International Journal of Electrical and Computer Engineering (IJECE)
Vol.2, No.3, June 2012, pp. 433~440
ISSN: 2088-8708

Modeling and Validation of the Dynamic Host Configuration Protocol with Colored Petri Nets

Farhad Soleimanian Gharehchopogh*, Reza Soufizadeh**
* Departementof Computer Engineering, Hacettepe University, Beytepe, Ankara, Turkey.
** Departement of Computer Engineering, Scince and Research Branch, IslamicAzad University, West Azerbaijan, Iran.

Article Info

Article History:

Received Apr 24th, 2012
Revised May 24th, 2012
Accepted May 29th, 2012

Keyword:

Petri Nets
Colored Petri Nets
DHCP
Modeling
ML

ABSTRACT

Petri Networks with a graphical language are based on mathematical logic which have many uses and have capability for modeling and validation of distributed systems and concurrent applications. Colored Petri Networks (CPNs) are a type of Petri Network models that are used in modeling of systems which contain discrete and scattered events. In general, CPNs are used to evaluate system performance and demonstrate the correctness of systems. Dynamic Host Configuration Protocol (DHCP) is one of the main systems of protocols special for servers that are used for dynamic allocation of IP to the network computers (clients). In this paper, we highlight to analyze the correctness and authenticity of DHCPs with the use of CPNs with using the CPN Tools and to prove the accuracy of our protocol's performance.

Corresponding Author:

Farhad Soleimanian Gharehchopogh,
Departement of Computer Engineering,
Hacettepe University,
06800, Beytepe, Ankara, Turkey.
Email: farhad@hacettepe.edu.tr

1. INTRODUCTION

Petri nets were introduced in the 1962 by Mr. Carl Adam Petri for modeling concurrent systems. But due to some weaknesses in the modeling of real systems, later in 1979 at Aarhus University in Denmark, Mr. Kurt Jensen introduced CPNs for the modeling of concurrent and distributed systems which had strong mathematical logic associated with ML programming language. In the CPN, sings have characteristics which they are called colors. Color of signs; actually indicate the belonging of signs to different groups and differentiation of their behaviors in the network. The meaning of color for signs resembles the meaning of chains or categories in lined networks. Any sign is a data object from a data type with a user-defined data structure which is defined in the ML functional language [1].

Nowadays the CPNsare used widely in the description of systems that are based on the mathematical logic. Specially, in recent years lots of researches had been done around the modeling of concurrent and distributed systems with CPNs. Also, for modeling and prove the correctness of it, and also performance of systems, different modeling languages have been introduced that each has its own strengths and weaknesses.

We can associate CPNs with time to examine performance of the system (to find the time consuming spots in system).One of the protocols existing on the Internet for dynamic allocation of IP to the clients is used by the server. This protocol's job is that, each computer in the network, needs an address to communicate with other computers, this address can be manually or automatically generated by a server named DHCP Server. In this paper, we model DHCP using CPNs' features and tools of CPNTools, and will show that this protocol works properly.

We will review performed previous works in this field in the section 2 of this paper. Section 3 and subsections of it are introduced the CPNs and modeling concepts with them. In the section 4 and subsection of it, we wil model the DHCP protocol by CPNTools and we will verified this protocol. Finally, in the section 5, we will compare the proposed modeling method with other methods and presented the results of the paper.

2. RELATED WORKS

CPN is as mathematical based modeling language for validation and verification of the concurrent, synchronization and distributed systems. As this regard, it has been many performed research in this domain. For example, Jensen et al. [2] are presented a brief introduction to the CPN modeling language and the associated analysis methods. They believe that the CPN models can be used to validate both the functional/logical correctness and the performance of the systems.

Phanse et al. [3] are considered an analytical model of the availability of the policy distribution service in a Mobile Ad hoc Networks (MANET) and analyze the performance by means of stochastic Petri nets (SPNs). The proposed model is composed of two levels of complexity which they are the simple Markovian model with Poisson assumptions and as a more accurate non- Markovian model with general distributions. Also, Jamali and Khosravi [4] are presented a new approach to ensuring correctness of an MANET routing protocols. They achieved a formal model for the protocols and analyze the models by manes of CPNs to determine if needed the protocol provides the defined service correctly. However, Othman et al. [5] applied the CPN to modeling and performance evaluation of the EQ-MAC protocol. EQ-MAC [6] is quality of service and energy efficient hybrid MAC protocol specifically used in wireless sensor networks.

