Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 18 (2013) 2525 – 2528

Procedia
Computer Science

International Conference on Computational Science, ICCS 2013

Formal Study of a Novel Network Role-Based Routing Intelligent Algorithm [1]

J. A. Mateo[a,*], H. Macià[a], M. C. Ruiz[a], J. J. Pardo[a], A. M. Ortiz[b]

[a]*Escuela Superior de Ingeniera Informtica*
Universidad de Castilla La-Mancha
Campus Universitario s/n. 02071. Albacete, Spain.
{*JoseAntonio.Mateo, Hermenegilda.Macia,MCarmen.Ruiz,juanjose.pardo*}@*uclm.es*
[b]*Dept. of Wireless Networks and Multimedia Services*
Institut Mines-Telecom, Telecom SudParis, France
torres@it-sudparis.eu

Abstract

NORIA (Network rOle-based Routing Intelligent Algorithm) is a novel routing algorithm for Wireless Sensor Networks (WSNs) which combines various effective techniques in order to reduce energy consumption and improve data routes. This paper presents a formal and rigorous study of NORIA. Prioritised-Timed Coloured Petri Nets (PTCPNs) have been used to describe complete and unambiguous specifications of system behaviour, whereas CPNTools is used to evaluate the correctness of the protocol using state space exploration.

Keywords: Wireless Sensor Networks ; Coloured Petri nets ; Formal Methods ; Routing Algorithms.

1. MOTIVATION

Recently, a novel generation of network protocols that are able to fulfil the new requirements of Wireless Sensor Networks (WSNs) is emerging. Tests and simulations are the usual validation techniques used for these protocols. Although these techniques give us an excellent overview of the protocol behaviour, some undesirable aspects of the protocol could still be undiscovered, such as deadlocks, livelocks and so on. Therefore, formal verification is needed.

In this paper, A novel role-based routing intelligent algorithm called NORIA (Network rOle-based Routing Intelligent Algorithm) [1] will be analysed by mean of CPNTools [2] (tools for Prioritised-Timed Coloured Petri Nets (PTCPNs) [3]). We formally model NORIA using PTCPNs. Next, we analyse this model using state space techniques and validate the correctness of the protocol as well as the absence of deadlocks and proper termination.

*Corresponding author. Tel.: (+34) 967 599200 Ext: 2679 ; fax: (+34) 967 599224.
E-mail address: JoseAntonio.Mateo@uclm.es.

*This work received financial support from the Spanish Government (cofinanced by FEDER funds) through the TIN2009-14312-C02-02 and TIN2012-36812-C02-02 Projects.

Selection and peer review under responsibility of the organizers of the 2013 International Conference on Computational Science
doi:10.1016/j.procs.2013.05.433

2. Prioritised-Timed Coloured Petri Net Semantics for NORIA

NORIA is a distributed routing algorithm for WSNs that creates an energy-efficient communication tree, whose root is the base station (i.e., coordinator or sink), and it assigns roles to the other nodes in order to forward the sensed information to the base station (the interested reader can refer to [4] for further information). Now, the PTCPNs semantics of the protocol is presented below.

2.1. PTCPN Model for WSNs

Assumptions. In order to simplify the model, we have opted to model the packet broadcasting with the transition *Packet broadcast* in such a way each node receives its corresponding packets in a kind of "buffer" (Channel Node_i), $\forall i \in [1 \dots n]$, $n = numnodes$ to temporarily store them before consuming.

Description. Our model is formed by two well-defined parts: the *WSN* and the *Node*. The first one depicts the behaviour of the system as a whole composed by an arbitrary number of nodes and the second one depicts the specific behaviour of each node. As a result, a model will expose only one page for representing the WSN, whereas it will state as many *node* pages as the designer wants to check.

