Coloured Petri Net Model for Vector-Based Forwarding Routing Protocol

Dina M. Ibrahim[1], Elsayed A. Sallam[2], Tarek E. Eltobely[3], and Mahmoud M. Fahmy[4]
Computers and Automatic Control Dept. Faculty of Engineering, Tanta University,
Tanta, Egypt
dina.mahmoud@f-eng.tanta.edu.eg[1], sallam@f-eng.tanta.edu.eg[2], tarekt@f-eng.tanta.edu.eg[3],
and mfn_288@hotmail.com[4]

ABSTRACT

Modelling and simulation has an important role in understanding the performance of systems like Underwater Wireless Sensor Networks (UWSNs) before its implementation, where development and testing of actual system become expensive and time consuming. Special characteristics of underwater environments makes UWSNs meet many challenges. Many routing protocols for UWSNs were proposed in order to find out an efficient route between the sources and the sink. The aim of this paper is to use Coloured Petri Nets (CPNs) to model and analyze the behavior for one of the routing protocols in UWSNs, Vector-Based Forwarding routing protocol. CPNs are preferred for their ability to study and designate discrete event systems that are considered as concurrent, parallel and asynchronous. Our model is verified by two phases: First, by the state space statistics analysis which results that the proposed CPN model is liveness and free from deadlocks. Second, by the performance analysis in which demonstrates that the proposed model increases both the packet delivery ratio and the average end-to-end delay.

KEYWORDS

Modelling systems; Underwater Wireless Sensor Networks; Routing protocol; Coloured Petri Nets

1 INTRODUCTION

One method to solve the developing challenges of concurrent systems is to construct a model for that system [1]. Modelling is a general method used through the development of systems. Several modelling languages were proposed, and others are applied for the development of systems. Building a system model is normally accomplished at the initial stages of development.

In the literatures we found works where Petri Nets were used to model and analyze several aspects of wireless sensor networks. In [2] the authors use Petri Nets to model the validity of an encryption scheme applied to wireless sensor networks. In [3] the authors proposed a new network topology in wireless sensor networks using Petri Nets. Authors in [4] developed a detailed Petri Nets model to evaluate the energy consumption on wireless sensor networks. There are many works related to using Petri Nets in modelling and analyzing wireless sensor networks [5], [6], and [7].

In this paper, we propose a model for one of the UWSNs routing protocols, Vector-Based Forwarding (VBF) routing protocol using CPNs. We analyze this model using state space techniques derived from CPN Tools [8]. CPN Tools is a method for describing, simulating, state space and performance analysis of CPN models.

The remainder of this paper is organized as follows. In Section 2, a brief introduction of CPNs is given. Section 3 represents the functionality of the VBF routing protocol. In Section 4, we introduce the proposed CPN model for the VBF protocol. Section 5 shows the state space statistics and performance analysis of the proposed model. Finally, we draw the main conclusions and possible future works in Section 6.

2 COLOURED PETRI NETS (CPNs)

Coloured Petri Nets (CPNs) are graphical language used for building concurrent system

ISBN: 978-0-9891305-5-4 ©2014 SDIWC

models and analyzing their characteristics [1]. CPNs are a modelling language especially for discrete-event systems. It can simply merge the abilities of Petri nets [9] with the skills of a high-level programming language. The models using CPNs can be divided to a group of sub-models. This is mainly important through modelling large systems with CPNs. These sub-models can interact with each other over predefined interfaces. The sub-models in CPNs are hierarchically structured that can collect a number of sub-models to create a model and lets a model to obtain a number of sub-models [1]. This allows the modeler to design from top to down or from bottom to up when building CPN models. Through the building step several levels of abstraction can be used for modelling the CPN model of a system can take different abstraction levels for one system. This means that one CPN model can be modelled with diverse number of levels. The constructed model of a system modelled by CPN consists of states describing the system and events which make the changes of the states of that system. Simulation results of the CPN model can possibly examine different scenarios and discover the behavior of the system.

Formally, a CPN can be defined by nine-tuple [10] as:

$$\mathrm{CPN} = (\mathrm{P}, \mathrm{T}, \mathrm{A}, \sum, V, C, G, E, I) \qquad (1)$$

where P is a group of places, T is a set of transitions, A is a number of directed arcs between places and transitions, $\sum$ is a group of colour-sets, V is a set of typed variables, C is a colour-set function which specifies a colour-set for each place, G is a guard function that specifies a guard for each transition, E is an arc expression functions that assigns an expression to each arc, and I is an initialization function that specifies an initial expression of each place.