Rauf and Jeyakumar[7] developed for various components of the experimental set-up and experimentation is conducted to measure the delay caused by the switch, the AP and the workstations based on CPN modeling language. As a result, throughput as network performance measure for WLAN decreases as the probability of collision increases. This measure decreased for a certain level, when the number of stations connected within the network increases. Also, reference [8] proposed a performance study of the distributed coordination function of 802.11 networks and proposed a detailed 802.11b model based on Object-oriented Petri Nets that precise back off procedure and time synchronization, performance analyses are evaluated by simulation for a dense wireless network. Azgomi and Khalili [9] proposed a new CPN model (hierarchical modeling capability of CPNs) for modeling and performance evaluation of a medium access control protocol in wireless sensor networks called sensor medium access control protocol (S-MAC). S-MAC is an energy-aware MAC protocol with nodes scheduling.

3. COLORED PETRI NETS

Places, transitions and symbols, meaning of colors, Guards and phrases have been introduced in the CPN for being capable of transferring data which calculated by the signs. In Petri network two phases of analysis and implementation take place simultaneously and also supports concurrency and are suitable for distributed systems and also are used for actual system implementation.

CPNs provide more accurate models of complex non-synchronous processing systems. In these networks, unlike the usual Petri networks tokens are recognizable from each other, as each of tokens has characteristics named color. In one side CPNs, unlike ordinary Petri networks are programmable, and the ML functional language is used to control and run the Petri net. CPNs unlike typical Petri network, have the ability to create hierarchical model, this technique can be used to obtain composition or reuse of models [10],[11].From advantages that CPN models have it can be mentioned the following:1)These techniques can be used to acquire composition or re-use of models like hierarchical models can be used. 2)For each transition the time needed for the fire can be defined and according to that each token can have the time as a feature, a suitable tool for evaluating performance and other related issues are related to time.3)CPNs have the capability of being combined with the ML programming language, so with the use of this feature much control can be imposed on the model [12].

3.1. CPN Difination

As mentioned, CPN is actually an extended version of typical Petri network. CPNs have strong mathematical logic this math logic (CPN = (P, T, C, I-, I +, M0)) is defined as follows:

[i] P, a non-empty finite set of places.

[ii] T, a non-empty set of transitions is finite.

[iii] $P \cap T = \emptyset$

[iv] C, is a color function mapping which is from the PUT, is a non-empty set.

[v] And I +, crossed forward and backward functions on P * T is defined in such a manner that:
$$I^{-}(p,t)\ I^{+}(p,t) : C(t) \Rightarrow C(p)\ \forall\ (p,t) \in C(p)$$
[vi] M0 is a function which is defined on p and describes the initial marking in the way that each p belongs to p this relationship **M0 (p) ∈ C (p)** is established [1].

In Figure1, one example of the CPN has been shown.

Figure 1. Shown CPN example

P1 is a place, and requestsection a pass, and words above passage is the guard that the logical value must be True in order to happen the shoot. In general, one passage is shot in a CPN if the following conditions are available:

[i] Each entrance place for transition, at least must have a sign.
[ii] For signals in entrance place, word needed must be attached to the input arc.
[iii] Guard (Boolean) for the transition to the desired value is numbered and the number of tokens in input places is consistent with the input arc.

In CPNs, the time can be given to the tokens and model, This time is defined by national or local clock which can be an integer variable (discrete) or real (constant). For fire the passage, the tokens of input places, should be smaller or equal to the current time. For time the Petri net model, it is enough to define the type of closets that are timed [1], [12].To model and analyze Petri networks, there are different tools with the simplicity and functionality used to point CPNTools [13]. After developing the model in this software, can run the model step by step or overall with the use of the software features and use the programming language ML [8], and produce Graph of state space in the form of output file for analysis.

3.2. Code in CPN

Code CPN, is a piece of code in ML language that are associated with a transfer and when the passage is fired, the mentioned code runs. The code is not only performing complex calculations, but consequently, the output variables of the input variables are obtained.

ML has a strategy to assess the amount of calls (call - by - value), static typing, type inference, algebraic data types, pattern matching and exception handling, first class functions, automatic memory management in dealing with unwanted data, parametric polymorphism and Dynamic types and type inference. More details about this are in [14], [15], and [16]. The reasons of using this language for CPN are: interactive (like Matlab), data type in this language isflexible, no hardware dependencies because it does not Crash, learning and development is very convenient, storage management by the programmer [1].

