Time Petri Net model for CL-MAC with Packet Loss Protocol in Wireless Sensor Networks

Ahmed Louazani
Computer Science Department,
University of Oran1, Ahmed Benbella, Algeria
Industrial Computing and Networking Laboratory,
ahmedlouazani@yahoo.fr

Larbi Sekhri
Computer Science Department,
University of Oran1, Ahmed Benbella, Algeria
Industrial Computing and Networking Laboratory,
larbi.sekhri@univ-oran.dz

Abstract

Wireless Sensor Networks (WSN) are expected to operate as long time as possible in all applications and environments. Facing energy challenge, software developers' have to design save-energy programs. In this paper, an extended version of CL-MAC (Cross-Layer-MAC) is slightly presented dealing with packet loss problem modeled using Time Petri net (TPN). TiNA (Time Net Analyzer) tool is used to validate the proposed model. The obtained properties such liveness, boundedness and reversibility prove the correct behavior of the new version of CL-MAC where lost packets problem is solved. Basically, CL-MAC protocol was designed to reduce both energy and latency. Its main engine is to wake-up only nodes within the routing path.

Keywords - Wireless Sensor Networks, Cross-layer Optimization, CL-MAC Protocol, Energy consumption, Delay sensitive, Time Petri net.

1. Introduction

In literature, many WSN protocols have been proposed to decrease energy wastage resources such as idle listening, over emitting, and collision. CL-MAC [Kec 13], [Kec 10] is one of them and it was designed for delay sensitive application such as forest fire detection and chemical industry monitoring. It's an energy efficient cross layer protocol, where MAC and network layers share control information variables to build routing table, neighbor list, and reserve a low latency path from the source to the sink. As mentioned in [Kec 08], [Kec 10], we have described CL-MAC protocol and compared it with concurrent solutions. The two adjacent layers MAC and network exchange control information to find the shortest path to the sink so that all nodes belonging to the same path relying initiator node to the sink must be ready to route packets at the right moment. Any other node which is a neighbor to one path-node not belonging to the path has to turn off its transceiver from the beginning to the end of the routing process. CL-MAC considerably reduces energy consumption using only three communication packets (*CTS*, *DATA*, *ACK*) instead of four ones (*RTS*, *CTS*, *DATA*, *ACK*) unlike other MAC protocols. Referring to energy consumption model [Est 99], [Kyu 14], and according to the node radio characteristics, for each hop, CL-MAC saves the amount of needed energy to communicate RTS packet (in [24], RTS packet size is equal to 118 bits). Hence, for only one hop communication, the amount of saved energy is calculated as follow (we assume that nodes are 10 meters distant):

$$(K \qquad) + \qquad (1)$$

$$(K \qquad) \qquad \times (2 \qquad) \qquad (2)$$

CL-MAC protocol has good performances under the following hypotheses: A safe network, pleasant environment, flat topology and using control information of two adjacent networks' layers (network and MAC). These hypotheses bring us to spell out its vulnerability face to security issue [Yas 12]. In [Lou 14] a defense mechanism for the CL-MAC protocol against wormhole attacks is presented. Another communication issue remains hidden by the second hypothesis that our paper deals with. In real environment field deployment, many other communication handicaps occurs leading to

---

In: Proceedings of the 3rd Edition of the International Conference on Advanced Aspects of Software Engineering (ICAASE18), Constantine, Algeria, 1,2-December-2018, published at http://ceur-ws.org

packet loss like: interference, obstacle, impediment, and nodes' moving. So, to promise packet delivery, the source node will retransmit to destination the no-acknowledged packet. The next section deals with packet loss problem in CL-MAC.

The remainder of this paper is structured as follows. Section 2 covers the new CL-MAC version reviews with packet loss problem. Section 3 uses TPN model for the proposed solution and experimental results are given. Section 4 is reserved for the conclusion to summarize the paper and propose futures works.

2. Proposed Solution

To overcome the lost packet problem, two variables are added to CL-MAC algorithm: (i) *Nb_transmission* variable to count the lost packet retransmission tentative number, (ii) *Waiting_delay* a countdown variable to measure the estimated needed packet propagation time.

When a source node wants to communicate with another one (destination), four packets are exchanged between both nodes as shown in figure 1. Before sending its first packet "RTS", node "*A*" sets waiting_delay variable to the estimated RTS packet propagation time value. This duration is the elapsed time from the injection of the first RTS bit in the network by node "*A*" till the reception of the last CTS bit by the same node as a replay. Hence, we formalize the packet propagation time by equation (3).

$$n(RTS) +$$

$$n(C\ \ S) + \qquad (3)$$

Where δ is the time for equation adjustment.

