PEPA Analysis of MAP Effects in Hierarchical Mobile IPv6

Hao Wang and Dave Laurenson
Institute for Digital Communications,
Joint Research Institute for Signal & Image Processing,
School of Engineering & Electronics,
University of Edinburgh
Email: {H.Wang, Dave.Laurenson}@ed.ac.uk

Jane Hillston
Laboratory for Foundations of Computer Science,
School of Informatics,
University of Edinburgh
Email: Jane.Hillston@ed.ac.uk

***Abstract*—To overcome the drawbacks of the Mobile IPv6 protocol on handling local mobility management, IETF proposed the HMIPv6 protocol which introduces an intermediate mobility anchor point (MAP) to hide the movement of a mobile node within a local area. However, the MAP forms a bottleneck in the network since all the traffic destined for its served nodes has to go through it. Most research on HMIPv6 focuses on protocol optimisation, and performance analysis of HMIPv6 is usually simulation-based. In this paper, we employ a performance evaluation formalism named PEPA to investigate the performance tradeoffs of MAPs in HMIPv6. Performance measures such as response time and MAP utilisation are presented.**

I. Introduction

To provide continuous connectivity when mobile users change their points of attachment to the Internet, the IETF proposed mobility management protocols Mobile IPv4 [1] and Mobile IPv6 [2] to support global mobility in IP-based networks. In Mobile IPv6-aware networks, a mobile node is always addressable at its home address regardless of its location. Whenever a mobile node moves into a new access network, it acquires one or more care-of addresses representing its current network attachment. The mobile node needs to send Binding Update messages (BUs) which associate its home address with its current care-of address to the mobile node's home agent (HA) and all the correspondent nodes (CNs) it is communicating with. The movement of the mobile node can then be made transparent to the transport and higher-layer by mapping home address to care-of address at the network layer. However, although the Mobile IPv6 protocol supports a route optimisation communication mode, the quality of service will decrease if the mobile node changes its point of attachment so frequently that handoff latency and signalling load caused by Binding Update messages become significant.

To overcome this drawback of the global mobility management protocols, IETF proposed local mobility management protocols such as Cellular IP [3] and Hierarchical Mobile IPv6 (HMIPv6) [4]. The HMIPv6 minimises the amount of signalling outside a local domain by using a new mobility agent, called a Mobility Anchor Point (MAP), that can hide the movement of the mobile node within a local domain. However, the MAP has to operate as a relay node between the mobile node and the CNs since by design all the traffic must go through the MAP. Under heavy traffic conditions, this local mobility management results in the MAPs becoming the bottlenecks of the network and thus network performance is degraded.

In this paper we use a performance evaluation formalism named PEPA to investigate the effects of MAPs on the response time and MAP utilisation in HMIPv6 with a client-server architecture. In particular we investigate the number and placement of MAP nodes within an access network. The rest of paper is organised as follows. In Section II we introduce the PEPA formalism. The HMIPv6 protocol is reviewed in Section III. We present our PEPA model of HMIPv6 and derive performance measures in Sections IV and V respectively. Section VI presents our conclusion.

II. PEPA

Performance Evaluation Process Algebra (PEPA) [5] is both a timed and stochastic extension of classical process algebra such as CCS [6] and CPS [7]. In PEPA a system is described as a component or a group of components that engage in activities. Generally, components model the physical or logical elements of a system and activities characterise the behaviour of these components. Each activity a in PEPA is defined as a pair (α, r) — action type α and activity rate r. The action type can be regarded as the name of the activity and the rate specifies the duration of the activity which is an exponentially distributed random variable. If a component P behaves as Q after completing activity a, then we can denote this transition as:

$$P \xrightarrow{\alpha} Q \ or \ P \xrightarrow{(\alpha,r)} Q$$

The PEPA formalism provides a small set of operators which are able to express the individual activities of components as well as the interactions between them. We only present the operators we used in our model in this section. For more details about PEPA operators, see [5].

Prefix: $(\alpha, r).P$

The component $(\alpha, r).P$ carries out an activity that is of action type α and has a delay that is exponentially distributed with rate r, which gives an average delay of $1/r$. After

completing this activity, the component $(\alpha, r).P$ behaves as component P.