In each CPN, there is a concept called color sets, definition of the color set is performed in the Global declaration node. A color set in ML can be defined as follows: Color name = definition

Figure 2, an example which shows the use of the ML code in CPN.

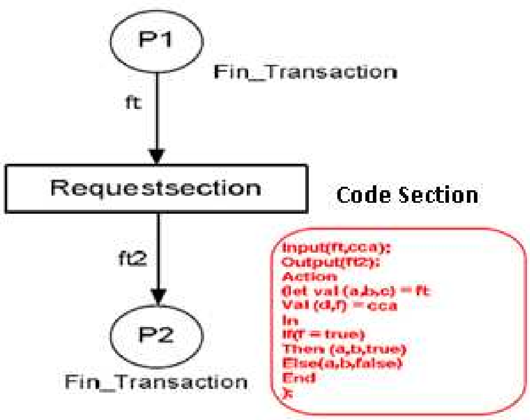


Figure 2. An example for shows the use of the ML code in CPN

4. MODELINGDHCP

To use this protocol, the computer which is chosen to get an IP address automatically, Broadcasts the DHCP Discover Packets in a Network. In this case DHCP Server sends an offer according to his table including the IP Address, Subnet and sometimes settings related to a DNS or Gateway settings, to the client computer, that this message is called DHCP Offer. After sending this message, DHCP Server waits for DHCP Request message confirming the settings from the client which finally DHCP Server with sending DHCP Server allows the client to use forwarded settings and IP. In this section, the way of modeling this protocol is given for the correctness of its function properly.

4.1. Receive IP Mechanism from DHCP Client by DHCP Server

DHCP clients use two different processes to communicate with DHCP Server and obtain Configuration. Initialization process occurs when a computer is turned on for the first time and tries to connect to a network. The renewal process occurs when a computer intends to have a new IP except his present IP after receiving an IP from the DHCP Server. IP allocation process is given in Table 1, respectively:

Table 1. DHCP process in the dynamic allocation of IP addresses to clients

NO	Description
1	DHCP client publish a DHCP Discover message to a Local Subnet (Broadcast)
2	A DHCP Server can reply by DHCP Offer Message, which contains an IP a proposal for allocation to the client.
3	Once DHCP Offer Message was received by the client, the client replies the server address offered by DHCP Request.
4	After the server receives the DHCP Request of the client, server sends a DHCPAcknowledgement named DHVP ACK for confirming the allocation of IP
5	When the client received DHCP Ack runs the settings that are already received to connect to the network [2,3]

4.2. Analizing DHCP Message

Message exchanged between client and server for IP allocation has a frame that has many fields but in this paper the fields which are used is given in Table 2 [17], [18]:

Table 2. Needed fields for defined variables in the language ML

FIELD	Description
Oper	Message Op Code, If Oper=1 then request else if Oper=2 then response
Xid	Transaction Id
Flag	Flags
Ciaddr	Client IP Address
Yiaddr	Your IP Address
Chaddr	Client Hardware Address
ReqIP	IPWhich Clientwant offer then Proposal from Server
Lease	IP ExpirationTtime
MsgT	Massage Name

At this stage, Color Set is defined in the language ML using the information above.

4.3. Color Sets Defination based DHCP with ML Language

```
colset UNIT = unit;
colset BOOL = bool;
colset NO = int;
colset DATA = string;
colset DATAxDATA = product DATA * DATA timed;
colset DATAxDATAxNO = product DATA * DATA * NO timed;
colset DATAxNO = product DATA * NO timed;
colset DHCPPacket = record oper:NO * xid:NO * flag:NO * ciaddr:DATA * yiaddr:DATA * chaddr:DATA * reqIP:DATA * lease:NO * msgT:DATA;
(* Mesage: op,xid,flags,ciaddr,yiaddr,chaddr, reqIP,lease,msgT,servID *)
```

4.4. Define Variables and Parameters Used in the Implementation of DHCP

Variable declaration :

```
var msg : DHCPPacket;
```

var n,xidn : NO;
var d, yiad, chaddr, sID : DATA;

Parameters Declaration :
val IPPool = Init [] "IP" NoOfIPs;
val Clients = Init [] "Client" NoOfClients;

4.5. DHCP Hierarchical Model with CPN

We create a hierarchical model with the use of CPN for the DHCP with Transitions Server, Client, and Network in CPNTools software as shown in Figure 3.

