Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing

QoS Protocol Verification using Petri-Net for Seamless Mobility in A Ubiquitous Environment: A Case Study

Dalju Lee and Jongmoon Baik
Information and Communications University,
119 Munjiro, Yuseong-gu, Daejeon, 305-732 Korea
{djlee,jbaik}@icu.ac.kr

Abstract

Ubiquitous computing is a worthy environment for autonomic/pervasive computing between users and devices or between devices and devices. To realize ubiquitous computing, supporting seamless mobility ensures continuous services during inter-networking movement without meddling users. For seamless mobility in a heterogeneous networking environment, an adequate service/network configuration is necessary in accordance with the given environment. We developed the middleware, assisting to decrease human effort for application implementation and to guarantee interoperability for effective sustaining. A protocol of QoS (Quality of Service) NSLP (NSIS Signaling Layer Protocol) with advance resource reservation is modeled during the development process; furthermore, the protocol verification prior to be implemented ensures the correctness of the protocol and additionally reduces risks from ambiguous designs. In this paper, we present the case-study of QoS protocol verification utilizing Petri-Net as one of well-known methods; for seamless mobility, the verified QoS protocol can guarantee service quality and effective use of resources.

1. Introduction

Since Mark Weiser introduced a new paradigm called ubiquitous computing where information processing has been thoroughly integrated into everyday objects and activities [1], many researchers have been focusing on providing seamless services between users, devices, and networks. To realize ubiquitous computing, one of important issues is how to support seamless mobility in a ubiquitous environment. Seamless mobility is the ability to provide and to ensure continuous service during movements among heterogeneous devices or networks.

In order to achieve seamless mobility in a ubiquitous computing environment, the following four technologies must be considered: service reconfiguration, service discovery, QoS management, and context awareness technology. Through the service reconfiguration technology, suitable services between different networks can be reconfigured. Service discovery technology can be used to find the available resources and services when network changes. Context awareness technology focuses on management of context information such as network status, user's location, and device status. QoS management handles QoS policy, characteristics of accessible networks, and messages for installation and reservation of network resources.

We implemented a middleware based on these technologies to support seamless services between the two wireless protocols, Wibro (Wireless Broadband, IEEE 802.16e) and WiMAX (Worldwide Inter-operability for Microwave Access, IEEE 802.16a).

In this paper, we describe the verification of the protocol we designed using Petri-Net, which is suitable for communication protocol verification and for seamless service mobility among different networks. In general, if a protocol is verified before implementation, the initial risks such as increasing of cost, ambiguous design, and delayed schedules can be reduced.

The paper is organized as follow. Section 2 introduces verification methods and Section 3 describes the QoS protocol we designed. Section 4 describes the empirical protocol verification results using Petri-Net. Finally, we describe conclusions in Section 5.

2. Verification Methods for Protocol

In this section, we introduce several formal verification methods which are widely used in software industry. For communication protocol verification, a

978-0-7695-3263-9/08 $25.00 © 2008 IEEE
DOI 10.1109/SNPD.2008.124

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on May 12,2010 at 00:51:15 UTC from IEEE Xplore. Restrictions apply.

proper method must be selected so that properties and flows of protocol can be verified.

Formal method can be used to ensure the correctness of models in the early software life cycle. There are two classes of formal methods which are formal specification method and formal verification method.

Formal specification describes properties of a system in a formal notation based on mathematical theory; therefore, the consistency of a system can be assured mathematically. There are several well known formal specification languages such as Z [2], LOTOS [3], VDM [4], Larch [5], and Petri-Net [6]. A Formal verification method is classified into model checking and theorem proving [7]. Model checking transforms a logical model into automata and checks whether it has finite states. Theorem proving describes terminological form about the properties of requirements and proves the consistency of a system through mathematical inference rules.

For QoS protocol verification, we used Petri-Net, which is widely used for the verification of network protocols. Petri-Net describes a system using state transition approach and all possible states can be traced using reachability analysis. Petri-Net is very useful to analyze the properties of communication protocols such as control flows rather than timely analysis. However, state transition approach causes a problem, called state explosion as the number of states increases. The primary strength of Petri-Net provides the abilities of exploring the states and representing states graphically. The places, the elements of Petri-Net model, represent states of nodes in the network. The transitions symbolize messages of the protocols; moreover, the flows of tokens indicate variations with control messages or data among nodes related to network protocols.