Choice: $P + Q$

The component $P + Q$ may either behave as P or Q. All the enabled activities in P and Q are also enabled in this component and compete with each other. The first activity to be completed will be an activity of P or Q and this will distinguish which component wins the race. When the first activity is completed, all the other activities will be abandoned.

Cooperation: $P \underset{L}{\bowtie} Q$

The component $P \underset{L}{\bowtie} Q$ models the interaction between P and Q. The letter L denotes a set of action types that must be carried out by P and Q together. For all activities whose action type is included in L, P and Q must cooperate to complete it. However, P and Q can carry out other activities independently.

Parallel: $P \| Q$

The component $P \| Q$ represents two concurrent but completely independent components. It is shorthand notation for $P \underset{\emptyset}{\bowtie} Q$.

Constant: $P \stackrel{def}{=} Q$

This expression is used to assign names to components. Such expressions may be mutually recursive leading to infinite behaviours over finite states.

Since the duration of the transition in PEPA is exponentially distributed, it has been shown that the stochastic process underlying a PEPA model is a discrete state space, continuous time Markov chain (CTMC). By deriving the steady state probability distribution for the Markov chain, together with the Markov reward models [8], we can achieve performance measures such as utilisation and throughput. Moreover, measures such as response time can also be calculated by transient analysis. These measures can facilitate model verification and system optimisation.

III. Hierarchical Mobile IPv6 Overview

In Hierarchical Mobile IPv6, there is a mobility agent called a Mobility Anchor Point (MAP) that covers a group of access routers (ARs). Each of these ARs represents a different IP access network and a MAP forms a local network domain. Every time a mobile node moves into a MAP domain, it acquires an on-link care-of address (LCoA) referring to the AR to which it is connected and a regional care-of address (RCoA) referring to the MAP domain. Outside the MAP domain, the mobile node is identified by its RCoA and all the packets addressed at RCoA are intercepted by the MAP and forwarded to the mobile node at LCoA. When the mobile node performs a localised handoff, i.e. switches to a different AR within a MAP domain, it just needs to send a local BU to the MAP to change the mapping between its LCoA and RCoA. The mobile node only needs to send BUs to its HA and CNs informing them of its new RCoA when it moves to a new MAP domain. Therefore, the MAP is able to hide the movement of a mobile node within a local network domain, thereby minimising the handoff latency and outbound signalling load.

However, this mobility management scheme requires all the traffic between the mobile nodes and the CNs to go through the MAPs, which can result in them being bottlenecks in the network and thus degrade network utilisation. Most research on HMIPv6 focuses on performance metrics associated with mobility such as signalling cost and handoff latency, etc., and the analysis is usually simulation-based. In this work, we investigate some other valuable metrics related to behaviour between handoffs i.e. the response time of ARs and the utilisation of the MAP.

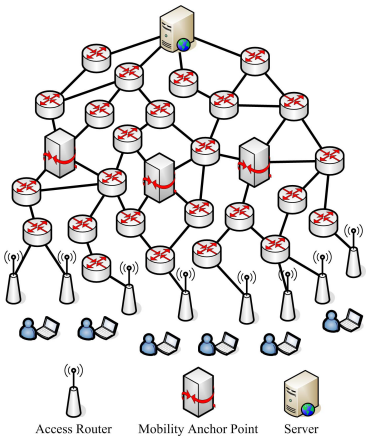

Fig. 1. A Typical Network Architecture of Hierarchical Mobile IPv6

When dealing with networked systems, researchers will typically use simulation tools such as *ns-2* [9] to evaluate performance. In order to avoid problems caused by topologies exhibiting uncharacteristic behaviours, a large set of topologies need to be evaluated, and each one multiple times with different instantiations of traffic flows. Whilst this may be possible for small sized networks, to evaluate large networks, such as those supporting HMIPv6, simulation is not a reliable means of determining performance characteristics. Instead, an analytical approach that is mathematically tractable, but captures the essential characteristics of the network is required.