Here, we have used three hierarchical transitions to enact the nodes. Each node has 2 input places: *Channel* $Node_i$ *and ON* N_i. The channel place is used to store the packets received for a short time. The input arc of this place is labelled with an *if statement*, whose mission is to control that the packet sender is in the coverage area of the node radio and its type is *IPM* or *RDM*. When the node has finished its role decision algorithm, the place *FIN* N_i is marked in order to visually check that each node has finished, helping us to discover whether a node is stuck. Nevertheless, a termination place, *Fin* has been added in order to check if all the nodes in the net have finished. Thus, a parameter *NUMNODES* is initialised with the number of nodes in the system, and when this place is marked with *NUMNODES* tokens, the transition *tend* is fired marking the place *end*. Notice that we have defined the type of this place as *BOOLT*, i.e., a boolean token with a time stamp attached, allowing us to extract automatically the time consumed in each simulation.

2.2. PTCPN Model for nodes

Assumptions. As commented above, the amount of battery consumed while the node is waiting differ from that consumed when the node is sending/receiving packets. Moreover, the nodes are started in different moments. The neighbours table is represented as a list that contains the following information at each element: The *ID* of the neighbour, its battery, the number of hops up to the base station and its role.

Description. In Figs. 1, 2 is shown the CPN model of a node. For the sake of clarity, we have decided to highlight the different parts of the algorithm in each figure. In Fig 1, *PART1* models the initialization of each node. As NORIA developers encouraged, we have opted to start all the nodes at a different time since these devices are not synchronised with respect to their initialization. This situation is depicted by means of the time inscription of transition *Init node*. In CPNTools, a transition with a time inscription *@+discrete(a,b)* means that the output token will have associated a time stamp increases with a random value between a and b. This transition is enabled when the place ONN_i of the system is marked. Nevertheless, the decision process starts when the place *Entry* (Figs. 1,2) is marked with the token *1'(RDM,(0,0,100,Master))@51* firing the transition *Receive RDM* on *PART4* of Fig 1, which represents the reception of a RDM packet from the base station. If more than one RDM is received, the place *Discards* is marked by means of the firing of the transition *Discard RDM*. Next, the neighbour table needs to be updated with the information contained in the packet. The next step is to broadcast the IPMs to the neighbours.

The transition *Send IPM* is responsible for marking the place *Exit*, where the tokens are available for being transmitted with the system transition, *Packet broadcast*. Once the messages are sent, a timer (Fig. 2 (top right corner of *PART7*)) must be executed during 51 time units in order to gather the information, in form of IPMs, of the neighbours. Looking back to top right corner of *PART7* in Fig. 2, we have modelled this timer by using the place *c*, which simulates a counter increased either a IPM is received (we suppose that the reception consumes one time unit) or a time unit has elapsed. Thus, for each time unit, the system can elapse a time unit doing nothing or receiving a message.

Here, the transitions *Receive IPM* and *Receive IPM_out* simulate the reception of an IPM, but the main difference between both is that the first one can be fired during that 51 units of waiting for packets and the packets information is used for the parent selection, whereas the second one (Receive IPM_out) is used to receive IPMs out