Petri-Net [13] has the following properties for protocol verification: reachability, boundedness, liveness, reversibility, and consistency.

Reachability means that a certain sate can be reachable from the initial state. Boundedness is used to model limits on available system resources. The property of liveness is used to model that the system can never lock up a deadlock. Reversibility is used to represent re-initialization and cyclic behavior. Consistency means the existence of a cyclic behavior for some marking and consistency.

3. Protocol of Quality of Service

In our approach, we conducted verification process for the mobility-aware QoS NSLP (NSIS [Next Step in Signaling] Signaling Layer Protocol) with advance resource reservation [14]. The protocol is redesigned for solving the problem of the older QoS NSLP [15] that requires excessive passive reservation and modification in current Internet architecture. The protocol is supposed to be implemented for a middleware to support seamless services between the two wireless protocols, Wibro and WiMAX.

The QoS NSLP are accomplished via three major actions, resource advance reservation, crossover node (CRN) discovery, and localized state update. A CRN is discovered before mobile node's handoff and reserve initial network resources for a new path in advance. After a mobile node finishes a handoff, reserved resources in a new path become active. It is much more effective rather than session re-establishment in entire path in terms of signaling latency.

The QoS NSLP is designed with some assumptions: A mobile node receives L2 beacon signal from multiple access points in overlapped wireless cells and select the stringiest beacon signal to decide the future location of the mobile node, the movement prediction is only active in overlapped area to decrease the overhead of resource reservation.

Figure 1 shows the message exchange procedure of the QoS NSLP. QoS reservation is achieved by CRN discovery before MN (Mobile Node)'s handoff and the state of QoS is installed after MN's handoff.

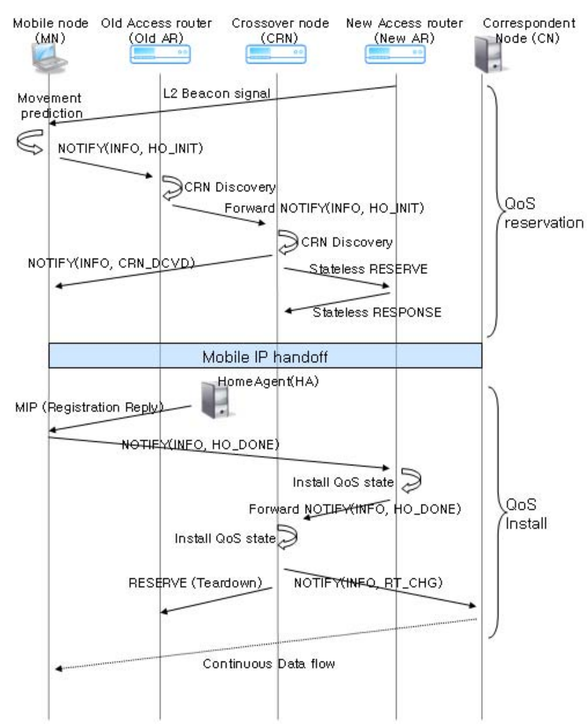

Figure 1. Message Exchange Procedure of QoS NSLP with Advance Resource Reservation

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on May 12,2010 at 00:51:15 UTC from IEEE Xplore. Restrictions apply.

4. Verification of QoS protocol

This section describes the verification of QoS protocol we performed. Firstly, state transition diagram for each node is described. Secondly, the verification using Petri-Net model is described.

4.1. State Transition Diagrams

As mentioned in the previous section, 4 nodes related to the QoS NSLP are identified and presented: mobile node (MN), old access router (old AR), crossover node (CRN), and new access router (new AR). We developed state transition diagram for each node to verify the protocol

Figure 2 illustrates the state transition diagram of MN; more exactly, MN has 4 states: normal, movement prediction, mobile handoff preparation, and handoff done. Normal state is the initial state and may receive multiple L2 beacon signal from multiple access points (AP) in overlapped wireless cells. Normal state transit into movement prediction state when a signal is received.