In this work we present a CTMC-based performance model which is constructed using PEPA to evaluate this local mobility management mechanism. Although there are other Markovian-based performance modelling techniques, such as queueing networks and stochastic Petri nets, PEPA is chosen because its component structure has a better reflection of the system structure, thereby providing a clear description of the system it models. Moreover, since PEPA models can be solved numerically, some restrictions which other modelling approaches must follow to exhibit a product form solution do not constrain PEPA models. In the following sections, this PEPA model of HMIPv6 is described and the effects of the MAPs are analysed.

IV. PEPA Model of HMIPv6

A typical network topology that might be encountered by Hierarchical Mobile IPv6 devices is shown in Fig. 1. The significant elements in the network are the access routers, which act as access points for the mobile devices, mobility anchor points to implement the local mobility function, and the server which serves data destined for the mobile user.

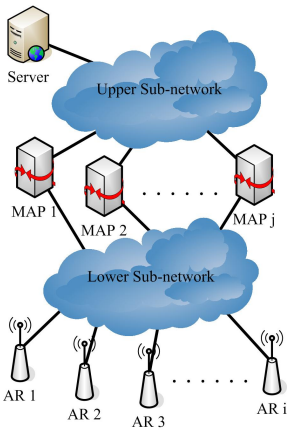

Fig. 2. An Abstract Network Architecture of Hierarchical Mobile IPv6

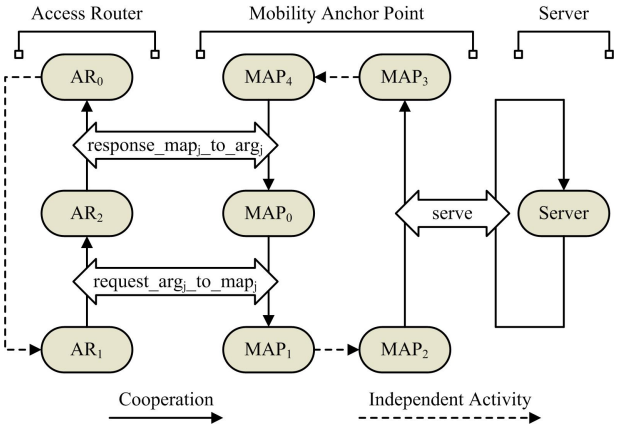

Fig. 3. Cooperations between AR, MAP and Server

The other entities in the figure act as routers for the traffic between these key elements. This network architecture can be considered as the common client-server architecture, where requests and responses have to go through the intermediate MAPs. With any Markov chain representation of interacting entities there is a disproportionate increase in the number of states as the number of entities increases. In order to produce a model that is analytically tractable, and to ensure that the results are more generic than those for a specific topology, the network has been abstracted into the one in Fig. 2. The upper and lower sub-networks model the effect of the entities between the server and MAPs, and those between the MAPs and access routers, are simply represented by delays within the model. We should point out that since we assume the sub-networks are not congested and regard the transmission of requests and responses in the networks as delays, it is not necessary to import components that model the upper and lower sub-networks in the model. This level of simplification is sufficient to derive performance measures that are discussed later in this paper. Moreover, the proposed model is not intended to model the exact HMIPv6 protocol. Instead, it is intended to reflect the interactions between the elementary components of the protocol. Below we show the PEPA definitions for the three types of components.

Access Router (AR): The traffic scenario used in our model is mobile users asking for access of web pages stored on the server through the ARs. An AR covers a number of mobile users and receives requests from them. To investigate the effects of the MAP on one mobile user and simplify the model, we model the ARs as the sources of requests instead of the mobile users and the AR generates the requests at the rate of λ_i. The AR then sends out the requests and waits for the responses. For the ARs that are within the domain of MAP_j, they are considered as an access router group ARG_j. The transmission delays between ARG_j and MAP_j are implemented in cooperations $(request\_arg_j\_to\_map_j, \top)$ and $(response\_map_j\_to\_arg_j, \top)$, corresponding to sending requests and receiving responses (The symbol $\top$ is used in PEPA to denote the passive activity whose activity rate is determined by its cooperation partner). Therefore, the ARs within the same MAP domain have identical behaviour. The component AR is defined as:

$$\begin{aligned} AR_{i0} &\stackrel{def}{=} (ar_i\_request, \lambda_i).AR_{i1}; \\ AR_{i1} &\stackrel{def}{=} (request\_arg_j\_to\_map_j, \top).AR_{i2}; \\ AR_{i2} &\stackrel{def}{=} (response\_map_j\_to\_arg_j, \top).AR_{i0}; \end{aligned}$$

