Volume 4, Issue 9, September 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Analysis of AODV Routing Protocol and Use of Colored Petri Net in AODV Routing Protocol

Meenakshi Dewangan*, Aakanksha Choubey
Department of Computer Science & Technology,
CSVTU, Bhilai (C.G), India

Abstract: As the growth of technology in lives of humans increases, the growth of interest and research on mobile ad-hoc networks is exponentially. Therefore in Mobile Ad-hoc NETwork (MANET), Ad-hoc On demand Distance Vector (AODV) Routing protocol is used, in which wireless transmission takes place where one mobile node can send messages directly to other mobile node. The node movement in the dynamic environment causes frequent topology changes in the network. Thus it is very much necessary for every node in the network to keep track of change so that an efficient packet transmission can be done. Colored petri nets is one such formal method in which models depicting the exact functionality of the system is being built, simulated and analyzed.

Keywords: AODV, Colored petri net tool, Mobile Ad-hoc Network, routing protocol, TPN.

I. INTRODUCTION

Wireless networks are classified as: Infrastructure-based wireless network and Infrastructure-less wireless network. In an Infrastructure-based wireless network (e.g. GSM networks and WLANs) nodes connect to an external network like Internet or Intranet with the help of an access point. On the other hand an Infrastructure-less network is a network in which mobile nodes communicate with each other through wireless links, such a network is also known as an Ad-hoc network. For example two laptops with wireless adapter cards can set up an Ad-hoc network. A wireless ad hoc network is a decentralized type of wireless network. The network is ad hoc because it does not rely on a pre existing infrastructure, such as routers in wired networks or access points in managed (infrastructure) wireless networks. An ad hoc network typically refers to any set of networks where all devices have equal status on a network and are free to associate with any other ad hoc network device in link range.

II. RELATED WORK

AODV routing protocol is jointly developed in Nokia Research Center, University of California, Santa Barbara and University of Cincinnati by C. Perkins, E. Belding-Royer and S. Das. Coloured Petri Nets have been used by some of the researchers for validating and modeling some of the features of the mobile ad hoc networks. Chinara et al. [8] have proposed the validation of neighbor detection protocol for ad-hoc network by using the CPN tools. Erbas et al. [4] proposed a two designed position based routing approach based model on Colored Petri Nets for mobile ad-hoc network. Here the author shows that the multicast routing protocol delivers better result than the basic ODMRP (On Demand Multicast Routing Protocol). This model (CPN model) developed reliable unicast and multicast routing method based on geographical position of a node. Kodikara et al. [13] proposed the simulation model of context exchange based on hierarchical coloured petri nets. Simulation of vertical communication model has done which is cross layer info exchange module of context exchange (conEX).State space used for verification of Petri nets dynamic behavioral properties like liveness, home property, boundedness and fairness. The approaches of both the papers ([9] and [10]) are almost same. mohamed et al The paper [19] addresses the mobility peoblem. Colored petri nets used as a simulation tool , which simulate the network without knowing the network topology. Colored petri net is a tool which provides great perceptivity to modeling a mobile ad-hoc network protocol. The author finds the route in the simulation part. In [18], ad hoc on demand distance vector routing protocol (AODV) has been modeled using Design/CPN as the simulator tool. They have proposed a topology approximation (TA) method to address the problem of mobility in MANET and used this mechanism to build a CPN model of a MANET. In [1], colored petri nets along with their various properties and applications are discussed. A brief introduction to Protocol Engineering and at what stage Colored Petri Nets are used in Protocol Engineering is also presented. In [5], This paper introduces random waypoint mobility model for describing simulation of mobile nodes distribution within then network. Before simulation, UML (Unified modeling language) modeling is done to describe the structural view of a model and also for error correction. The simulation process has been done with the help of hierarchical coloured petri net which is extension of normal coloured petri nets. In [2], Alessandro Bianchi, Sebastiano Pizzutilo proposes a tool, which allows both formally modeling and simulating a Mobile Ad-hoc NETwork – MANET. They developed the prototype of moDEling MObile NEtworks (DEMONE) is able to simulate MANET behavior as well as other simulators: DEMONE has been applied to 3000 simulations. DEMONE allows the formal description of MANET through Petri Nets, so it can be used for specifying MANETs protocols and services as well as for studying performance. In [3], this paper has reviewed the working of DYMO Routing Protocol in comparison with the existing AODV Protocol. The overall study shows that DYMO is a better protocol when it comes to networks with high mobility and changing topology, moreover its performance outperforms the conventional AODV protocol when it comes to large networks with large number of nodes and changing topology. In [11], this paper presented and discussed model checking techniques of time Petri nets. For model checking LTL properties, they proposed a contraction for the state class graph (SCG), called RSCG, which is both smaller and faster to compute than other abstractions. For CTL* model checking, they showed that refining abstractions contracted by inclusion or convex-combination allow to improve significantly the refinement process. In 2013, Vishal Pahal, Amit Verma, Payal Gupta published a paper titled Classification of Routing Protocol in Mobile Ad Hoc Networks: A Review. This Paper deals with number of ways of categorization of protocol and also present some specified protocols according to that classification. The emphasis of this paper is not to present protocol in detail but present main feature of wide variety of different protocols and discuss their suitability. In [19] paper, published by Lisa Wells, has described new facilities that are fully integrated in CPN Tools and that support simulation-based performance analysis using CP-nets. The facilities include support for collecting data during simulations, for generating different kinds of performance-related output, and for running multiple simulation replications.