Movement prediction state is a state where the future location of MN is determined. It keeps the current state when MN sends NOTIYF (INFO, HO_INIT) message, which contained INFO and HO_INFO: INFO is indication of message; HO_INFO is represented handoff initiation, for CRN discovery to upstream signaling path. However, movement prediction state transit into mobile handoff preparation state if NOTIFY (INFO, CRN_DCVD) message, which means that crossover node for handling MN's handoff is discovered, is received.

Mobile handoff is achieved using the Mobile IP [17][18]; hence, its verification is excluded.

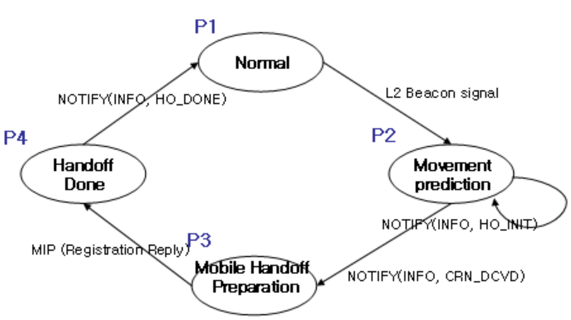

Figure 2. State Transition Diagram of MN

When MN receives MIP (Registration Reply) message (the mobile handoff finish) from home agents (HA), the state of MN transit into handoff done state.

As a consequence, handoff done sate transits into normal state after MN sends NOTIFY (INFO, HO_DONE) message (the handoff finish) to new AR.

The state transition diagram for old AR is shown in Figure 3. Old AR's states are defined by normal state and CRN discovery state. Normal state is the initial state and it transits into CRN discovery state when NOTIFY (INFO, HO_INIT) is received from MN. CRN discovery state is a certain sate for deciding a crossover node for MN's seamless handoff. If the current node is not CRN then old AR send forward NOFITY (INFO, HO_INIT) message and the current state is kept.

CRN discovery state transits into normal state when RESERVE (Teardown) message, which contained Teardown flog of NSLP and means to release reserved resources, is received

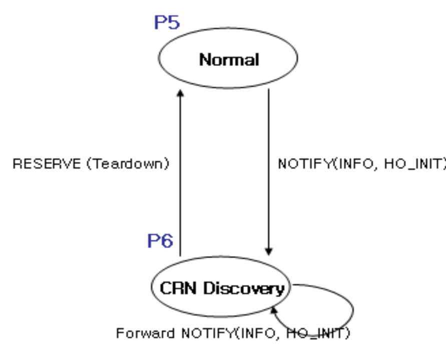

Figure 3. State Transition Diagram of old AR

Figure 4 illustrates the state transition diagram of CRN. CRN has 4 different states: normal, CRN discovery, reservation preparation, and QoS installation. Normal state is the initial state and it transits into CRN discovery state when forward NOTIFY (INFO, HO_INIT) from old AR is received. In CRN discovery state, if CRN is confirmed then CRN send stateless RESERVE message to new AR and it send NOFITY (INFO, CRN_DCVD) message to MN, however, the current state is kept.

CRN discovery state transits into reservation preparation when stateless RESPONSE message, which means that routing information is saved without installing QoS NSLP state, from new AR is received. After finished mobile IP handoff among MN, FA (Foreign Agent), and HA (Home Agent); reservation preparation state transits into QoS installation state by receiving forward NOTIFY (INFO, HO_DONE) from new AR.

In QoS installation state, reserved QoS NSLP state is activated. The current state is kept when CRN sends NOTIFY (INFO, RT_CHG) message, which contained RT_CHG: RT_CHG represent that routing table

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on May 12,2010 at 00:51:15 UTC from IEEE Xplore. Restrictions apply.

change, to CN (Correspondent Node). After CRN send RESERVE (Teardown) message to old AR, CRN's state transit into normal states.