The main idea for state spaces is to describe all the reachable states and state modifications of the CPN model and then representing these changes as a directed graph, where nodes are representing the states and arcs are representing the events [9]. The construction of state spaces can be done automatically. Through the constructing state spaces, answers on an enormous number of questions about the verification and the behavior of the system can possibly found.

3 VBF ROUTING PROTOCOL

Vector-based forwarding (VBF) is a location-aware routing approach for UWSNs proposed in [11], [12]. In this protocol, data packets are forwarded from the source to the sink along interleaved routs, which helps solving the problems of packet losses and node failures. It is considered that each node previously knows its position, and each packet knows the location of all the nodes including the source, forwarders, and sink. The forwarding path is identified by the routing vector from the source to the sink. As soon as a packet is received, the node calculates its position with respect to the forwarder through measuring its distance from the forwarder and knowing the signal arrival angle. Repeatedly, all the nodes receiving the same packet calculate their positions. If a node found that it is near enough to the routing vector, it attaches its computed position in the packet and remains forwarding the packet; else, it ignores the packet simply. In this method, all the nodes forwarding the packets in the network create a “routing pipe”, the nodes in this pipe are qualified for forwarding the packet, and the nodes outside the routing pipe do not forward the packets.

Additionally, a localized self-adaptation algorithm is developed to improve the VBF performance [11]. The self-adaptation approach allows each node to estimate the density in its neighborhood and forward packets adaptively. This algorithm is based on the definition of a desirableness factor, α. This factor measures the capability of a node to forwarding the packets. Given a routing vector S_1S_0 and forwarder node F, the value of the desirableness factor of a node A is:

$$\propto = \frac{\mathrm{P}}{\mathrm{W}} + \frac{\mathrm{R} - \mathrm{d} \times \cos\theta}{\mathrm{R}} \qquad (2)$$

where P is the projection length of A on the S_1S_0 routing vector, W is the routing pipe radius, R is the transmission range, d is the distance between A and F, and θ is the angle between vector FS_0 and vector FA.

Fig. 1 represents the different variables used in the description of the desirableness factor [11]. From the definition, it is seen that for each node near enough to the routing vector, i.e., inside the pipe ($0 \leq P \leq W$), the range of the desirableness factor value of this node is in [0, 3] depending on position of node A.

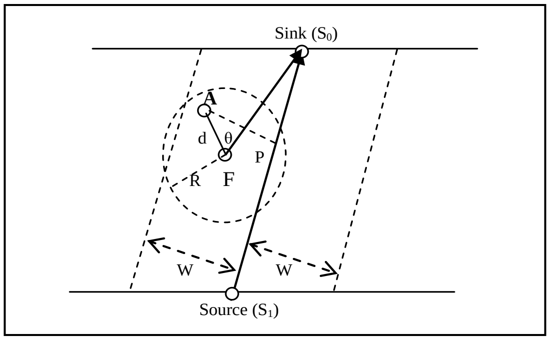

Figure 1. Desirableness factor in self-adaptation algorithm

In this algorithm, when a packet received by a node, it first determines if it is eligible for packet forwarding [11], [13]. If yes, this node then retains the packet for a time interval. The duration of this time depends on the value of the calculated desirableness factor and other network parameters. This means that, each eligible node delays packet forwarding by a time interval. Mostly, the self-adaptation algorithm provides the desirable node a higher priority to continue packet forwarding. The theoretical analysis can be found in [11].

4 THE PROPOSED CPN MODEL OF VBF PROTOCOL

In this Section we discuss how to build the VBF routing protocol model by CPNs, according to a given routing technique as in Section 3. Every place in the CPN model is described by a set of values, called a colour set that identifies the values of tokens allowed to mark that place [14].

Each sensor node is modelled by a CPN that describes the ability of the node to forward data packets to the correct forwarder. In this work, we only model the routing of the data packets from the source to the sink. The following subsections shows the assumptions needs to be taken into consideration when creating the model and proposes the structure of the CPN model for VBF routing protocol.