III. MOBILE AD HOC NETWORK (MANET)

Mobile ad hoc network (MANET) is an autonomous system with no pre existing infrastructure or centralized administration. Nodes in a MANET communicate through wireless channels, without any predefined physical infrastructure. They are autonomous agents and they can dispose without according to a predefined topology. During their lifetime, hosts enter or leave the network, and continuously change their relative position, so the network must quickly adapt to changes. The system itself is the unique responsible of setup, communication management and adaptation.

A. Classification of Routing Protocol:

Routing Protocols can be classified into three categories:
1) Reactive (On-demand),
2) Proactive (Table-driven), and
3) Hybrid.

1) Proactive or Table-Driven Routing Protocol:

The proactive routing protocols are table-driven3. They usually use link-state Routing algorithms flooding the link information. Link-state algorithms maintain a full or partial copy of the network topology and costs for all known links. In these protocols, each node maintains routing information to every other node in the network. The routing information is usually kept in number of different routing tables. These tables are periodically updated if the network topology changes. Some of the most used on proactive routing protocols are DSDV (Destination-Sequenced Distance-Vector Routing), WRP (Wireless Routing Protocol). The main disadvantages of Proactive Routing protocols [3] are:

- Wastage of bandwidth due to unnecessary advertising of routing information.
- Maintaining a routing table for each node and advertising of this table leads to overhead, which consumes more bandwidth.
- Regular update of its routing tables uses up battery power.
- Slow reaction on restructuring and failures.
- Many redundant route entries to the specific destination needlessly take place in the routing tables.

2) Reactive or On-Demand Routing Protocols:

In Reactive routing protocols, when a source wants to send packets to a destination, it invokes the route discovery mechanisms to find the route to the destination. The route remains valid till the destination is reachable or until the route is no longer needed. Unlike table driven protocols, all nodes need not maintain up-to date routing information. Some of the most used on demand routing protocols are DSR (Dynamic Source Routing), DYMO(Dynamic MANET On Demand) and AODV(Ad-hoc On Demand Distance Vector Routing). The main disadvantages of Reactive Routing protocols [3] are:

- High latency time is required in finding the route to the destination,
- Flooding to can lead to network clogging.
- RREP, RREQ & RERR messages leads to Control overhead.