Nb_transmission variable is set to one indicating that's the first packet transmission. Then node "*A*" waits until *waiting_delay* variable expire. When it elapsed (waitin_delay = 0), node "*A*" concludes that his packet was lost (didn't reach node "*B*"), then will fetch whether *Nb_transmission* didn't exceed a given threshold retransmission number. In our proposed solution, we fix the threshold to 3 retransmissions of the same lost packet (here it's fixed to 3 just to test the proposed solution behavior). If *Nb_transmission* is equal or less than 3, then node "*A*" re-sets *waiting_delay*, increments *Nb_transmission* and retransmits the packet. If *Nb_transmission* exceeds 3 transmissions temptations, the source node halts the current communication and start finding new path process or checks for an alternative one. Then, the interrupted communication will restart from the beginning on the new calculated path (see figure 2).

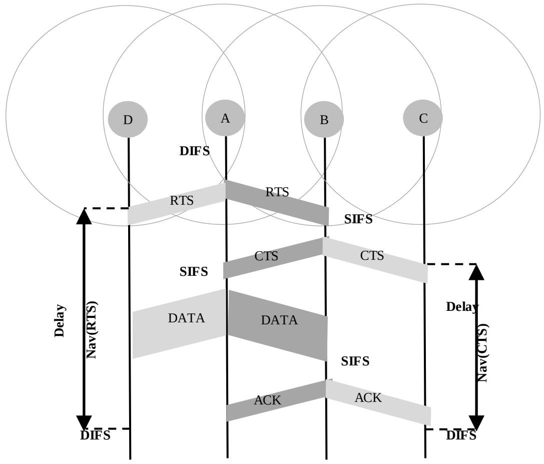


Figure 1: Exchanged packets between node A and node B

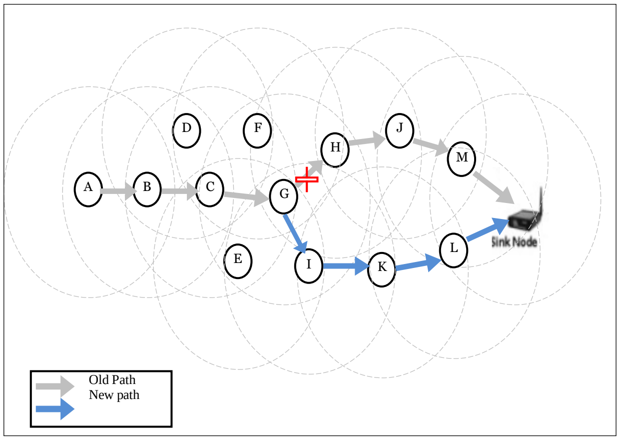


Figure 2: Path linking node G to node H is broken

3. Modeling CL-MAC Protocol With Packets Loss

In order to formally prove and verify the correct behavior of our proposed solution, we have chosen a suitable mathematical model according to communication protocol specifications. The protocol operating mechanism is a time discret events so Time Petri nets (TPN) seem more attractable and suitable for both their ability to easily model temporal constraints of communication scenarios, and the existence of TiNA tool [Ber 04]. TiNA is a software tool for TPN and automatons properties' verification like boundedness, liveness, deadlock-freeness, reversibility, etc. [Ber 83].

Figure 3 depicts the TPN model of CL-MAC protocol with packet loss. This model is a revised one (see [Kec 10]). Intervals values associated to transitions refers to relative time of transmitting packets (RTS, CTS, DATA, ACK) according to IEEE 802.11 standard and respected by our proposed solution.

3.1 Model Hypothesis

CL-MAC with packet loss TPN model works under the following hypothesis: DIFS duration = SIFS duration = 1 time unit, control packets RTS, CTS and ACK consume 3 time units each one, DATA packet requires 10 time units for its transmission. One unit time is added (reserved) for packet processing (a new packet generation, received packet reading) and also for some environment's handicap. M0 is the initial marking. Initially, only the places p1, p8, p14 and p17 are marked by one token each one.

$$M_0 = \begin{pmatrix} p_1 \\ p_8 \\ p_{14} \\ p_{17} \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \qquad (4)$$

3.2 Model explanation

In this section we give a TPN model for the proposed solution. The lost packet event may occur when node sends RTS, CTS, DATA, or ACK packet that didn't reach its destination (one hop neighbor). After waiting the replay a sufficient time needed for packet and packet replay propagation, the sender node retransmits the packet again (the lost one). Whatever is the packet, the process is the same. A packet is retransmitted at most four times. After the fourth packet transmission fail, then chronologically node will interrupt this communication, looks for a new path and restarts the communication on the new path. So, to model this idea, we first describe TPN model transitions (see table 1).