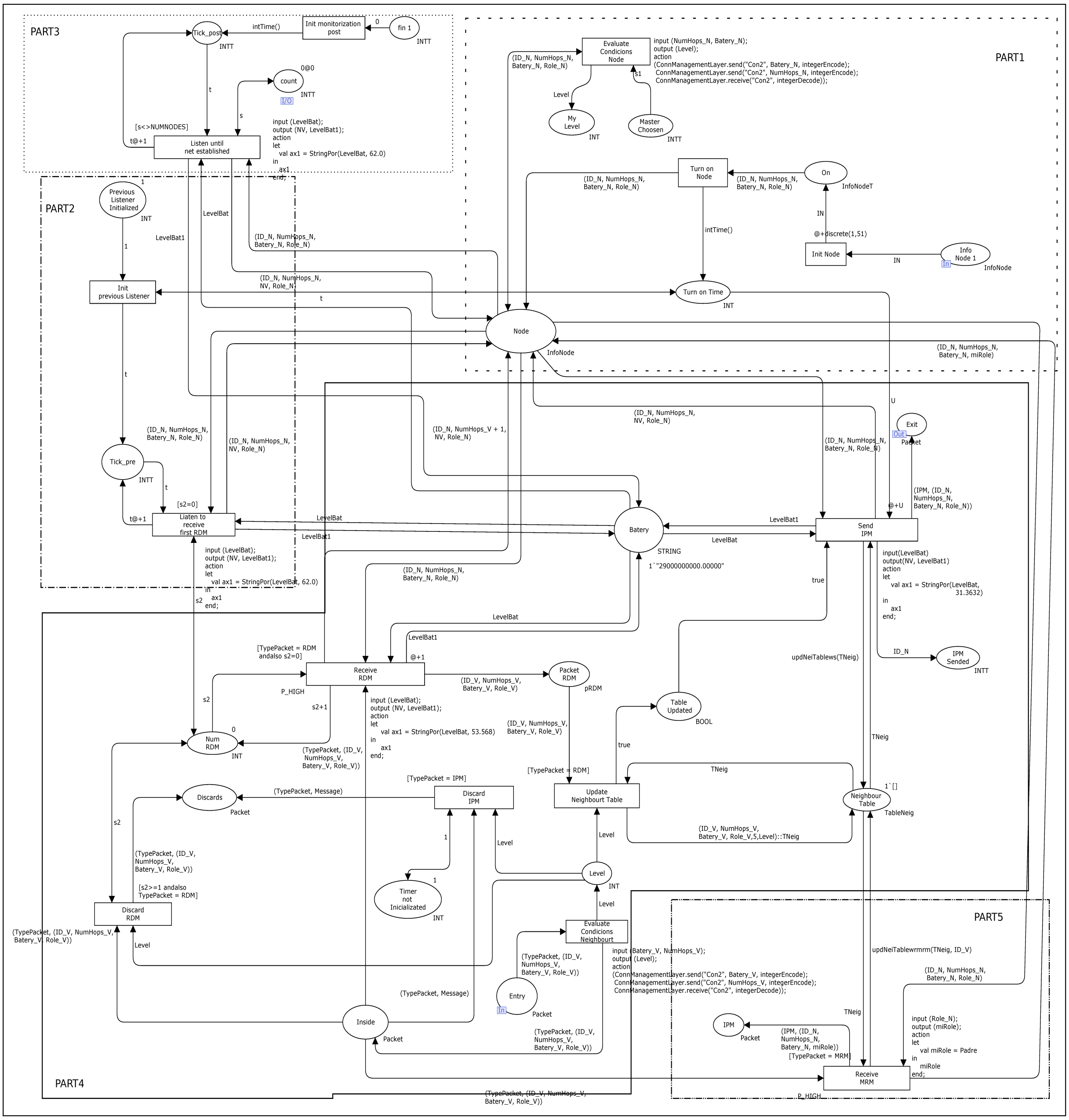

Fig. 1. CPN model for a node (part I).

of the duration of the timer. Next, the node must consult its neighbour table and select its parent and its role. On the one hand, the parent selection is done by the transition *Select Master* of Fig. 2 (lower right corner of *PART5*) and by the function *select_master*. This function encodes the fuzzy algorithm for parent selection. If one master is found in the neighbours table, the node will continue selecting its role with the code accompanying the transition *Choose Role* (lower right corner of the transition). In CPNTools, each transition may have an attached code segment which contains ML code and it is executed when the transition is fired. This code implements the algorithm for role decision. Nevertheless, if no master is found, the node must send an MRM (master request message) to its best neighbour leaf continuing with the role decision process as if a master has been found. Looking at *PART5* of Fig. 1, when the MRM is received (transition Receive MRM), the node sends an IPM in order to notify to its neighbours that its role has changed.