3) Hybrid Routing Protocol:

Hybrid routing protocol combines the advantages of both proactive and reactive routing protocols. The routing is initially established with some proactively prospected routes and then serves the demand from additionally activated nodes through reactive flooding. Some of the existing hybrid protocols are ZRP (Zone Routing Protocol). The main disadvantages of Hybrid Routing Protocols are [3]:

- Large overlapping of routes.
- Longer delay if route not found immediately.
- Core nodes movement affects the performance of the protocol.

B. Ad hoc On Demand Distance Vector Routing Protocol:

AODV stands for Ad-hoc On Demand Distance vector Routing Protocol. It is a reactive/on demand routing protocol means route discovery process is started only when source node raises the demand for it. AODV is a self-starting and dynamic algorithm where the large number of nodes can participate for establishing communication and maintaining AODV network. The topology of AODV changes time to time as the nodes are not fixed to any standard position. In AODV, all nodes maintain a routing table containing the entry for each destination node. Each entry includes the next hop, sequence number and number of hops requires for reaching destination node. Using the destination sequence number ensures loop freedom. AODV makes sure the route to the destination does not contain a loop and is the shortest path. Route request and route reply query cycle is used in order to build the routes in AODV routing protocol. Whenever there is a link breakage, affected nodes are notified so that they invalidate the routes using that link. Route Requests (RREQs), Route Replay (RREPs), Route Errors (RERRs) are control messages used for establishing a path from source to the destination.

1) *AODV Process*:

As AODV is reactive protocol i.e. the route is created only when it is required. Packet transmission in AODV routing protocol can be multi hop i.e. A node can send packet to another node beyond its transmission range using other nodes as relay point and thus a node can function as a router. Each node in the network maintains its own neighbor table. If a node N wants to send a data packet, it first checks its routing table to find whether there is an existing path to the destination node or not. If it has, it sends the data packet along this route immediately. Otherwise the route discovery procedure starts which is as follows:

- *Route Discovery*: If the route between source and destination is not available, a RREQ (Route Request) packet is broadcasted throughout the network as shown in the Figure 1 below [3].

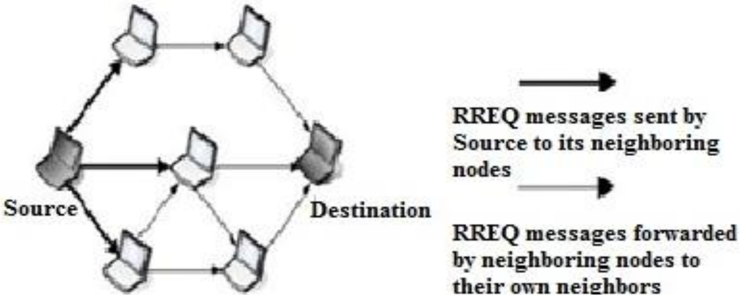


Figure 1 Route Discovery process: via RREQ Messages

As soon as a node receives a RREQ packet, it first checks whether it has received this packet earlier. If yes then the node simply discards the packet and if not, a reverse routing entry towards the originator of RREQ packet is created. This route can be used to forward route reply later on. If any intermediate node has a valid route towards the destination node, it unicasts a RREP(Route REPly) packet towards the source node. A node on receiving RREP packet creates a reverse route entry towards the originator of RREP packet as shown in figure 2. [3].

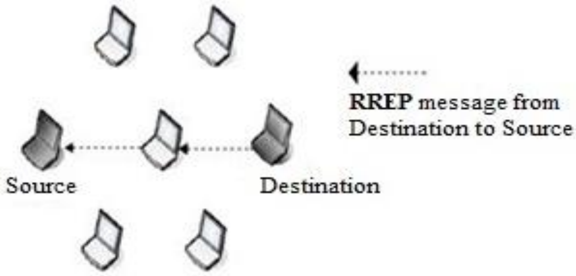


Figure 2. Route Discovery process: via RREP Messages.

- *Route Maintenance*: Every node in the network periodically broadcasts HELLO messages to its neighbors to indicate its presence. If the node does not receive a HELLO message from its neighbor then it marks that particular route as invalid and the node is considered to be exhausted or moved away from the network. Figure 3 shows the broadcasting of HELLO message.