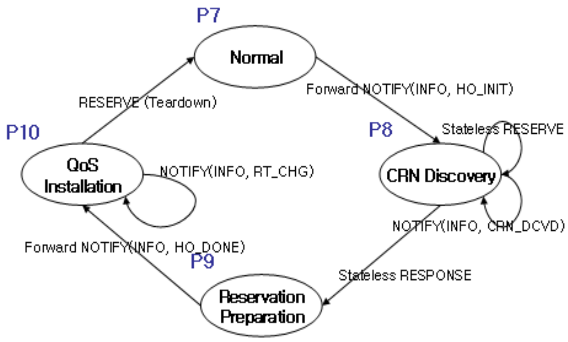

Figure 4. State Transition Diagram of CRN

Figure 5 illustrates the transition diagram of new AR. New AR has 3 states: normal, reservation preparation, and QoS installation.

Normal state is the initial state and it transits into reservation preparation state when stateless RESERVE message is received. Reservation preparation state is a ready state where QoS NSLP state is not active until mobile IP handover is finished.

New AR's state transits into QoS installation state when NOTIFY (INFO, HO_DONE) message is received. QoS installation state transits into normal state when new AR send forward NOTIFY (INFO, HO_DONE) to CRN.

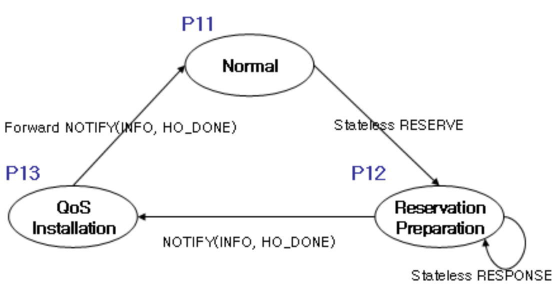

Figure 5. State Transition Diagram of new AR

4.2. Verification using Petri-Net model

In this section, we describe the model verification for the QoS NSLP protocol using Petri-Net based on the state transition diagrams we developed.

Figure 6 shows the Petri-Net model which represents message and transition among MN, old AR, CRN, and new AR nodes. The states of the state transition diagram are represented as places (P1, P2, …, P13) in the Petri-net. Messages of the state transition diagrams are represented as transitions (T1, T2, …, T11) in the Petri-net model.

P1, P5, P7 and P11 places are the initial places for the different nodes; the black points in P1, P5, P7 and P11 places are the initial marking; each place transit into next place through a transition.

Table 1 summarizes each place of Petri-Net model and Table 2 is also summarizes each transition and corresponding message.

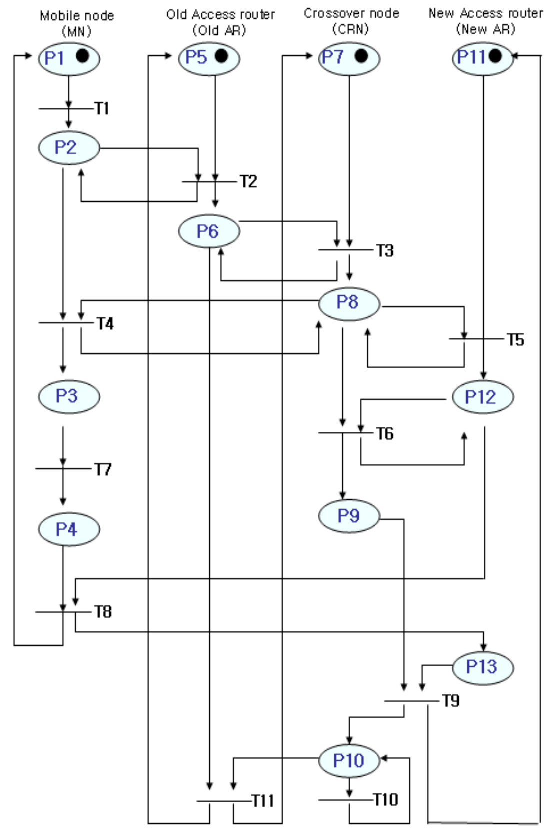

Figure 6. Petri-Net Model of QoS protocol

Table 1. Place of Petri-Net model

Place	Place followed in transition
P1, P5, P7, P11	Normal
P2	Movement prediction
P3	Mobile handoff preparation
P4	Handoff done
P6, P8	CRN discovery
P9, P12	Reservation preparation
P10, P13	QoS installation

Table 2. Transition of Petri-Net model

Transition	Corresponding message
T1	L2 beacon signal
T2	NOTIFY(INFO, HO_INIT)
T3	Forward NOTIFY(INFO, HO_INIT)

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on May 12,2010 at 00:51:15 UTC from IEEE Xplore. Restrictions apply.