4.1 Model Assumptions

We use the software tool, CPN Tools [8], [9], when creating our VBF routing protocol using CPN model, we select the following assumptions:

1. The sensor nodes are distributed randomly in a volume of 600 × 600 × 600 m^3. They can move in the X-Y dimensional space.
2. The source and the sink nodes are fixed and the number of sensor nodes is 30 nodes.
3. The node speed is 2 m/s.
4. The transmission range of the nodes 100 m and the routing pipe radius is 100 m.

4.2 Model Structure

To construct our model, we follow the hierarchically structuring mechanism. As a first step, the model is described in top level of abstraction, in which the main parts of the model are presented. As shown in Fig. 2, the main states of the VBF routing protocols are summarized in three sub-models: a sub-model for the *sender*, a sub-model for the *forwarder*, and a sub-model for the *receiver*. The model contains three substitution transitions (plotted as double border rectangular boxes), sender node, forwarder node, and receiver node and three places generating the interface that the sub-models can exchange tokens with each other.

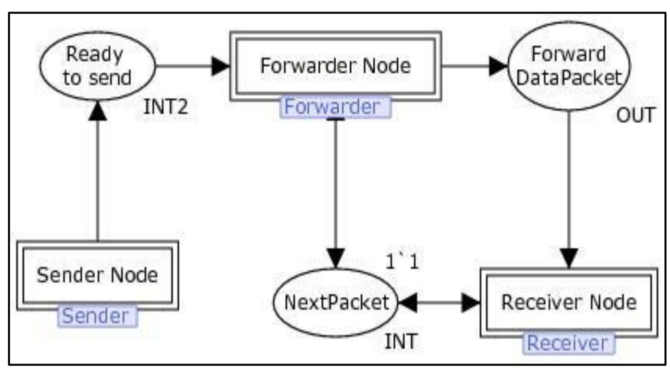

Figure 2. CPN Main Model

The CPN model execution is defined through an occurrence sequence, which identifies the occurred steps and the reached in-between markings. The marking that is reached by an occurrence sequence and beginning at the initial marking is called a reachable marking.

Figure 3 shows the CPN sender sub-model, which contains four transitions and twelve places. Here, the parameters of the node such as its co-ordinates, crd, energy, E, speed, S, Projection, Pro, angle between this node, and the forwarder node, Theta and transmission range, R, are described. Transition *GetDistance* is fed with the sender and forwarder co-ordinates to calculate the distance, d, between them after firing this transition. Both the sender and the forwarder are lying in the routing pipe with radius W.

The *Update-Energy* transition is fed with the node energy and returns the value of the updated energy which is depending on the state of the node: transmit, receive, or idle state. Place *Ready to send* is an output port to the next sub-model, forwarder sub-model. All the places are typed by the colour-set identified beside each place which represents the types of tokens holed by that place. Arcs between transitions and places are described by expressions written on the arc, as shown in the CPN Sender sub-model.