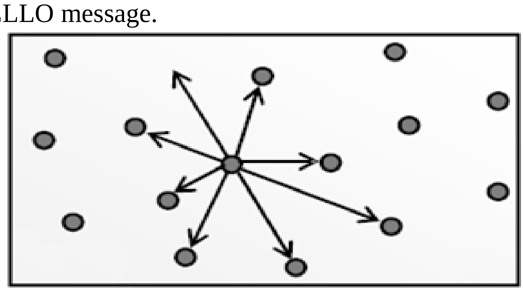


Figure 3. Broadcasting of Hello Messages.

Hence route table is updated and a RERR (Route Error) packet is sent to all the affected nodes linked with that particular node as shown in figure 4 [3]. Every node in the network maintains a sequence number to ensure that no loop exists among the nodes. This sequence number is incremented by the node every time a packet is sent and is stored along with the route information in the route table. It is sent along with RREQ (for source) and RREP (for destination). A node with larger sequence number is always preferred since it indicates the most recent path to the destination.

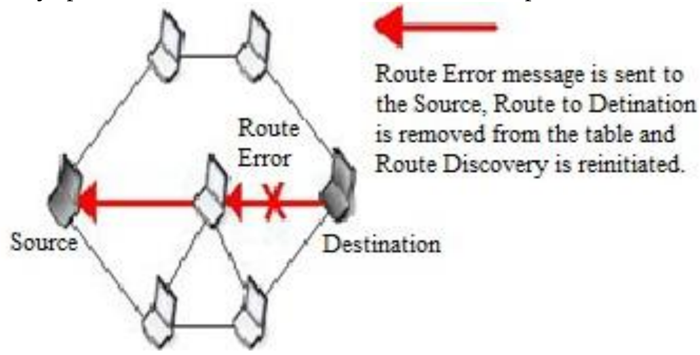


Figure 4 Route Maintenance process: via RERR Messages

Whenever a source node was to send data packet, routing table is checked for an unexpired entry, which if exists, packet is transmitted directly by using that route otherwise RREQ packet is broadcasted. This packet contains source ID, broadcast ID, sequence number of source, sequence number of destination, previous ID, hop count and destination ID. The purpose of destination sequence number is to prevent the loop in the route discovery process. Every node has its own sequence number which is increment by one every time there is a link breakage [1]. A node upon receiving this packet checks whether it has received this packet earlier, if it has it simply discards the packet otherwise a RREP packet is unicasted to the source along the reverse path to that of RREQ packet [21]. When an intermediate node receives this RREP packet, it sets a forward path entry to the destination. AODV protocol has following 4 states:

1. Routcheck: check the routing table to find if the source node has an unexpired path to the destination node. This is done using the guard has validRoute().
2. RREQInit: Initiate RREQ message when necessary. Its main function is to direct the RREQ message and rebroadcast it using the function arc rebroadcast() if necessary.
3. RREQProcess: Process the RREQ message received and output proper results. Its function is to initiate RREP message if possible or forward RREQ message if necessary. This functionality is achieved by arc newBID() and arc initiateRREP().
4. RREPProcess: Process the incoming RREP message and output proper result. The main function of this subpage is to update the route table and forward the RREP message if necessary. To achieve these functionalities, two functions arc updateRoute() and arc forwardRREP() are used. RREQInit, RREQProcess and RREPProcess are three substitution transitions. If a node wants to send data packet, it first enters Routcheck state to check for an existing path. If there is no existing route in the routing table, the node enters RREQInit state and initiates route discovery process i.e Broadcasting of RREQ packet.

IV. COLORED PETRI NET

Colored Petri Net provides a framework for construction and analysis of concurrent and distributed systems. Colored Petri Nets [1] is a graphical oriented language for design, verification and validation of systems. It is mainly used in systems where concurrency, communication and synchronization are important. CP-nets is a discreet event modeling language combining the capabilities of Petri nets with the capabilities of a high level functional programming language Standard ML. Petri nets give the basis of graphical notation and primitives for modeling concurrency, communication and synchronization. The strength of CPNs over traditional Petri Nets is that, it supports hierarchy, colour, and time in the model. Hierarchy in the CPNs indicates that the models can be structured in a number of related modules. This concept is based on the concept of hierarchical structuring of the programming language, that supports the bottom-up or top-down style.