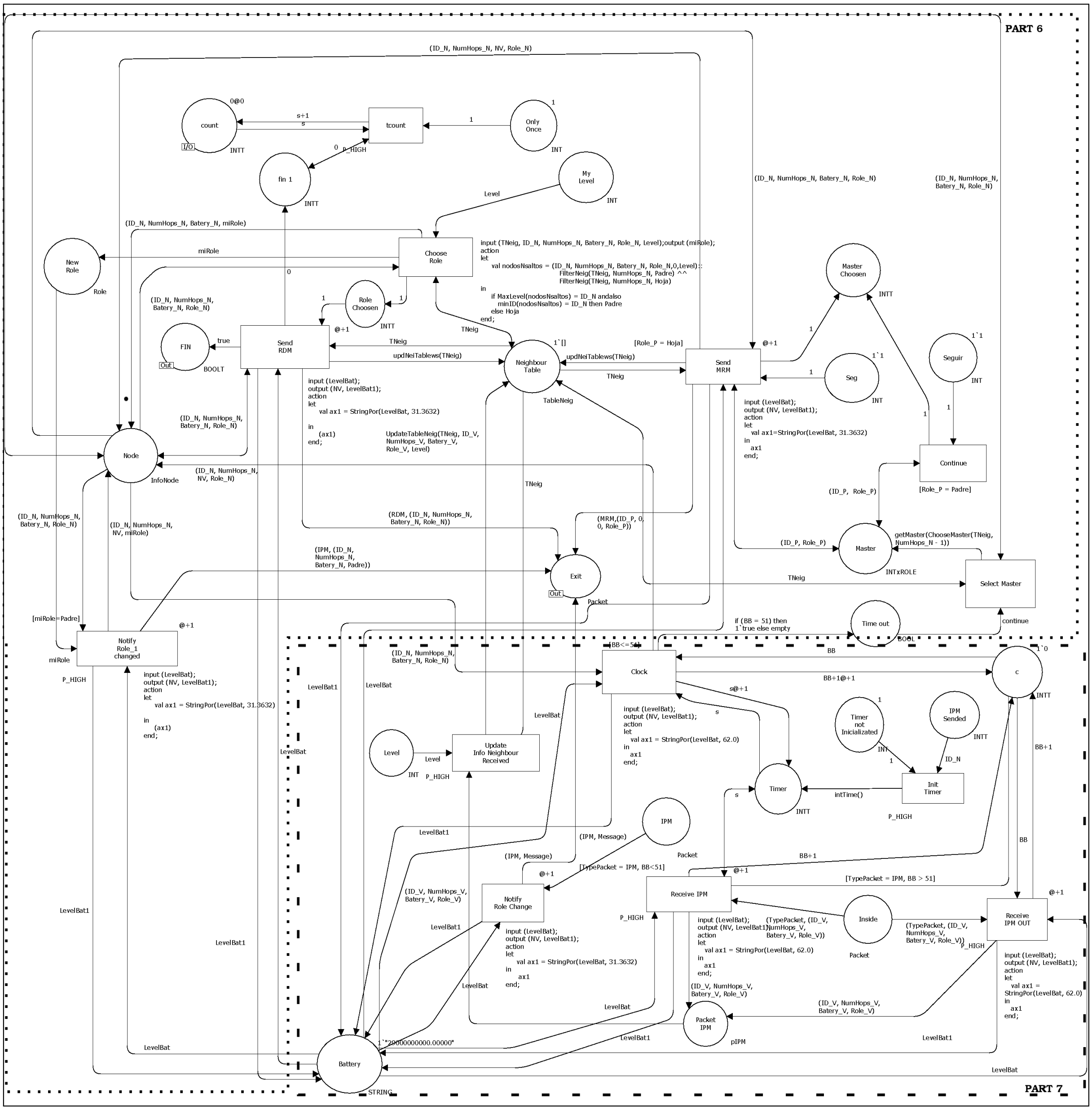

Fig. 2. CPN model for a node (part II).

Finally, once the parent and role decision processes are finished, the node sends a RDM (Send RDM transition in *PART6* of Fig. 2) to start the decision process in the next level and increments the counter of finished nodes. Let us note that the transition *tcount* and the place *count* are depicted with accountability purposes since they store transiently the number of nodes that have finished.

References

[1] A. Ortiz, F. Royo, T. Olivares, J. Castillo, L. Orozco-Barbosa, P. Marron, Fuzzy-logic based routing for dense wireless sensor networks, Telecommunication Systems.

[2] Cpntools web site.

[3] K. Jensen, L. M. Kristensen, Coloured Petri Nets - Modelling and Validation of Concurrent Systems, Springer, 2009.

[4] A. M. Ortiz, T. Olivares, L. Orozco-Barbosa, Smart routing mechanism for green zigbee-based wireless sensor networks, in: ISCC, 2011, pp. 397–403.