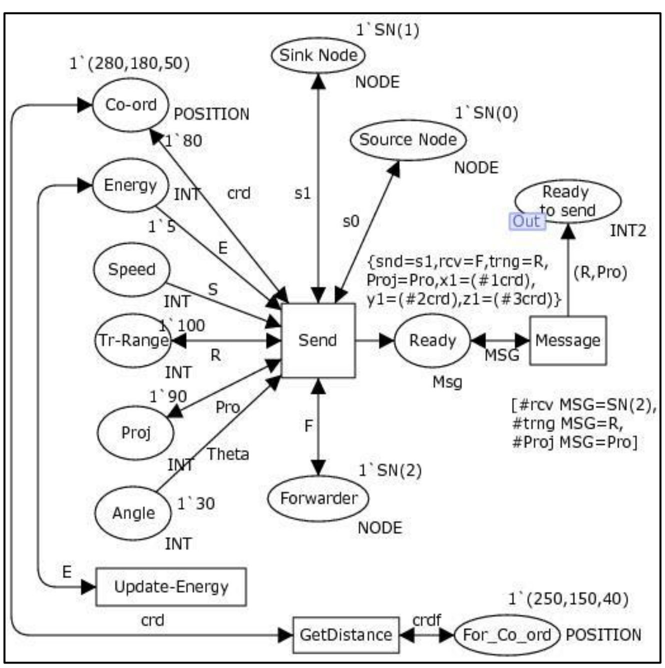

Figure 3. CPN Sender sub-model

Figure 4 depicts the CPN forwarder sub-model, which contains four transitions and nine places. Here, the desirable factor, α, as in (2) of the node is computed in order to choose the forwarder that has minimum α and allows this node to forward the data packets. Transition *check pipe's node* ensures that the forwarder node is inside the pipe. If yes, the transition *Check Position in R* checks that the forwarder position in the transmission range of the sender. Else, this node is avoided by place *Avoided Node.* Transition *Calculate desirable factor* is fed with the sender transmission range, the distance and angle, θ, between sender and forwarder, and the projection, P, of the forwarder on the vector between the source and the sink to calculate α. Place *Packets* holds the initial marking that contains all the data packets need to reach the sink through the forwarders. The initial marking of the place *Packets* is defined on the top of place in the *AllPackets* variable represented in Fig. 4

Place *Ready to send* is an input port from the sender sub-model where places *Forwarder Node* and *NextPacket* are output ports to the next sub-model, receiver sub-model.

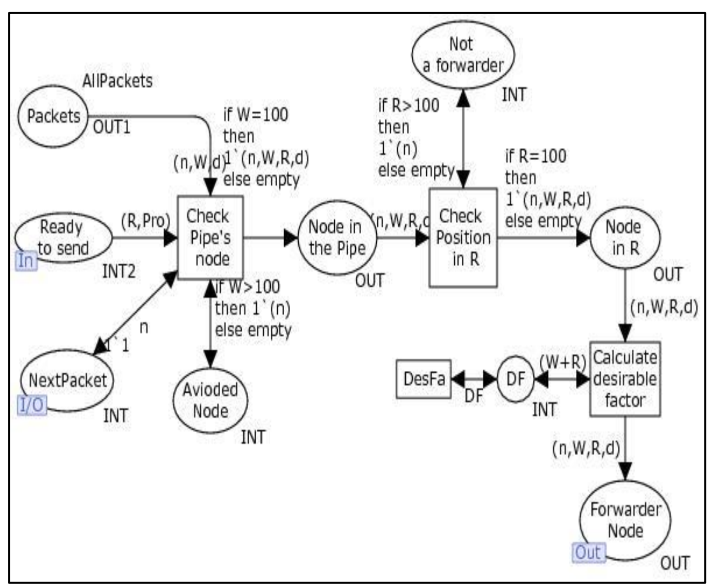

Figure 4. CPN Forwarder sub-model

Figure 5 shows the CPN receiver sub-model, which contains two transitions and five places. In this sub-model the forwarder node receives the first Data Packet from the sender by transition

Receive Packet and sends an acknowledge back to the sender in order to sends the next data packets. Place *Data Received* receives all data packets. All the arcs used in the represented CPN models are directed and contain all the conditions used to firing there assigned transitions or reaches the corresponding places.

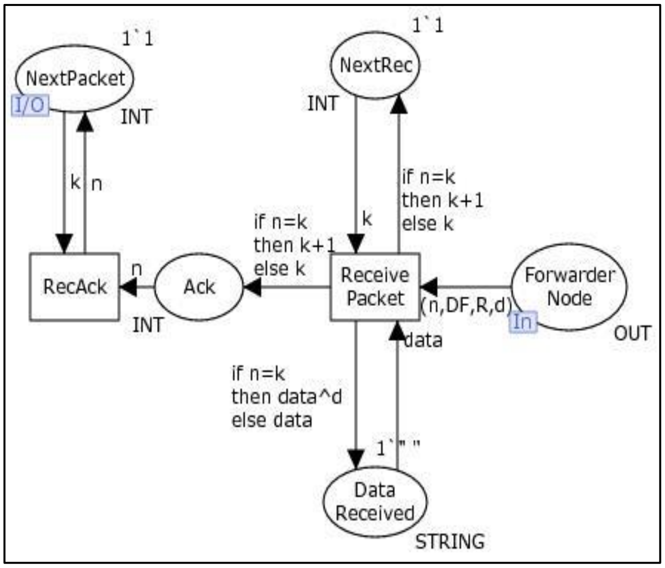

Figure 5. CPN Receiver sub-model

In the hierarchical CPN model existing in the previous three figures, 2, 3, 4, and 5, we found two levels of abstraction. The highest level of abstraction is the Main model, while the lowest level of abstraction is the Sender, Forwarder, and Receiver sub-models. In general, there can be different levels of abstraction [10].