Mobility Anchor Point (MAP): Each MAP handles a certain number of ARs and relays all the traffic between the ARs and the server. The MAP receives requests from the ARs and forwards them to the server. Once the requests are answered at the server, i.e., completing the $(serve, \top)$ cooperation, the MAP collects and sends the responses back to the ARs. The delay of the traffic in both directions is divided into two parts, i.e., the delays in the lower sub-network and upper sub-network. The length of delays in lower and upper sub-networks is determined by the parameters α_{j1} and α_{j2} respectively. Unlike the delay in the lower sub-network, the delay in the higher sub-network is not implemented as a cooperation between the MAPs and the Server since doing this will prohibit the Server from serving other MAPs. It should be pointed out that since we consider the MAPs as the bottlenecks of the network, a MAP is modelled as a scarce resource and is not able to accept requests from other ARs while it is already engaged by one AR. The definition of MAP is given below:

$$\begin{aligned} MAP_{j0} &\stackrel{def}{=} (request\_arg_j\_to\_map_j, \alpha_{j1}).MAP_{j1}; \\ MAP_{j1} &\stackrel{def}{=} (request\_map_j\_to\_server, \alpha_{j2}).MAP_{j2}; \\ MAP_{j2} &\stackrel{def}{=} (serve, \top).MAP_{j3}; \\ MAP_{j3} &\stackrel{def}{=} (response\_server\_to\_map_j, \alpha_{j2}).MAP_{j4}; \\ MAP_{j4} &\stackrel{def}{=} (response\_map_j\_to\_arg_j, \alpha_{j1}).MAP_{j0}; \end{aligned}$$

Server: We assume the server has infinite buffer size and is able to manage as many requests as the MAPs can submit, i.e., it carries out the serve activity in an iterative way. Since the server can cooperate with whichever MAP at a time, it is regarded as a server with random order service strategy. The component Server is defined as:

$$Server \stackrel{def}{=} (serve, \mu).Server;$$

TABLE I
PARAMETERS VALUES

Type (Role)	Average Time (ms)	Rate (1/ms)
ar_i_*request* (request rate)	10	0.1
request_arg$_j$*_to_map*$_j$ (delay of request in lower sub-network)	5	0.2
request_map$_j$*_to_server* (delay of request in upper sub-network)	5	0.2
response_server_to_map$_j$ (delay of response in upper sub-network)	5	0.2
response_map$_j$*_to_arg*$_j$ (delay of response in lower sub-network)	5	0.2
serve (service rate)	0.01	100

TABLE II
MAPPING BETWEEN ACCESS ROUTERS AND MAPS IN SCENARIOS I –V

Scenario	MAP1	MAP2	MAP3
I	1,2,3,4,5,6	N.A.	N.A.
II	1,2,3	4,5,6	N.A.
III	1,2,3,4	5,6	N.A.
IV	1,2	3,4	5,6
V	1,2,3	4	5,6

System Definition: This definition specifies how the system is constructed from the defined components. Generally, a PEPA system is defined as the interactions between the components. For our model if there are m ARs and n MAPs then the system is expressed as:

$$System \stackrel{def}{=} (AR_1 \| \cdots \| AR_m) \underset{L_1}{\bowtie} (MAP_1 \| \cdots \| MAP_n) \underset{L_2}{\bowtie} (Server)$$

where

$$L_1 = \{request\_arg_j\_to\_map_j, response\_map_j\_to\_arg_j\},$$

$$L_2 = \{serve\}.$$

The cooperations between the defined components are represented diagrammatically in Fig. 3.