A. Need of Colored Petri Net:

With the explosive involvement of technology in the lives of humans, it is extremely important to design such systems which are free from errors and are able to satisfy their users in terms of performance, efficiency, correctness, ease of use etc. Such systems may be applied in various applications in day to day use such as communication. If due to certain reasons, these systems are designed and hence implemented incorrectly, the cost of correcting these errors becomes huge. So, it is of utmost importance to have a mechanism where design errors can be uncovered and corrected early. Colored petri nets is one such formal method in which models depicting the exact functionality of the system is being built, simulated and analyzed. This is a promising technique with a lot of research done to prove the correctness of a vast variety of systems. With the review of work done in CPN so far, it has been found that CPN can be successfully applied to model and verify a vast range of systems. Communication protocols, particularly ad hoc routing protocols are also one of the systems which has been modeled and analyzed to prove the availability of certain properties and to discover the existing errors.

B. Properties of Colored Petri Net:

Some basic properties [1] of CPN are:

1) *CP-nets have a graphical representation:* The graphical form is very easy to understand and grasp. Even the people who are not very familiar with the details of CP-nets can easily understand it. An algorithm or a communication protocol can be presented by drawing a directed graph, where the nodes represent states and actions, while the arcs describe how to go from one state to another, by executing some of the actions.
2) CP-nets have a well-defined set of rules which uniquely defines the behavior of each CP-net.
3) CP-nets are very general and can be used to describe a large variety of different systems: CPnets are applicable in informal systems like description of work processes and formal systems like communication protocols. They are also applied to distributed algorithms, VLSI.
4) CP-nets have very few, but powerful, primitives.
5) *CP-nets have a separate description of both states and actions:* This differs from most system description languages where either the states or the actions are described but not both.
6) *CP-nets have rules which work on concurrency instead of interleaving:* This means that the notions of conflict and concurrency can be defined in a simple way. In an interleaving semantics it is impossible to have two actions in the same step, and thus concurrency only means that the actions can occur after each other, in any order. In our opinion, true-concurrency semantics is easier to work with – because this is the way human beings usually think.
7) *CP-nets offer hierarchical descriptions:* This means that we can relate smaller CP- nets and construct large CP-net in a well-defined way. This is similar to that of functions, procedures and modules of programming languages. The hierarchical CP-nets enable us to model very large systems in a manageable and modular way.
8) *CP-nets can be extended with a time concept*: This means that it is possible to use the same modeling language for the specification/validation of functional/logical properties (such as absence of deadlocks) and performance properties (such as average waiting times).
9) *CP-nets can be simulated in an interactive manner:* where the results are presented directly on the CPN diagram.
10) CP-nets have a large number of formal analysis methods by which properties of CP-nets can be proved: There are four basic classes of formal analysis methods: construction of occurrence graphs (representing all reachable markings), calculation and interpretation of system invariants (called place and transition invariants), reductions (which shrink the net without changing a certain selected set of properties) and checking of structural properties (which guarantee certain behavioral properties).
11) CP-nets have computer tools which support their drawing, simulation and formal analysis.

C. Components of CPN:

CPN model of a system consists of:-

1) *Place:* Ellipses/circles called places which depict the various states of the system.
2) *Transition*: Rectangles called transitions which depict the various actions or events or processing that the system will undergo.
3) *Arcs:* Arcs running between places and transitions describe the flow of tokens and tell how actions modify the state and when they occur.
4) *Token:* Each place contains a set of markers called tokens. Each of these tokens carries a data value, which belongs to a given type. Places may contain any non-negative number of tokens.
5) *Marking:* A distribution of tokens over the places of a net is called a marking. The initial state of a place is denoted as its initial marking. It is usually written in the upper left or right of the place as shown in figure 5. Every place in coloured petri net is connected to a transition which fires the tokens from one place to another. No two places and transitions can be connected.