5 STATE SPACE ANALYSIS RESULTS

The state space analysis gives some basic information about the state space size and the behavioral properties of the CPN model. From the statistics displayed in Fig. 6, it is seen that CPN model of the VBF routing protocol have 6822 nodes and 9385 arcs. The full state space construction took 150 seconds. The full state space generation is followed by producing the Strongly Connected Component graph (SCC graph) [10]. The nodes in the SCC graph are called Strongly Connected Components (SCCs) and are achieved by making a disjoint division of the state space nodes, such that two state space nodes are in the same SSC if and only if they are commonly reachable. In our model, we also get the SSC graph statistics shown in Fig. 7. It has 4796 nodes, 5195 arcs, and was computed in 3 seconds. The difference between the number of nodes in the state space and the number of nodes in the SCC graph statistics inform us that there exist cycles in the state space of the CPN model. This indicates that we have unlimited occurrence sequences and the VBF routing protocol may not terminate.

State Space Statistics

State Space	SCC Graph
Nodes: 6822	Nodes: 4796
Arcs: 9385	Arcs: 5195
Seconds: 150	Seconds: 3
Status: All	

Figure 6. State space statistics of the CPN model

The *Home properties* specify the number of the home marking node. The home marking is a marking that can be reached from any other reachable marking [10]. While the *liveness properties* identify the node number of the dead marking. The dead marking is a marking that have no enabled binding elements. From Fig. 7 we find that the marking generated by node 3486 is considered to be a home and also a dead marking. This fact tells us that the routing protocol identified by the CPN model is partially correct. Moreover, because node 3486 is also a home marking it is constantly possible to terminate the routing protocol with the right result. Figure 7 also states that there are no *live transition* and no *dead transitions* [10]. We have already realized that our routing protocol contains a dead marking, and this resulting the absence of live transition, which means that there is no transition can be fired from the dead marking. The absence of dead transitions in the protocol means that each transition has the probability to occur or fire at least once. Finally, the CPN model for the VBF routing protocol is liveness and free from deadlocks.

ISBN: 978-0-9891305-5-4 ©2014 SDIWC

Home Properties

Home Markings: [3486]

Liveness Properties

Dead Markings: [3486]
Dead Transitions: None
Live Transitions: None

Figure 7. State space behavioral properties of the CPN model

Table 1 shows the state space analysis results with the increasing of the number of the forwarder nodes between the sender and the receiver. It is seen that the state space size rises with the increasing of the number of forwarder node. Furthermore, in each case, the number of nodes and arcs in the SCCs is less than the number of nodes and arcs in the state space, indicating the occurrence of cycle behavior in the routing protocol.

Table 1. State space analysis results as increasing the number of forwarder nodes

Properties	Number of Forwarder Nodes				
	10	***20***	***30***	***40***	***50***
State space nodes	2340	4001	6822	8431	10792
State space arcs	4314	7832	9385	10340	12850
State space time (sec)	60	105	150	200	250
SCCs-graph nodes	2114	3871	4796	6355	8644
SCCs-graph arcs	1907	2154	5195	5350	9610
SCCs-graph time	1	2	3	5	7
Number of the sent data packets	500	500	500	500	500
Number of the received data packets	445	446	456	455	465
Average end-to-end delay (sec)	4.002	8.723	10.832	13.456	14.355

The number of the sent and received data packets and the average end-to-end delay are also reported in Table 1. These results are averages over 20 runs with an arbitrarily generated network topology. The state space statistics analysis results that the proposed CPN model is liveness and free from deadlocks.

5.1 Performance Analysis

Another kind of output can be generated from CPN model which is the report simulation performance. It holds informations that are computed for the data packets that has been sent by the sender or received by the sink. Performance is quantified through measures of packet delivery ratio, and average end-to-end delay [15]. The packet delivery ratio is the fraction of the number of packets successfully received by the sink to the number of packets produced by the source. The average delay is the average end-to-end delay for each packet received by the sink. Simulation is performed by the underwater package Aqua-Sim of ns-2 [16], [17]. In the simulations, the parameters are set similar to UWM1000 LinkQuest Underwater Acoustic Modem [18] and take into consideration the same model assumptions considered in Section 4.