Parameters Setting: To derive performance measures we first need to set the activity rates within the model. There are three types of activity in the model, i.e. the request rate of the AR, the delay in the sub-networks and the service rate. In our experiments, we assume an AR receives 10^2 web page requests per second from the mobile users and the server is able to handle 10^5 requests every second. For the delay of the sub-networks, we assume there are two or three hops for every message transmission, and based on the default value used in *ns-2*, the delay of each hop lies between 1 and 2ms. Therefore we set the average delay of the sub-networks to be 5ms. These activity rates are shown in Table I. Moreover, we assume the MAPs sit in the middle of the network and both requests and responses take the same routes, hence the activity rates for delay in the sub-networks are set to be the same.

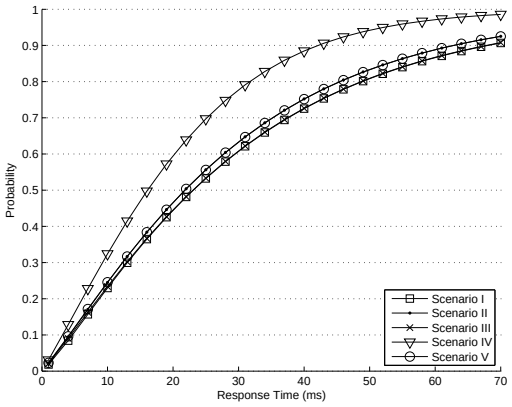

Fig. 4. CDF of the Response Time of AR1 in Different Scenarios

V. PERFORMANCE EVALUATION

Unlike most of the performance analysis of HMIPv6 that focus on handoff delay and signalling cost, the performance measures we investigate are the response time of the ARs and the utilisation of the MAPs. We carry out our experiments using the PEPA Workbench [10] and associated tools such as ipc [11] and Hydra [12]. More details on these tools can be found at http://www.dcs.ed.ac.uk/pepa. We analyse five network scenarios with 6 ARs and different numbers of MAPs. The connectivity of the ARs and the MAPs are shown in Table II.

A. Response Time

Response time is the time between an AR sending a request to the AR receiving the response. The response time for an AR in our model comprises three time periods, namely: queueing time at the MAP; delay in the sub-networks; and waiting time at the server.

We first investigate the response time, in the form of a cumulative distribution function (CDF), of AR1 in all of the five network scenarios. The result is shown in Fig. 4. Given the request rates of 0.1 for all ARs, the response time of AR1 degrades and reaches the limit rapidly as the MAP domain size increases. As we can see from the figure, forcing MAP1 to serve 4 ARs is as bad as connecting 6 ARs to it, and about 10% of the requests cannot be answered within 70ms. In the scenarios where the MAP1 serves three ARs, since the MAPs behave independently and do not block the server, AR1 has the same response time distribution, which is shorter but the improvement is not significant. However, making AR1 compete with only AR2 for MAP1 provides us a much faster response from the server. This suggests that the response time of an AR strongly relies on the domain size of its MAP and more MAPs does not necessarily mean a faster response.

The relationship between the response time and request rate is another performance metric we are interested in. We examine the response time of AR1 against the request rate of AR2 in scenario IV, where AR1 has the shortest response time.

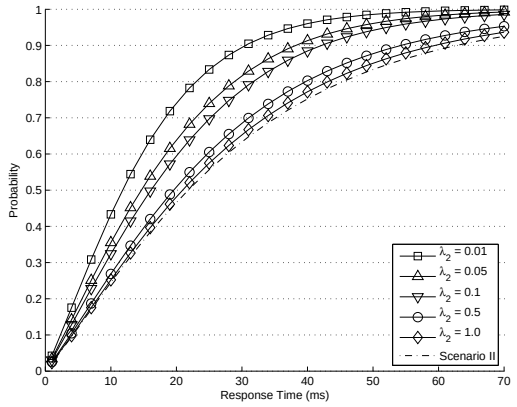

Fig. 5. CDF of the Response Time of AR1 in Scenario IV with Increasing Request Rate at AR2