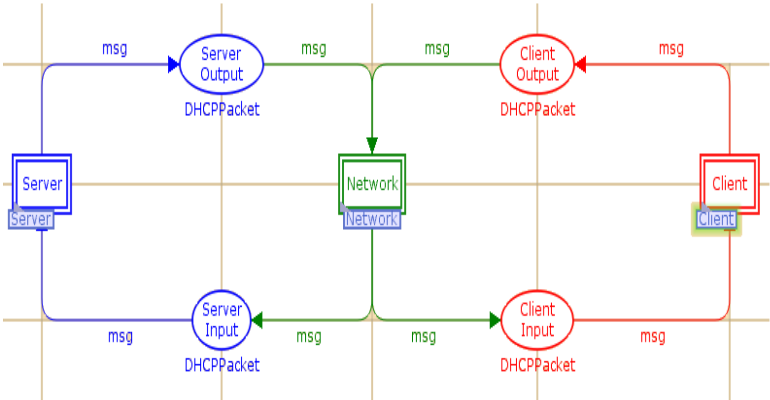


Figure 3. DHCP Hierarchical Model with CPN

4.6. Analysis of client sent messages on the server side and client

When the turned on client, plans to enter the network, it must request IP from the DHCP Server, so sends its MAC address or the name of its computer (Hardware Name) in the DHCP Server in a form of a message, Figure 4 shows this process.

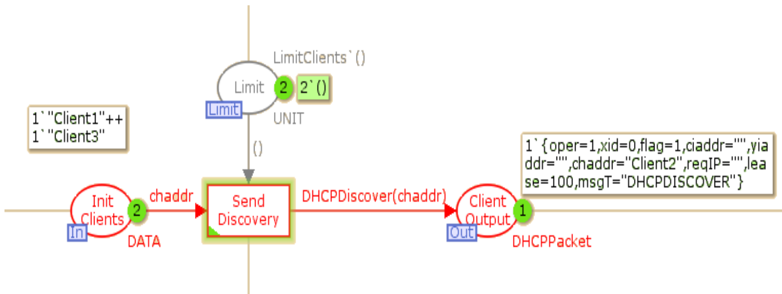


Figure 4. Sending Discover Model for Demonstrate the IP Request by the Client from the Server

Function DHCPDiscover: This function is defined by the needed arguments showed with Table 2 is as follows:

fun DHCPDiscover(cliHwAdd : DATA) = {oper=1, xid=0, flag=1, ciaddr="" , yiaddr="",chaddr=cliHwAdd, reqIP="", lease=defLeaseT, msgT="DHCPDISCOVER"};

(* Mesage: op,xid,flags,ciaddr,yiaddr,chaddr, reqIP,lease,msgT,sID *)

As shown in Figure 4, with firing SendDiscover and running DHCPDiscovery function, below are valued as follows:

1`{oper=1,xid=0,flag=1,ciaddr=””,yiaddr=””, chaddr=”Client2”, reqIP=””, lease=100, msgT= DHCPDISCOVER}

After client sent its message receiving IP to server, the server analyzes the message and will respond to it depending on the request. If DHCPDiscover is the client's message to the server, the server realizes that the client has a request for IP, so offers IP based on its allocation table. This is shown in Figure 5:

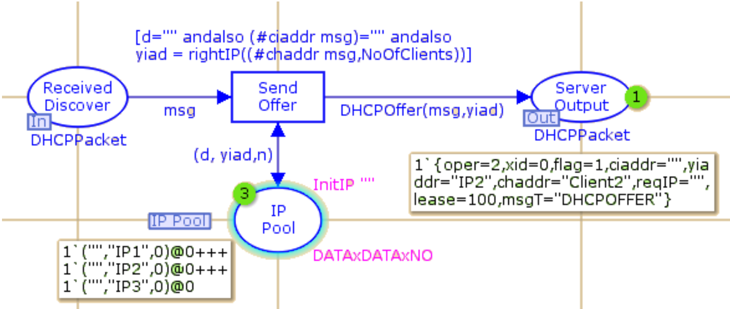


Figure 5. Processing discover model for demonstrate the ip request by the server to the client