Fig. 8 identifies the proposed CPN model packet delivery ratio with the number of forwarder nodes. The values used for charting this figure are computed from the results of Table 1. It is realized that the packet delivery ratio rises with the increase of the number of sensor nodes. When more than 30 nodes are deployed in the space, the value of the packet delivery ratio remains above 90% for both VBF protocol simulated by Aqua-Sim and the VBF protocol modeled by CPNs. Table 2, extracted from Fig. 8, shows that our CPN model gives better results in packet delivery ratio.

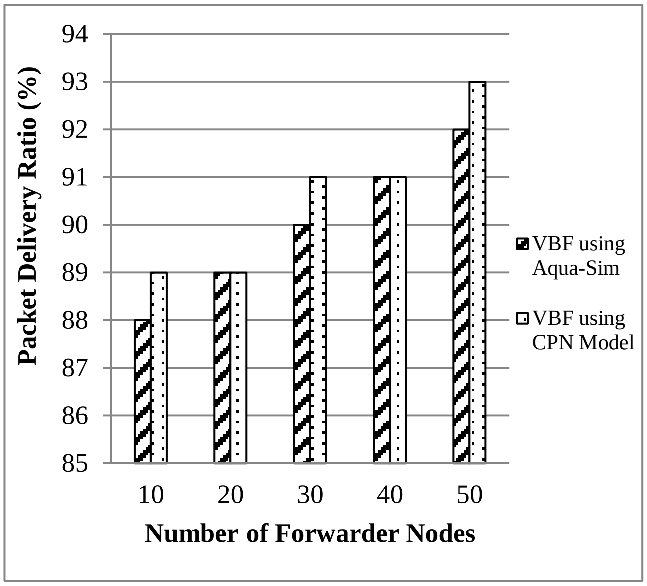

Figure 8. Proposed CPN model packet delivery ratio *vs.* number of forwarder nodes

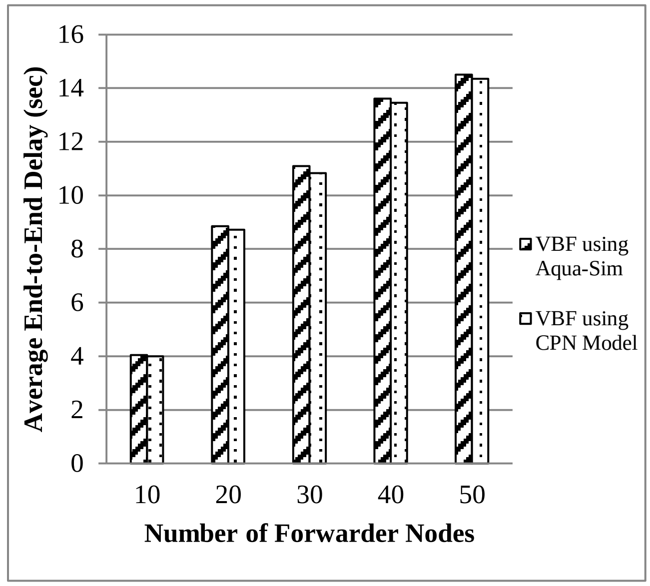

Figure 9. Proposed CPN model average end-to-end delay *vs.* number of forwarder nodes

Table 2. Increase in Packet Delivery Ratio

Number of Forwarder Nodes	Packet Delivery Ratio (%)		
	VBF using Aqua-Sim	*VBF using CPN model*	*Increasing percentage*
10	88	89	1.13%
30	90	91	1.11%
50	92	93	0.98%

Table 3. Reduction in end-to-end delay

Number of Forwarder Nodes	Average End-to-End Delay(sec)		
	VBF using Aqua-Sim	*VBF using CPN model*	*Reduction percentage*
10	4.049	4.002	1.16%
30	11.095	10.832	2.37%
50	14.598	14.355	1.66%

Fig. 9 describes the proposed CPN model average end-to-end delay with the number of forwarder nodes. The values used for plotting this figure are taken from the results of Table 1. It is seen that the average end-to-end delay reduced with the increase of the number of forwarder nodes in the network. By increasing the number of sensor nodes the paths from the source to the sink are nearer to the optimal path (α=0); hence, the average end-to-end delay decreases, as shown in Table 3, extracted from Fig. 9.