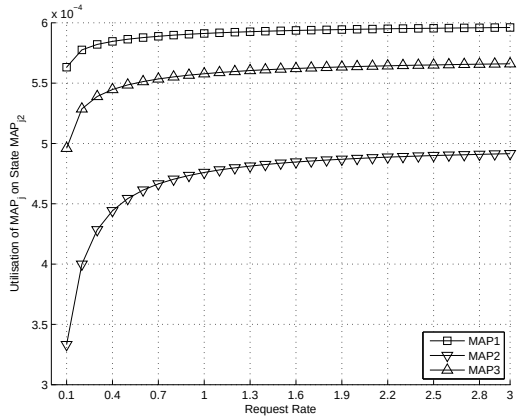

Fig. 6. Utilisation of MAP_j on State MAP_{j2} with Increasing Request Rate at All ARs in Scenario V

The result is shown in Fig. 5. It is clear to see that AR1 has to wait a longer time if the request rate of AR2 increases. This is because as AR2 sends requests more frequently, MAP1 is more likely to be engaged by AR2, which prevents AR1 from using MAP1. Comparing Fig. 4 and Fig. 5, it can be found that as we increase the request rate of AR2, the response time of AR1 is getting close to that in scenario II where three ARs are attached to MAP1. This means that the heavily loaded AR2 is so greedy that it starves AR1 of MAP1 as if another AR were introduced. On the other hand, AR1 could receive faster response if its competitor AR2 is relatively lightly loaded with request rate much smaller than 0.1. Therefore, it can be concluded that the response time of an access router also relies on the workload of its MAP and a heavily loaded AR is best connected to a MAP with other ARs which are lightly loaded.

B. MAP Utilisation

Another important performance measure is the utilisation of each MAP. This metric can tell us whether a MAP is well or underutilised. In [5] the utilisation is defined as the fraction of the time in which a component stays in different states.

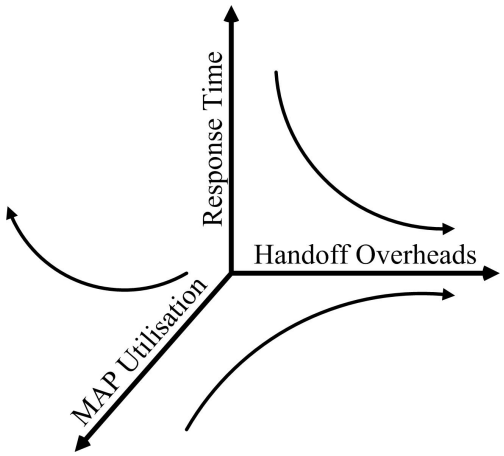

Fig. 7. Trade-off Between Response Time, MAP Utilisation and Handoff Overheads

Although in practice a MAP does not have states in which it can engage in the activities corresponding to the delay in the network, those states can represent the idle phases of a MAP. To investigate the utilisation of a MAP, we can analyse the proportion of time that each MAP spends on the serve activity, i.e., being in the MAP_{j2} state, which can indicate the workload of that MAP.

We carry out the experiments using scenario V and increasing request rates of all ARs from 0.1 to 3.0 and keeping the other activity rates the same as set in Table I. The results are shown in Fig. 6.

As the request rates increase, the MAPs can engage in the service activity more frequently. However, the speed of increase is different in each MAP. This can be explained as follows: In our model, the mean sojourn time for MAP_j in state MAP_{j0} is comprised of two periods, namely, waiting for a request and delay of a request. Since the mean waiting time (for a request) of a MAP is inversely proportional to the request rate and the delay in the lower sub-network is not affected by the request rate, the mean sojourn time in state MAP_{j0} does not change in a linear way, which results in a non-linear increase in sojourn time in all the other states, including MAP_{j2}. From Fig. 6, we can find that even if the request rate of AR4 rises to 3.0, the utilisation of MAP2 is very close to that of MAP3 where both of its ARs have requests rate of 0.1. Also, MAP1 can have almost the same utilisation as MAP2 with very busy ARs while its ARs are only lightly loaded. This means that connecting a heavy load to a MAP does not necessarily mean a better utilisation of that MAP, but a larger MAP domain size can improve its utilisation.