DHCPOffer function with the required arguments, are defined as in Table 2 as follows:

fun DHCPOffer (msg : DHCPPacket, yiad : DATA) = {oper=2, xid=(#xid msg), ciaddr=””, yiaddr=yiad,flag = (#flag msg), chaddr=(#chaddr msg), reqIP=””, lease=(#lease msg), msgT=”DHCPOFFER”}

As indicated in Figure 5 with firing SendOffer and running DHCPOffer following values are initialized as follows:

1`{oper=2,xid=0, flag=1, ciaddr=””, yiaddr=”IP2”, chaddr=”Client2”,reqIP=””,lease=100,msgT=”DHCPOFFER”}

Client analyzes content (DHCPOFFER) using Field msgT and realizes that in yiaddr field placed amount of proposed IP. So it adopts the same IP and requests the server to give this IP to it. ProcessingOffer process runs so that by the DHCPRequest fraction, the values will change as follows:

1`{oper=1,xid=0,flag=1,ciaddr=””,yiaddr=”0”, chaddr=”Client2”,reqIP=”IP2”,lease=100, msgT=”DHCPREQUEST”}

DHCPRequest function with the required arguments, in accordance with Table 2 is defined as follows:

FunDHCPRequest (chadd: DATA, ciadd: DATA, req: DATA, xidn: NO) = {oper=1, xid=xidn, Ciaddr=ciadd, yiaddr="0", flag = 1, chaddr=chadd, reqIP= req, lease=defLeaseT,msgT="DHCPREQUEST"}

In this situation, server realizes that client wants that itself offered.

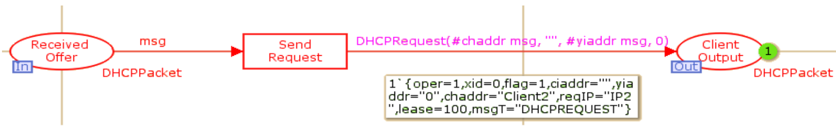


Figure 6. ProcessngOfferr model to demonstrate the application process

After the server receives a message, DHCPREQUEST, using IPPool, on the allocation table of IP, allocates the desired IP to the client and sends Ack message to the client's to approve and register chosen information on its table. If the server is not able to assign the desired IP, NACK message is sent to the client for the next steps. The order process is shown in Figures 7 and Figures 8.

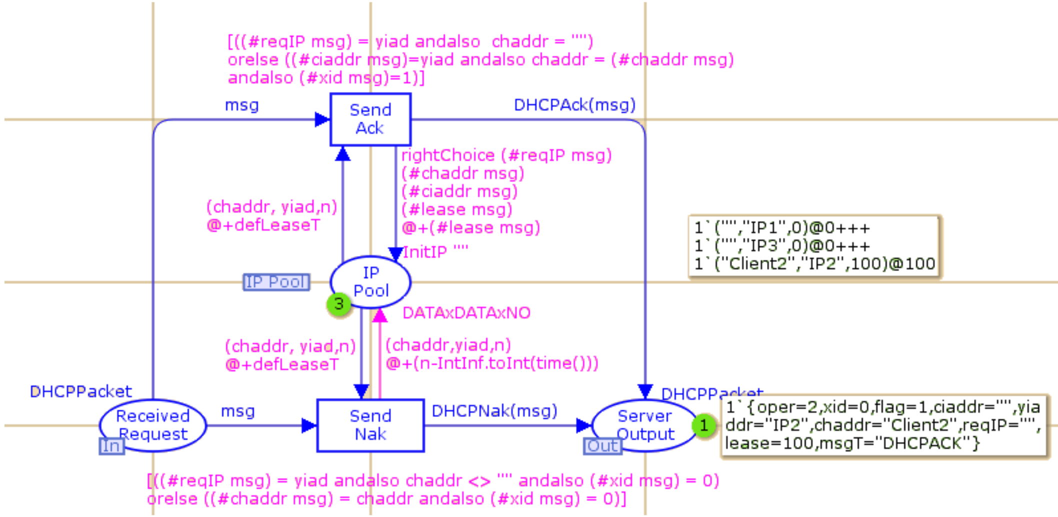


Figure 7. ProcessingRequest Model for Demonstrate the Ack and Nack process to the Client