6 CONLUSIONS

In this paper, we present a CPN model for the VBF routing protocol in UWSNs that allows the study of several network properties. Our proposed model is verified by two phases: First, by the state space statistics analysis which results that the proposed CPN model is liveness and free from deadlocks. Second, by the performance analysis in which demonstrates that the model efficiently increases the packet delivery ratio and reduces the average end-to-end delay as the increasing of the number of forwarder nodes in the routing protocol. As a future work, it is expected to use CPNs to model other features of the routing protocol in the underwater wireless sensor networks in order to cover all the network properties.

7 REFERENCES

[1] K. Jensen and L. M. Kristensen, "Coloured Petri Nets: Introduction to modelling and validation," Springer-Verlag, 2009.

[2] H. Rodriguez, R. Carvajal, B. Ontiveros, and I. Soto, "Using Petri Net for Modelling and Analysis of a Encryption Scheme for Wireless Sensor Network," Pawel Pawlewski (Ed.), InTech, 2010.

[3] S.-Y. Kim, O. Guzide, and S. Cook, "Towards an Optimal Network Topology in Wireless Sensor Networks," on Proc. SUJUR Conf., vol. 1, pp. 55-60, 2009.

[4] A. Shareef and Y. Zhu,"Energy Modeling of Wireless Seneor Nodes Based on Petri Nets," Inter. Conf. on Parallel Processing, pp. 101 – 110, 2010.

[5] S. Chen, Q. Ge, Q.-M. Shao, and Q. Zhu, "Modelling and Performance Analysis of Wireless Sensor Network Systems Using Petri Nets," The 23rd Inter. Tech. Conf. on Circuits and Systems, Computers and Communication (ITC-CSCC), pp. 1689-1692, 2008.

[6] B. Lacerda and P. U. Lima, "Petri Nets as an Analysis Tool for Data Flow in Wireless Sensor Networks," Proc. of CNRS 2011 - 1st Portuguese Conf. on WSNs, Coimbra, Portugal, pp. 1-6, 2011

[7] J. Ben-Oth, S. Diagre, L. Mokdad, and B. Yahya, "Performance Evaluation of a Medium Access Control Protocol for Wireless Sensor Networks using Petri Nets," in Proc. of HET-NETs Conf. , pp. 335-354, 2010.

[8] CPN Tools, www.daimi.au.dk/CPNTools/.

[9] K. Jensen, L. M. Kristensen and L. Wells, "Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems," Inter. Journal on Software Tools for Technology Transfer (STTT), vol. 9, no. 3-4, pp. 213-254, Springer-Verlage, 2007.

[10] K. Jensen and L. M. Kristensen, "Coloured Petri Nets: Modelling and validation of concurrent systems," Springer-Verlag, 2009.

[11] P. Xie, J.-H. Cui and L Lao, "VBF: Vector-based Forwarding Protocol for Underwater Sensor Networks," Inter. Conf. on networking (IFIP networking), pp. 1216-1221, 2006.

[12] R. Thumpi, R.B. Manjula, and S.M. Sunilkumar "A Survey on Routing Protocols for Underwater Acoustic Sensor Networks " Inter. journal of recent technology and engineering (IJRTE), vol. 2, May 2013.

[13] N. Nicolaou, A. See, P. Xie, J.-H. Cui, and D. Maggiorini, "Improving the Robustness of Location-based Routing for Underwater Sensor Networks," Proc. Of the OCEANS'07, pp. 1-6, Europe, June 2007.

[14] J. Billington, A. Gupta, and G. Gallasch, "Modelling and Analysing the Contract Net Protocol-Extension Using Coloured Petri Nets," Inter. Federation for Information Processing (IFIP), pp. 169-184, 2008.

[15] W. Bo, L. Yong-mei, J. Zhigang, "ES-VBF: An Energy Saving Routing Protocol," Proc. of the 2012 Inter. Conf. on Information Technology and Software Engineering, pp. 87-97, 2012.

[16] NS-2: Network Simulator, http://www.isi.edu/nsnam/ns.

[17] P. Xie, Z. Zhou, Z. Peng, H. Yan, T. Hu, J.-H. cui, Z. Shi, Y. Fei, and S. Zhou, "Aqua-sim: A Ns-2 based Simulator for Underwater Sensor Networks," Proc. of IEEE/MTS OCEANS, pp. 1-7, 2009.

[18] LinkQuest, http://www.link-quest.com.