VI. Conclusion

In this paper we have presented the performance evaluation formalism PEPA and demonstrated its application to the HMIPv6 protocol with a common client-server network architecture. Since a MAP is a bottleneck in the network, performance modelling can play an essential role in assessing its effects on the network. We investigate the impacts of the

MAP on the response time of AR and the utilisation of MAP. The response time is a very important metric because it is a measure of the network QoS that is experienced by the user. A large response time usually results in unsatisfactory service or even service interruption. The results indicate that the MAP domain size is an important performance factor to the response time of ARs. A smaller MAP domain size provides a better response time. However, reducing the MAP domain size implies uneconomical network deployment with more MAP nodes and more inter MAP domain handoff, which minimise the expected benefits of HMIPv6. Moreover, the MAP domain size also affects the utilisation of a MAP, and larger domain size implies a better use of that MAP. This kind of trade-off is shown in Fig 7. As we increase the MAP domain, we can achieve better MAP utilisation and smaller handoff overheads, at the expense of larger response time.

Furthermore, the results also show that a heavily loaded AR can starve the other ARs sharing the same MAP. An intuitive solution of this problem would be connecting the heavily loaded AR to a MAP with light workload. However, this requirement cannot be easily fulfilled in mobile communication scenarios where the heavily loaded ARs are continually changing. This issue is important if we want to integrate the mobility management and QoS mechanisms. In the mobile network (NEMO) scenario [13], where mobile nodes move as a group, this can easily reduce the QoS of their visited AR. This situation should be improved if ARs can choose their MAPs adaptively according to their requested load. In our future work, we will design a more sophisticated PEPA model of HMIPv6 that can express different types of data traffic and investigate possible mechanisms of integrating mobility and QoS management.

ACKNOWLEDGEMENT

The work reported in this paper has formed part of the Ubiquitous Services Core Research Programme of the Virtual Centre of Excellence in Mobile & Personal Communications, Mobile VCE, www.mobilevce.com. This research has been funded by the DTI-led Technology Programme and by the Industrial Companies who are Members of Mobile VCE. Fully detailed technical reports on this research are available to Industrial Members of Mobile VCE. J. Hillston is also supported by EPSRC Advanced Research Fellowship EP/c543696/01 and EU FET-IST Global Computing 2 project SENSORIA (Software Engineering for Service-Oriented Overlay Computers (IST-3-016004-IP-09)). H. Wang and D. Laurenson acknowledge the support of the Scottish Funding Council for the Joint Research Institute with the Heriot-Watt University which is a part of the Edinburgh Research Partnership.

REFERENCES

[1] C. Perkins, "IP Mobility Support," RFC 2002, Oct. 1999.
[2] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," RFC 3775, Jun. 2004.
[3] A. T. Campbell, J. Gomez, and A. G. Valko, "An overview of cellular ip," in *Proc. Wireless Communications and Networking Conference '99*, Sep. 1999, pp. 606–610.
[4] H. Soliman, C. Castelluccia, K. E. Malki, and L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)," RFC 4140, Aug. 2005.
[5] J. Hillston, *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.
[6] R. Milner, *Communication and Concurrency*. Prentice Hall, 1989.
[7] C. A. R. Hoare, *Communicating Sequential Processes*. Prentice Hall, 1985.
[8] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. John Wiley & Sons, 1998.
[9] The Network Simulator - *ns*-2. [Online]. Available: http://www.isi.edu/nsnam/ns/
[10] S. Gilmore and J. Hillston, "The PEPA workbench: A tool to support a process algebra-based approach to performance modelling," in *Proc. Modelling Techniques and Tools for Computer Performance Evaluation '94*, May 1994, pp. 353–368.
[11] J. Bradley, N. Dingle, S. Gilmore, and W. Knottenbelt, "Derivation of passage-time densities in PEPA models using IPC: The Imperial PEPA Compiler," in *Proc. Modeling, Analysis and Simulation of Computer and Telecommunications Systems '03*, Oct. 2003, pp. 344–351.
[12] J. Bradley, N. Dingle, S. Gilmore, and W. Knottenbelt, "Extracting passage times from PEPA models with the HYDRA tool: A case study," in *Proc. UK Performance Engineering Workshop '03*, Jul. 2003, pp. 79–90.
[13] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," RFC 3963, Jan. 2005.