Table 1: CL-MAC with packet loss model transitions

Transition	Description
t1	RTS packet sending in network by the sender node
t2	DATA packet unicasted by the sender
t3	Sender switches to sleep mode
t4	Sender switches to weak-up mode to start a new communication (frame)
t5	Receiver sends a CTS packet as a replay to the RTS one generated by t1
t6	Receiver sends an ACK packet after receiving the DATA packet
t7	Receiver node switches to sleep mode after communication completion
t8	Neighbor node, not belonging to the routing path, switches to sleep mode
t9	Neighbor node, not belonging to the routing path, switches to weak-up mode after communication completion.
t10	CTS packet sent by next hope node
t11	Receiver sends DATA packet to the next hope node.
t12	Next hope node switches to sleep mode
t13	DATA packet losing.
t14	DATA packet retransmission after $(2 \times \text{prpagation\_delay(packet)} + \delta)$ time unit
t16 et t18	Empty retransmission counter after a successful transmission
t17	Sender initialization for a new communication after failures (after 4 retransmission fails).

t19, t20	Receiver initialization after a long waiting time for DATA packet
t21	Next hop node initialization after a long waiting time for DATA packet.

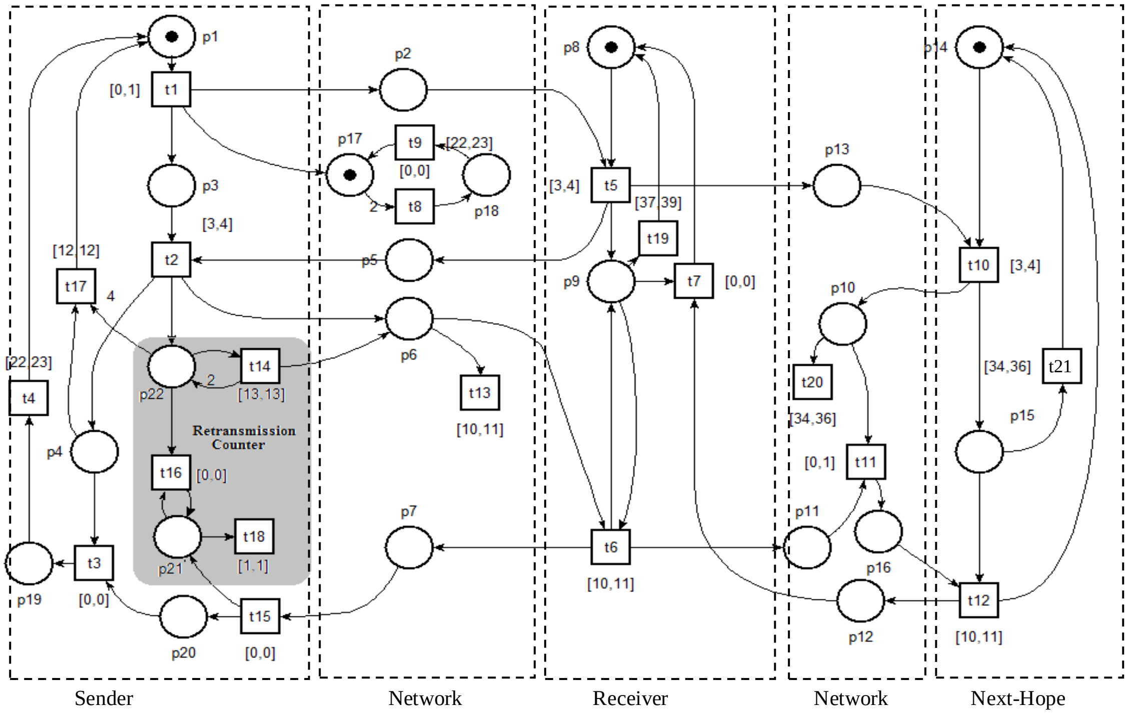


Figure 3: Time Petri Net Model for CL-MAC with Packet Loss.

The communication nature in such networks is a multi-hop scenario. An end-to-end packet communication is a replication of the same atomic communication between two neighbors node. So that the destination node in the i^{th} communication will be the source of the $(i+1)^{th}$ one. For this reason, our model is restricted to only one hop communication.