T4	NOTIFY(INFO, CRN_DCVD)
T5	Stateless RESERVE
T6	Stateless RESPONSE
T7	Registration Replay
T8	NOTIFY(INFO, HO_DONE)
T9	Forward NOTIFY(INFO, HO_DONE)
T10	NOTIFY(INFO, RT_CHG)
T11	RESERVE(Teardown)

Figure 7 shows the reachability tree for the validation of compatibility such as checking protocol deadlock and liveness. M_0 is the initial place having token (black point) and M_0 transit into M_1 by t1 transition. We were able to confirm that the initial place always come back without a deadlock through reachability tree (M_0-> M_1->…->M_o); all places are passable to transit from the initial place. Marking count is always limited under 1; therefore we could identify the stability of the designed protocol.

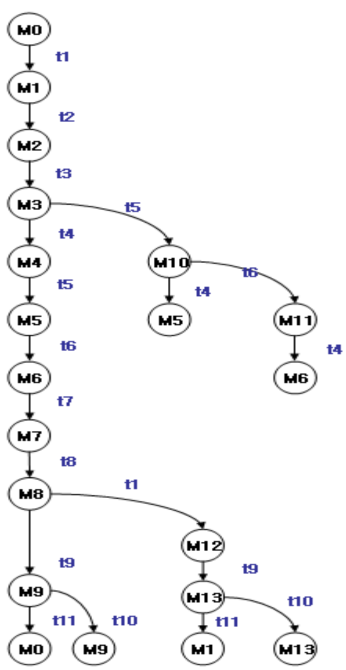

Figure 7. Reachability tree

From the reachability tree, all possible place sets are identified. Table3 summarizes the place set of the reachability tree. The header represents places and the left-most column represents the place set having marking. The reachability analysis was very useful to validate state transitions of the places and the correctness of the Petri-net model using simulation tools.

Table 3. Place set of the reachability tree

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13
M0	1				1		1				1		
M1		1			1		1				1		
M2		1				1	1				1		
M3		1				1		1			1		
M4			1			1		1			1		
M5			1			1		1				1	
M6			1			1			1			1	
M7				1		1			1			1	
M8	1					1			1				1
M9	1					1				1	1		
M10		1				1		1				1	
M11		1				1			1			1	
M12		1				1			1				1
M13		1				1				1	1		

Finally, we represent the Petri-Net model using the Volker Guth's Simulator as shown in Figure 8 [16]. Through the simulation, we could check the all status of the designed protocol according to the possible transitions and validate that the result of the simulation and the place set of the reachability tree are consistent. Thereby, it is proved that the QoS NSLP is correct and stable.

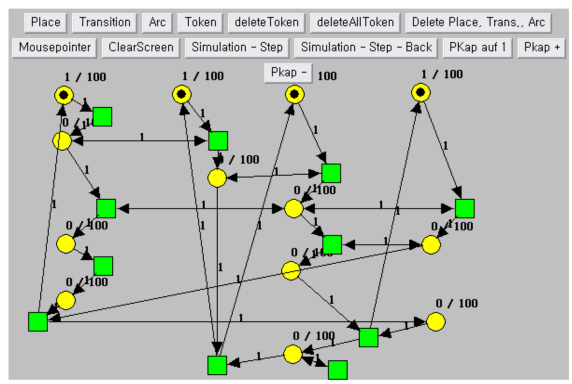

Figure 8. Petri-Net simulation using Volker Guth's Simulator

5. Conclusion

In this paper, we introduced QoS NSLP with advance resource reservation for seamless mobility in a ubiquitous environment. We also introduced the case study of protocol verification using Petri-net we performed. States and interaction messages of each node are represented as a state transition diagram. The correctness and consistency of the designed protocol is verified through Petri-Net modeling and the reachability tree, and the results of the reachability tree are checked by simulation. This paper shows procedure of protocol verification, therefore, it is useful to apply

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on May 12,2010 at 00:51:15 UTC from IEEE Xplore. Restrictions apply.

the verification method using Petri-Net in other domains.

