Morgan Kaufmann, San Francisco (2010)
11. Lima Filho, D.F., Amazonas, J.R.: A new algorithm and routing protocol based on convolutional codes using TCNet: Trellis Coded Network. In: The 1st EAI International Conference on Smart Grid Assisted Internet of Things – SGIoT/2017. Springer. ISBN 978-1-63190-152-2. http://eudl.eu/proceedings/SGIoT/2017
12. Proakis, J.G., Salehi, M.: Digital Communications, 5th edn. Mc Graw Hill, New York (2008)
13. Fullmer, C.L., Garcia, J.J.: Solutions to hidden terminal problems in wireless netwoks. In: Proceedings of ACM SIGCOMM 1997, pp. 39–49, September 1997
14. Biglieri, E., Divsalar, D., Mclane, P.J., Simon, M.K.: Introduction to Trellis - Code Modulation with Applications. Macmillan Publishing Company, New York (1991)
15. Varga, A.: OMNeT++ Discrete Event Simulation System (2011). http://www.omnetpp
16. Nasipuri, A., Zhuang, J., Das, S.R.: A Multi-channel CSMA MAC protocol for multi-hop wireless networks. In: Proceedings of IEEE WCNC (1999)
17. Lima Filho, D.F., Amazonas, J.R: A Trellis Coded Networks-based approach to solve the hidden and exposed nodes problems in WSNs. In: Segundo Seminário Taller Latinoamericano de Instumentación Control y Telecomunicaciones, Proceedings of SICOTEL, p. 297 (2014). ISBN 978-958-8593-44-9