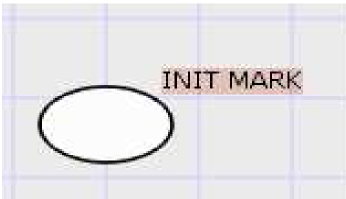


Figure 5: Place inscription: initial marking

6) *Arc Expression:* The Arc Expression describes how the state of the CP-net changes on the occurrence of a transition. place and transition are connected through arcs and an arc has an inscription which may contain a condition or an expression as shown in Figure 6.

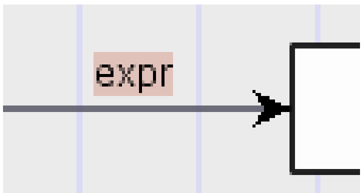
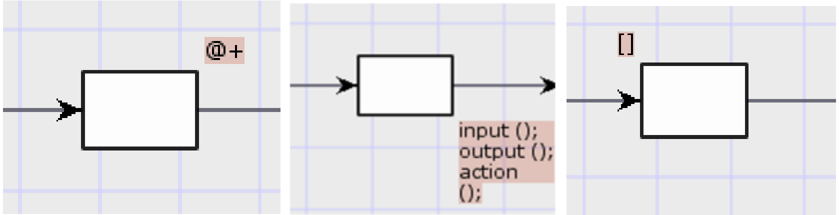


Figure 6: Arc inscription

The inscriptions of transition are shown in Figure 7. Figure 7(a) shows how time values are added to the transition. Guard functions are written in top left corner of the transition as shown in Figure 7(c). Figure 7(b) shows the code segment where the inputs and outputs are written and the code written action part is performed whenever a transition is enabled.



(a) Time inscription (b) Code segment (c) Guard

Figure 7: Transition inscriptions

D. Enabling a Transition:-

The execution of a CPN consists of occurrence (firings) of transitions. A transition can occur if it is enabled and it is enabled when the following conditions are true [18]:

1) Each of the input places of the transition contains sufficient tokens as required by the corresponding input arc expression. When the transition has variables, there must exist a binding such that each evaluated input arc expression with this binding is satisfied by the tokens available in the corresponding input place.

2) The guard[1] of the transition (if any) is evaluated to true with the same binding used above to evaluate input arc expressions. A green aura is created around the rectangular box whenever a transition is enabled as Figure 8.

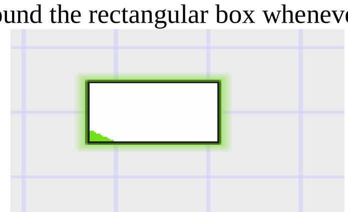


Figure 8: Enabling of a transition

When a transition (or binding element) is enabled, it is ready to occur (fire). When it occurs, two actions are taken:

1. Tokens are consumed from each of the input places of the transition as specified by the corresponding (evaluated) input arc expressions.
2. Tokens are created in each of the output places of the transition as specified by the corresponding output arc expression. If it contains variables, then the tokens created for the output place to which this arc is attached are determined by the evaluated arc expression with the same binding as that used to evaluate the input arc expressions and the guard.

A CPN moves to a new marking (i.e. state) after the occurrence of a transition.

V. CONCLUSIONS

The nodes in AODV keep changing their position dynamically thus making it difficult to model the system. Colored Petri Nets has been proved to be a powerful tool for simulating and analyzing the non-determinism, concurrency and different level of abstraction of any communication protocol. Future enhancements can be to further improve the performance of AODV routing protocol to obtain better result in terms of the packet delivery ratio, network lifetime and the average end to end delay with less routing overhead. The future work includes the validation of the protocol by using network simulator NS3.