Protocol verification before the implementation of a system can provide the benefits such as cost-effectiveness and project risk reduction by design changes. Also logical behaviors of a protocol can be guaranteed.

6. References

[1] M. Weiser, “The Computer for the 21st Century”, ACM SIGMOBILE Mobile Computing and Communications Review, Vol.3, No.3, pp 3-11, 1999.

[2] J. M. Spivey, “The Z Notation: A Reference Manual”, Prentice-Hall, International Series in Computer Science, 1992.

[3] E. Cusack, S. Rudkin and C. Smith, “An Object-Oriented Interpretation of LOTOS”, in S. Vuong (ed), Vol.2, pp 211-226, North Holland, 1990.

[4] A. Laorakpong and M. Saeki, “Object-Oriented Formal Specification Development using VDM”, Proceedings. of ISOTAS '93, Lecture Notes in Computer Science, Vol.742, pp 529-543, Springer-Verlag, 1993.

[5] J. V. Guttag and J. J. Horning, “Larch: Languages and Tools for Formal Specification”, Springer-Verlag, 1993.

[6] R. Zurawski and M. Zhou, “Petri Nets and Industrial Applications : A Tutorial”, IEEE Transactions on Industrial Electronics, Vol.41, No.6, 1994.

[7] E. Clarke and J. Wing, “Formal Methods: State of the Art and Future Directions”, ACM Computing Surveys, Vol.28, No.4, pp.626-643, 1996.

[8] Ki-won Chong, Yong-sun Cho, Sung-goo Kwon, “Detecting Error and Checking Consistency in the Object-Oriented Design Models”, KIPS transactions, Vol.6, No.8, 1999.

[9] A. Tsiolakis, “Consistency Analysis of UML Class and Sequence Diagrams based on Attributed Typed Graphs and their Transformation”, Technical Report 2000/3, Technical University of Berlin, 2000.

[10] T. Sunetnanta and A. Finkelsteing, “Automated Consistency Checking for Multiperspective Software Specifications”, ICSE2001, 2001.

[11] M. Richters and M. Gogolla, “Validating UML models and OCL Constraints”, Proc. of UML 2000 - The Unified Modeling Language. Advancing the Standard Third International Conference, pp 265-277, 2000.

[12] M. Richters, “A Precise Approach to Validating UML Models and OCL Constraints”, PhD thesis, University Bremen, Logos Verlag, Berlin, BISS Monographs, No.14, 2002.

[13] M. Zhou and M. D. Jeng, ”Modeling, Analysis, Simulation, Scheduling, and Control of Semiconductor Manufacturing systems: A Petri Net Approach”, IEEE transaction, Vol.11, No.3, August 1998.

[14] Sooyong Lee, Myungchul Kim, Kyounghee Lee, Soonuk Seol, and Gabsoo Lee, “Seamless QoS Guarantees in Mobile Internet using NSIS with Advance Resource Reservation”, AINA2008, 2008.

[15] B. Benmammar and F. Krief, “MQoS NSLP: a mobility profile management based approach for advance resource reservation in a mobilie environment,” in The 7th IFIP IEEE International Conference on Mobile and Wireless Communications Networks (MWCN), pp 19-21, 2005.

[16] Guth’s Petri Net Java Applet, http://www.informatik.uni -hamburg.de/TGI/PetriNets/tools/java/Guth.

[17] RFC3775: Mobility Support in IPv6, http://tools.ietf.org /html/rfc3775.

[18] RFC3344: IP Mobility Support for IPv4, http://tools.ietf .org/html/rfc3344.

Authorized licensed use limited to: Korea Advanced Institute of Science and Technology. Downloaded on May 12,2010 at 00:51:15 UTC from IEEE Xplore. Restrictions apply.