As illustrated in Figure 3, each node in the network is modeled as follow:

Sender: Transitions t1, t2, t3, t4, t14, t15, t16, t17 and t18.The first transitions t1 stands for RTS packet sending process at precise moments modeled by temporal intervals associated to the transitions (RTS packet is sent after a DIFS and DATA packet is sent after reception of a CTS within 3 to 4 time units). Place p3 models node waiting state for CTS packet. After sending a packet, the sender node will wait for a replay from his path member neighbor as shown in figure 1 (from the receiver). Whenever the sender didn't receive a replay to its initial sent packet, retransmit it again and increments the counter retransmission number. This action is modeled by transition t14. Place p22 models the counter variable, will then get one more token after t14 was getting fired. If four token will be gathered in place p22, transition t17 will then be fired (arc linking p22 to t17 is 4 weights). This situation means that DATA packet is retransmitted four times. So the node has to halt this communication and checks for another alternative path taken into consideration that the actual destination node is broken. Transition t3 depicts node switching from waiting state to sleep state (p19). Transition t15 is fired when ACK packet is received (as a replay to DATA packet reception) and t16 then empties the place p22 (resets the retransmission counter variable) if there are still tokens in. Transition t4 allows node to switch to weak-up mode after a half communication frame. In CL-MAC, a frame is modeled by 44 time's unit.

Receiver is represented by four transitions, t5, t6, t7, and t19. Transition t5 represents RTS packet reception. Here, the receiver is ready to start a new communication according to the exchanged schedule in neighbor discovery phase (place p8 is marked by one token). After receiving RTS packet, the receiver node will then generates and transmits a CTS packet as a replay to the sender node. Place p9 will be then marked by one token as indication of the waiting state to DATA packet. Whenever t6 is fired indicates reception of DATA packet from the sender node. ACK packet will be sent by the same transition t6 represented by the token injected in place p7. Transition t19 forces the receiver node to switch to sleep mode after staying a long time waiting for DATA packet. Also t7 allows the receiver to switch to sleep mode but is fired when place p12 gets one token representing a communication completion (ACK is received from the next hope node). This allows the communication process re-initialization on the side of the destination receiver; the second one is the part of the network between the receiver and the next-hope node.

Network is modeled by two parts. The first one describes the neighborhood of both sender and receiver. Places p18, p17 and, transitions t8 and t9 represent a neighbor node not belonging to the routing path. Place p17 is marked by one token telling that the node is ready to take part of the communication. When it receives a packet not addressed to it (p17 will get the second token), immediately it turns its transceiver and switches to sleep mode. Transition t8 is fired and p18 gets one token as indicator of node sleep state. Place p2 models the RTS packet propagation in the network while p5 stands for the propagation of the RTS packet replay (CTS) sent from the receiver. As mentioned, a packet may be lost due to the environment nature where the network is deployed. Transition t13 here models all the obstacles causing the loss of the packet. In our TPN model, both transitions t6 and t13 have the same firing time's intervals drawing the competitively of these two transitions. Only one will be fired (a packet will be either received by the receiver as its intermediate destination (here t6 is fired) or lost (t13 is fired)). Place p7 stands for ACK packet propagation.

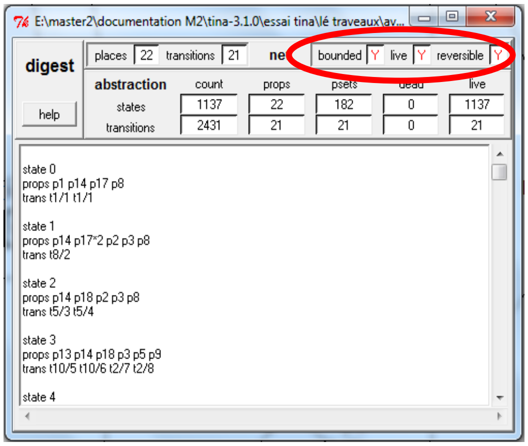
3.3 CL-MAC with packet loss TPN Model Results

Figure 4 illustrates the reachability analysis results of the time Petri net using TiNA tool. The results reveal that the TPN model, of the new version of CL-MAC with packet loss protocol, has effectively good properties:

i. *Boundedness*: the number of tokens in every place is limited to one token except the place p22 witch can get four token inside (that is 4-bounded).The K-bounded property describes the well operating of the CL-MAC protocol with packet loss. In our solution, we have proposed that a lost packet will be retransmitted again and again at most four times (line 22 in the algorithm above reflects this situation).

ii. *Liveness*: the net is deadlock freeness and each transition is always able to be fire infinitely. This property tells us that our solution ensures packet delivery even if some communication handicap or obstacles are present in the deployment environment.

iii. *Reversibility*: the return of the TPN to its initial state shows that the CL-MAC TPN model is reversible. This last property confirms that the new CL-MAC version didn't halt whenever is lost elsewhere, and the protocol tries to find other alternative path after a communication fails.