DHCPAck function arguments required with Table 2 is defined as follows:

FunDHCPAck (msg:DHCPPacket) = {oper=2, ciaddr= (#ciaddrmsg),xid=(#xidmsg),yiaddr=(#reqIPmsg), flag = (#flag msg), chaddr=(#chaddrmsg), reqIP="", lease=(#lease msg), msgT="DHCPACK"}

After the firing the SendAck following values are initialized:

1`{oper=2,xid=0,flag=1,ciaddr="",yiaddr="IP2", chaddr= "Client2", reqIP="", lease=100, msgT="DHCPACK"}

DHCPNak function definition is as follows:

FunDHCPNak (msg:DHCPPacket) = {oper=2, ciaddr="", xid= (#xidmsg), yiaddr="0", flag = (#flag msg),chaddr= (#chaddrmsg), reqIP="", lease=0, msgT="DHCPNAK"}

This function is when is not received any answer from the server.

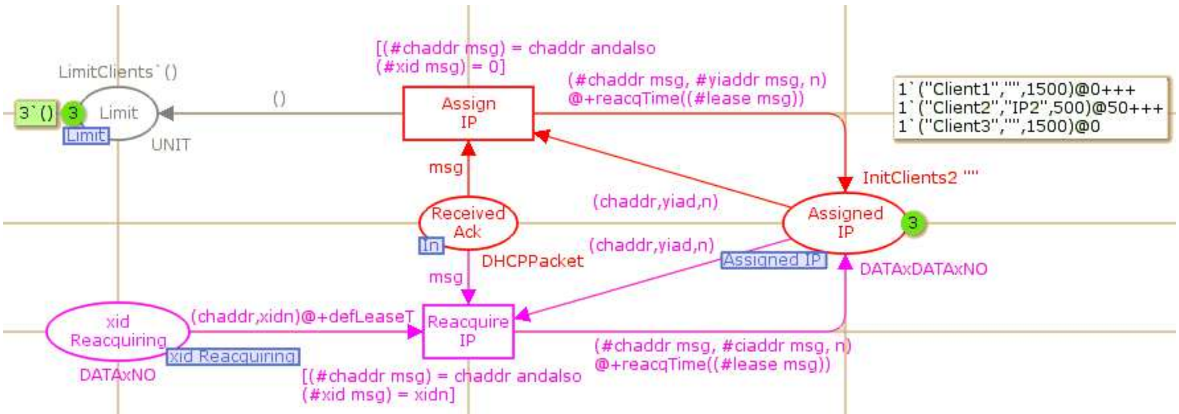


Figure 8. Model Processing Ack for the allocation of IP for a client

In figure 8 Reacquire IP transit is activated when the client wants to re-request IP. In this case the server will provide the same IP to the client; otherwise another IP will be assigned to the client.

5. CONCLUSION

CPNs are used to assess and demonstrate the efficiency and accuracy in concurrent and distributed systems. They have the ability to integrate with the ML programming language; this ability multiplies the power. In this paper the function of allocated protocol IP with dynamic namedDHCP was studied and the correctness of function of this algorithm with modeling of this protocol and with the CPNs was analyzed. Modelingresultsshow thatthisprotocolindynamic allocation ofIP addressestoclient function 100% correct.

REFERENCES

[1] K. Jensen and L. M. Kristensen, "Coloured Petri Nets, Modelling and Validation of Concurrent Systems", *Springer,* 2009, Ch. 10.

[2] K. Jensen, *et al.*, "Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems", *International Journal Software technology Transfer (2007),*vol. 9, 213-254, 2007.

[3] K.S. Phanse, A.Bhat, L.DaSilva," Modeling and Evaluation of a Policy Provisioning Architecture for Mobile Ad-Hoc Networks", *Journal of Network and Systems Management*, 1-18, 2006.

[4] M.A.J Jamali,T. Khosravi,"Validation of Ad hoc On-demand Multipath Distance Vector Using Colored Petri Nets", '2*011 International Conference on Computer and Software Modeling,* Singapore, vol. 14, pp. 29-34, 2011.

[5] J.B. Othman, S. Diagne, L. Mokdad, B. Yahya,"Performance Evaluation of a Hybrid MAC Protocol for Wireless Sensor Network", *ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWIM'10,* pp. 327-334, 2010.