REFERENCES

[1] Richa Agarwal, Ashish Jain. A Review Paper on Colored Petri Nets and their application in Protocol Verification. International Journal of Research in Computer and Communication Technology, Vol 2, Issue 8, August -2013.

[2] Alessandro Bianchi, Sebastiano Pizzutilo. A Coloured Nested Petri Nets Model for Discussing MANET Properties. International Journal of Multimedia Technology Jun. 2013, Vol. 3 Iss. 2, PP. 38-44.

[3] Jatinder Pal Singh, Anuj Kr. Gupta. A Review on Dynamic MANET On Demand Routing Protocol in MANETs. Volume 2, No.2, March - April 2013 International Journal of Advanced Trends in Computer Science and Engineering.

[4] Lakshmi P.S., Pasha Sajid and Ramana M.V. Security and Energy efficiency in Ad Hoc Networks. Research Journal of Computer and Information Technology Sciences Vol. 1(1), 14-17, February (2013).

[5] Agrahari S., and S. Chinara. Simulation of Random Waypoint Mobility Model Using Coloured Petri Nets. In ICCTS, 18th 19th August, New Delhi, 2012.

[6] Simaremare, Harris. Energy consumption analysis of modified AODV routing protocol under random waypoint and reference point group mobility models. Advanced Computer Science and Information Systems (ICACSIS), 2012 International Conference on IEEE, 2012.

[7] Hadjidj, Rachid, and Hanifa Boucheneb. Efficient reachability analysis for time Petri nets. Computers, IEEE Transactions on 60.8 (2011): 1085-1099.

[8] Suchismita Chinara, Santanu Kumar Rath. CPN validation of neighbor detection protocol for ad-hoc networks. In 8th International conference on information, communication and signal processing, 2011.

[9] Ali Nakhaee, Ali Harounabadi, Javad Mirabedini. A Novel Communication Model to Improve Mobile Ad hoc Network Routing Reliability. In Application of Information and Communication Technologies (AICT), 5th International Conference on 2011, Page(s): 1 - 7.

[10] Ali Nakhaee, Ali Harounabadi, javad Mirabedini. A Novel Communication Model to Improve AODV Protocol Routing Reliability. Application of Information and Communication Technologies (AICT), 2011 5th International Conference on 2011, Page(s): 1- 7

[11] Hanifa Boucheneb and Rachid Hadjidj. Model Checking of Time Petri Nets (2009).

[12] K. Jensen and L.M. Kristensen . Coloured Petri Nets - Modelling and Validation of Concurrent Systems. Springer, ISBN: 978-3-642-00283-0, 2009.

[13] Ruwini Kodikara, Sea Ling, and Arkady Zaslavsky. Evaluating Crosslayer Context Exchange in Mobile Ad-hoc Networks with Colored Petri Nets. In IEEE International Conference on 2007, Page(s): 173-176.

[14] C Siva Ram Murthy and B S Manoj. Ad Hoc Wireless Networks. In Pearson Education, 2005.

[15] Fazli Erbas, Kyandoghere Kyamakya, Klaus Jobmann . Modeling and Performance Analysis of a Novel Position based Reliable Unicast and Multicast Routing Method Using Coloured Petri Nets. In Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th, Volume:5.

[16] Bai R., Singhal M. Salvaging Route Reply for On-Demand Routing Protocols in Mobile Ad-Hoc Networks. In MSWIM 2005, Montreal, Quebec, Canada October 2005.

[17] H. Zhou. A Survey on Routing Protocols in MANETs. In Eas. Lansing, MI 48824-1027 Zhouhon1@msu.edu,Technical Report: MSU-CSE-03-08, Mar 28, 2003.

[18] C. Xiong, T. Murata, J. Tsai. Modeling and Simulation of Routing Protocols for Mobile Ad Hoc Networks Using Colored Petri Nets. Conferences in Research and Practice in Information Technology, Vol12, 2002.

[19] Lisa Wells. Performance Analysis using CPN Tools. Department of Computer Science, University of Aarhus Aabogade 34, 8200 Aarhus N, Denmark.

