INTERNATIONAL JOURNAL OF NETWORK MANAGEMENT
Int. J. Network Mgmt 2009; **19**: 1–24
Published online 17 January 2008 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/nem.684

QoS management in mobile IP networks using a terminal assistant

Badr Benmammar[1,*,†], Zeina Jrad[2] and Francine Krief[1]

[1]*LaBRI Laboratory, University of Bordeaux I, Talence, France*
[2]*LIPN Laboratory, University of Paris XIII, Villetaneuse, France*

SUMMARY

This paper describes a signalling environment for Quality of Service (QoS) negotiation and advance resource reservation in mobile IP networks. This environment is built in conformance with the generic signalling environment, which is standardized by the NSIS IETF working group. The advanced resource reservation protocol, called MQoS NSLP, is based on the QoS NSLP signalling application. It provides to mobile terminals the QoS required based on the user's mobility and QoS profile. In this work, we investigate the use of some techniques of the AI (Artificial Intelligence) domain to implement a user interface called NIA (Negotiation Individual Assistant) in order to determine the QoS profile and negotiate the QoS parameters in the new domain after the handover. Therefore, we use connectionist learning in the management of the negotiation profiles and agent technology to help the user choose the best service provider, dynamically negotiate the QoS on the user's behalf, and follow the user's behaviour to be able to anticipate the negotiation and manage renegotiation. The advance resource reservation is based on an object MSpec (Mobility Specification) which determines the future location of the mobile terminal. The MSpec object is a part of the mobility and QoS profile and is determined by the NIA in the mobile terminal.

1. INTRODUCTION

In recent years, the appearance and development of real-time applications as well as multimedia applications have witnessed an exponential increase. The real-time constraints of these applications present a big challenge for their integration. That is why we need services adapted to specific application needs with a guaranteed Quality of Service (QoS) [1]. The new-generation Internet has to provide these services, particularly for real-time applications.

However, the implementation of QoS mechanisms is a very heavy task. It is difficult to manually configure all the network devices because of the abundance of QoS information and because of the dynamic nature of QoS configurations. The operator must control the attribution of network resources according to applications and users' characteristics. Using management tools adapted to QoS quickly proves essential. In order to simplify the router's configuration by permitting its automation, the IETF proposed a general framework called policy-based networking [2] for the control and management of these IP networks.

In such an environment, most applications cannot dynamically express their QoS requirements to obtain the adapted level of service. For each application, the customer and the provider have to agree on rules of assignment of service levels. They sign a contract called SLA (Service-Level Agreement) which is then translated into high-level policies. These policies are not directly executable by the network devices. They must be translated into intermediate and then into low-level policies which are understandable by network devices. The SLS (Service-Level Specification) is the technical version of the SLA [3] and the QoS parameters (also called performance metrics) are a part of the SLS parameters.

*Correspondence to: Badr Benmammar, LaBRI Laboratory, University of Bordeaux I, Talence 33400, France
†E-mail: benmamma@labri.fr

Received 13 February 2007
Revised 1 September 2007
Accepted 29 October 2007

There are some protocols [36–39] that allow dynamic negotiation of the required level of service and the needed quality between the user and the network entities. However, this negotiation seems to be complex because the user has to indicate himself the technical parameters that reflect the required QoS and associate values with the technical parameters according to his needs. The difficulty of the process can be reduced by replacing the user in finding the application needs in terms of QoS parameters according to the context of use. We therefore propose to help the user to express his needs in terms of QoS, to choose the service provider that best answers his demand and then to dynamically (re)negotiate the technical QoS parameters. The first objective of our work is to propose an intelligent user interface called NIA (Negotiation Individual Assistant) in order to determine a negotiation profile and manage the negotiation of QoS. In this work, we investigate the use of some techniques of the Artificial Intelligence (AI) domain in order to implement this assistant. We use connectionist learning in the management of the user's profiles and agent technology to help the user in the negotiation of SLS parameters according to the determined profile.

On the other hand, in 2002 the IETF launched the Next Steps in Signalling working group (NSIS). The initial objective of this group was to unify all the existing solutions of IP signalling or to make them coexist. With the emergence of IP networks and the increasing number of applications requiring a high level of QoS, the signalling problem became increasingly critical. Providing universal signalling which takes into account QoS as well as security and mobility is a very difficult task. Initially, the NSIS working group aimed the QoS, and proposed the QoS NSLP [4] signalling application.

The second objective of our work is to propose a signalling environment for QoS negotiation and advance resource reservation in mobile IP networks in conformance with the generic signalling environment standardized by the NSIS IETF working group. The QoS negotiation and resource reservation is based on the profile determined by the NIA.