[6] B. Yahya and J. Ben-Othman, "An Energy Efficient Hybrid Medium Access Control Scheme for Wireless Sensor Networks with Quality of Service Guarantees", *In the Proceedings of GLOBECOM'08,* New Orleans, LA, USA, pp. 1-5, 2008.

[7] H.A. Rauf, A.E. Jeyakumar, "Colored Petri Net Modeling and Throughput Analysis for Wireless Infrastructure Networks", *Ubiquitious Computing and Communication Journal,* vol. 3, no.3, pp. 155-160, 2008.

[8] A. Masri, *et al*., "Performance Analysis of IEEE 802.11b Wireless Networks with Object Oriented Petri Nets", *Electronic Notes in Theoretical Computer Science 242*, pp. 73–85, 2009.

[9] M.A. Azgomi and A. Khalili," Performance Evaluation of Sensor Medium Access Control Protocol Using Coloured Petri Nets", *Electronic Notes in Theoretical Computer Science 242 (2009)*, pp. 31–42, 2009

[10] Z.Esik, C.Martin-Vide, V. Mitrana, "Recent Advances in Formal Languages and Applications", *Springer-Verlag Berlin Heidelberg*, ch. 14, 2006.

[11] A. Bobbio, "System Modelling With Petri Nets", *A:G Colombo and A. Saiz de Bustamante, System Reliability Assessment*, *Kluwer p.c,* pp. 102-143, 1990.

[12] K. Jensen: "An Introduction to the Theoretical Aspects of Coloured Petri Nets",*In: J.W. de Bakker, W.-P. de Roever, G. Rozenberg (eds.): A Decade of Concurrency, Lecture Notes in Computer Science,* Springer-Verlag, vol. 803, pp. 230–272. 1994.

[13] Http://cpntools.org/ last available 20.05.2012.

[14] K. Jensen, "Coloured Petri Nets: A High-level Language for System Design and Analysis", *In: G. Rozenberg (ed.): Advances in Petri Nets 1990, Lecture Notes in Computer Science,* Springer-Verlag , vol. 483, 342– 416, 1991.

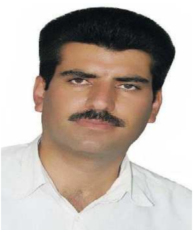
[15] R. Harper, "Programming in Standard ML (DRAFT: VERSION 1.2 OF 11.02.11.) *Robert Harper Carnegie Mellon University Spring Semester*, 2011.

[16] E.R. Ganser, "The Standard ML Basis Manual Edited", *Gansner AT&T Laboratories John H. Reppy University of Chicago, Cambrige*, 2004.

[17] L. McLean and J.C. Mackin, "MCSA/MCSE Self-Paced Training Kit (Exam 70-291): Implementing, Managing, and Maintaining a MicrosoftWindows Server 2003 Network Infrastructure", *Microsoft,* 2003.

[18] J. Davies, "Windows Serevr 2008 TCP/IP protocols and services", *Microsoft* , 2008.

BIOGRAPHY OF AUTHORS



FarhadSoleimanian Gharehchopoghreceived his B.Sc. degree in Computer Engineering from Islamic Azad University,Shabestar,Iran. He then received his M.Sc. degree in Computer Engineering in from Cukurova University, Adana, Turkey. Since, he has been working towardthe Ph.D degree in Computer Engineering at Hacettepe University in Ankara, Turkey. He works as a lecturer in the Department of Computer Engineering in Scince and Research Branch, Islamic Azad University, West Azerbaijan, Iran.Hisresearch interests include Machine Learning, Operating Systems, Software Cost Estimation, Natural Language Processing, and Bioinformatics. For more information please visit www.soleimanian.com.

Reza Soufizadehreceived his B.Sc. degree in Computer Engineering from Iran University of Science and Technology, Tehran, Iran. Scince, he has been working toward the M.Sc degree in Computer Enginering at Scince and Research Branch, Islamic Azad University, West Azerbaijan, Iran.

He is working on Microsoft SharePoint portal server & Microsoft SharePoint Services (2003) and MOSS 2007, WSS3 & Microsoft BizTalk server 2006 & Web part programming and Digital Signature (RSA & X509 Certificate) with C#.net & SQL Server, in VitaCo, RedRose, ShakhesAndishan, RahkareKhalagh companies.

For more information please visit www.reza4. net and content him by Soufizadeh@gmail.com