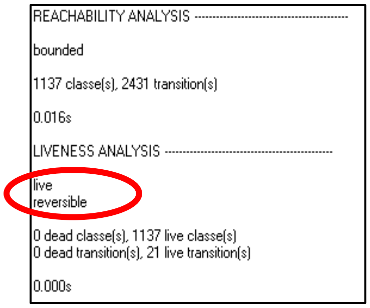
places 22 transitions 21

bounded Y live Y reversible Y

abstraction	count	props	psets	dead	live
states	1137	22	182	0	1137
transitions	2431	21	21	0	21

```
state 0
props p1 p14 p17 p8
trans t1/1 t1/1

state 1
props p14 p17*2 p2 p3 p8
trans t8/2

state 2
props p14 p18 p2 p3 p8
trans t5/3 t5/4

state 3
props p13 p14 p18 p3 p5 p9
trans t10/5 t10/6 t2/7 t2/8

state 4
```

Figure 4 (A)



```
REACHABILITY ANALYSIS ----------------------------------------

bounded

1137 classe(s), 2431 transition(s)

0.016s

LIVENESS ANALYSIS ----------------------------------------

live
reversible

0 dead classe(s), 1137 live classe(s)
0 dead transition(s), 21 live transition(s)

0.000s
```

Figure 4 (B): Reachability analysis using TiNA

After running several times, TiNA generates 1137 classes and 21 transitions. This great number of classes makes impossible to represent the class graph in this paper.

4. Conclusion And Future Work

In this paper, a new version of CL-MAC protocol recovering the packet loss problem is presented. The proposed solution is based on retransmitting the lost packet until it will be received by destination or the retransmission fails four successive times. We have proposed a time Petri net based approach to model the solution depicted by the given algorithm. Formal analysis using TiNA tool allows proving some properties of the TPN model. The obtained results illustrate clearly the well operating of the new extended version of CL-MAC protocol.

In order to strengthen theoretical results, this work is now under² implementation using OMNET++/Castalia simulator to obtain empirical preliminary results in order to enhance TPN model properties. As future work, we try to secure this version of CL-MAC with packet loss protocol against wormhole attacks.

References

[Yas 12] M. Yasir. An Outline of Security in Wireless Sensor Networks Threats, Countermeasures and Implementations. Wireless Sensor Networks and Energy Efficiency. Protocols, Routing and Management Book, pp. 507-527, 2012.

[Est 99] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks, in Proceedings of the 5th IEEE/ACM Annual Conference on Mobile Computing and Networks(MobiCOM'99), Seattle, pp. 263-270. August 1999.

[Kyu 14] Shi-Kyu Bae. Power Consumption Analysis of Prominent Time Synchronization Protocols for Wireless Sensor Networks. In (JIPS) Journal of Information Processing Systems, vol. 10, no. 2, pp. 300-313, 2014.

[Kec 13] B. Kechar and L. Sekhri. CL-MAC: Cross-layer MAC Protocol for Delay Sensitive Wireless Sensor Network Applications. in Petri Nets-Applications, Book chapter, pp. 181-208, May 14, 2013.

[Kec 08]B. Kechar and A. Louazani. CL-MAC: An Energy Efficient Cross-Layer MAC Protocol with Latency Improvement in Wireless Sensor Network, in 17th International Conference on Computer Communication and etworks (ICCCN). Virginia Institute, USA, 3-7 Aug. 2008.

[Kec 10]B. Kechar. L. Sekhri and M.K. Rahmouni. CL-MAC: Energy Efficient and Low Latency Cross-Layer MAC Protocol for Delay Sensitive Wireless Sensor Network Applications. In The Mediterranean Journal of Computers and Networks, Vol.6, No.1. pp. 1-14, 2010.

[Ber 04] B. Berthomieu, P.O. Ribet and F. Vernadat. The tool TINA - Construction of abstract state spaces for Petri Nets and Time Petri Nets . In International Journal of Production Research, Vol.42, No.14, pp. 2741-2756, 2004.

[Ber 83] B. Berthomieu and M. Menasche. An Enumerative Approach for Analyzing Time Petri Nets. In FIP Congress Series, Elsevier Science Publishers, Vol. 9, Amsterdam, Netherland, pp. 41-46. 1983,

[Lou 14] A.Louazani, L. Sekhri and B. Kechar. A Security Scheme Against Wormhole Attack in MAC Layer for Delay Sensitive Wireless Sensor Networks. In International Journal of Information Technology and Computer Science (IJITCS), Vol.6, No.12, pp.1-10, November 2014.