This paper is organized as follows. Section 2 presents a synthesis of the research relating to resources reservation in an IP mobile environment. Section 3 defines the NSIS environment in which we specify the advance resource reservation protocol. Section 4 presents our user's mobility and QoS profile. We then present the user assistant (NIA) for QoS negotiation in Section 5. Finally, Section 6 is a description of a use case of the signalling environment.

2. QoS IN MOBILE IP NETWORKS

Recent research takes an interest in advance resource reservation to provide the necessary QoS to mobile terminals. In mobile integrated services networks, the majority of research is interested in extending the RSVP protocol. User mobility prediction also represents a key factor for providing QoS over mobile networks.

Talukdar and Badrinath [5] proposed a new protocol of resource reservation in a mobile environment called MRSVP (Mobile RSVP). In this model of reservation, the mobile terminal can make advance reservations in a set of cells named MSPEC (Mobility Specification). The MSPEC is not very clear; it only indicates the future locations of the mobile terminal but the MSPEC is not described. The authors proposed new RSVP messages in order to manage the user's mobility. This technique requires additional classes of service, major changes to RSVP and a lot of signalling.

The same authors [6,7] described an architecture which supports *mobility independent* and *dependent services* in the same network. In this architecture, the concept of *active* and *passive* reservation is used to obtain a better use of resources. The reservation for a flow in a link is called active if the packets of this flow pass through this link in order to reach the receiver. The reservation is called passive if the resources are reserved for this flow on the link, but the current packets for this flow are not transmitted on this link. The resources of the passive reservation can be used by other flows which do not require a QoS guarantee, such as Best Effort flows.

Min-Sun Kim *et al.* [8] proposed a resource reservation protocol in a mobile environment. The proposed protocol introduces the *RSVP agent* concept in order to guarantee the necessary QoS through an

DOI: 10.1002/nem

anticipation of the resource reservation. In this protocol there are three classes of resource reservation to obtain a better use of resources:

- *The Free class*. This represents the resources used by the best effort flows.
- *The Reserved class*. This represents the reserved resources for a specific flow that are currently used.
- *The Prepared class*. This represents the reserved resources for a specific flow that are not currently used.

Levine *et al.* [9] described the shadow cluster concept, aimed at improving resource allocation and call admission procedures in wireless networks. Shadow clusters are used to allocate resources that need to be reserved for the handover call. They also determine whether a new call should be admitted into a wireless network based on the call's requirements and local traffic conditions. The shadow cluster concept is targeted for ATM-based wireless networks with a micro/nanocellular architecture, where the service will be provided to users with very diverse requirements. With this concept, the QoS of mobile calls can be improved by reducing the number of dropped calls during the handover, and by disallowing the establishment of new calls that are highly likely to result later in a dropped call.

Le Grand *et al.* [10] proposed a new protocol called MIR (Mobile IP Reservation Protocol), which provides QoS guarantees to mobile terminals. This protocol is an adaptation of the CLEP (Control Load Ethernet Protocol) described in Horlait and Bouyer [11], which supports mobility. A distributed algorithm allowing each cell to operate separately and the fact that the use of RTS/CTS is not needed at the IEEE 802.11 level are the main advantages of this proposal. Two parameters are used in order to improve the necessary QoS:

- *speed of terminals* (fast or slow), measured in number of handovers during a certain period of time.
- *type of connection* (degradable or not).

For the simulation, the authors modelled the behaviour of the cells using Markov chains.

Ferrari *et al.* [12] described a distributed mechanism in order to make reservations in advance for real-time connections. In this mechanism, the reservation demand is classified according to two types: *immediate* and *in advance*:

- An immediate reservation is activated at the moment of the demand; its length is not specified.
- A reservation in advance is associated with two parameters: *starting time* (the time of the reservation activation) and *duration* (the reservation period).

Another way to obtain a better use of resources is to determine the future locations of the mobile terminal.

Samaan and Karmouch [13] present an architecture named Mobility Prediction Agent (MPA) that accurately performs mobility prediction using knowledge of the user's preferences, goals and spatial information without imposing any assumptions about the availability of his movement's history. Using the concept of evidential reasoning of the Dempster–Shafer theory, the MPA captures the uncertainty of the user's navigation behaviour by gathering pieces of evidence concerning different groups of candidate future locations. These groups are then refined to predict the user's future location when evidence accumulates using Dempster's rule of combination.

Akyildiz and Wang [14] proposed a User Mobility Profile (UMP), which is a combination of historic records and predictive patterns of mobile terminals, to serve as fundamental information for mobility management and enhancement of QoS in wireless multimedia networks. The authors developed the UMP framework for estimating service patterns and tracking mobile users, including descriptions of location, mobility and service requirements. For each mobile user, the service requirement is estimated using a mean-square error method. The authors proposed a new mobility model which is designed to characterize not only stochastic behaviours but also historical records and predictive future locations of mobile users. The authors used an adaptive algorithm which is designed to predict the future positions of mobile terminals in terms of location probabilities based on moving directions and residence time in a cell.

DOI: 10.1002/nem

Aljadhai and Znati [15] proposed a framework for call admission control and QoS support in wireless environments. In order to provide the necessary QoS to the mobile user, the authors take care of the current mobility context, such as past movement history and current direction of movement. The prediction technique uses the direction prediction method in geometric space to derive directional probabilities of cells being visited by a mobile unit. This essentially depends on the tacit assumption of relative positions and structure of cells in the wireless network.

3. SIGNALLING ENVIRONMENT

The IETF decided in 2002 to launch the NSIS working group to try to unify all solutions of signalling or to make them coexist. This group standardized two-layer architecture: the NSLP layer to generate signalling flows for different purposes and the NTLP layer to transport those flows in a path-coupled way. Some concepts used in this architecture are inspired from RSVP but a modification and simplification were made to support generic signalling.

3.1 GIST

GIST (General Internet Signalling Transport) is the protocol that NSIS has adopted as a standard for the NTLP layer [16]. GIST is conceived for an in-band transport of signalling flows generated by the NSLP layer, i.e. signalling flows follow the same path as the data flows. Besides, it only treats unicast signalling. Finally, GIST collaborates with the underlying transport and security layers to assure good routing of signalling flows. Two functions are assured by GIST:

- routing: to determine the next adjacent GIST node on the data path;
- transport: GIST uses the datagram mode with UDP to discover GIST nodes on the data path thanks to a specific Router Alert Option (RAO), whereas the connection mode is used with TCP or SCTP to transport NSLP flows.

3.2 QoS NSLP

Whereas the NTLP layer has the transport of signalling as an essential goal, the NSLP layer assures the generation of this signalling in accordance with user needs. QoS NSLP [4] is the first NSLP layer protocol to be elaborated in NSIS: it permits the generation of signalling to provide a certain level of QoS by making reservations on the data path independently of the QoS models (Diffserv, Intserv, etc.) adopted by the different domains. With NTLP, QoS NSLP spreads functionalities of RSVP, such as the creation, refreshing, modification and elimination of a reservation state. On the other hand, QoS NSLP proposes an interaction with the Resource Management Function (RMF) for access control in accordance with specified control policies.

NSLP leans on NTLP so that signalling generated by its applications is transported correctly toward target nodes, which is why QoS NSLP, an example of NSLP protocol, interacts with the lower-layer NTLP in order to achieve these objectives. NTLP is independent of the NSLP layer signalling application and it is through the intermediary of one API that parameters asked by one layer are obtained.

QoS NSLP generates four messages types:

- *Reserve*. The only message, which handles the reservation state (refresh, create, remove).
- *Response*. Using this message, a response is sent to a message received.
- *Query*. This message is used to request information concerning the nodes which are on the data path, for example the available resources.
- *Notify*. Using this message, it is possible to inform a node without preliminary request.

DOI: 10.1002/nem

All QoS NSLP messages contain a common header, followed by objects whose use is given according to the type of each message. Among the objects we find Response Request, Refresh Period, Session Id, Error Spec and QSpec.

We are interested particularly in the QSpec object, because this object is used to specify the desired QoS. The following parameters are proposed for the QSpec [17]: QSpec ID, QSM control information, QoS description—traffic descriptors, QoS class, QoS characterization, excess treatment, priority and reliability, service schedule, monitoring requirements.

3.3 MQoS NSLP

We name MQoS NSLP, the procedure of resources reservation, in advance using QoS NSLP messages in a mobile environment. This reservation procedure is applied in an HMIPv6 architecture. The Mobility Anchor Point (MAP) plays a significant role in reserving the resources in advance on behalf of the mobile terminal.

QoS NSLP operates according to the two following modes: *sender-initiated reservation* and *receiver-initiated reservation.* In the first mode, the sender of the flow initiates the reservation (generates the RESERVE message). In the second mode, the reservation is initiated by the receiver of the flow.

The Mobile Host (MH) can be a sender or a receiver of the flow, so there are four possible scenarios:

- The MH is the receiver of the flow with the mode *sender-initiated reservation.*
- The MH is the receiver of the flow with the mode *receiver-initiated reservation.*
- The MH is the sender of the flow with the mode *sender-initiated reservation.*
- The MH is the sender of the flow with the mode *receiver-initiated reservation.*

The advance resource reservation is based on an object Mobility Specification (MSpec) which determines the future location of the mobile terminal. This object is defined in Section 4. An example of advance resource reservation by using MQoS NSLP is given in Section 6.

4. USER ASSISTANT FOR QoS NEGOTIATION

The user assistant proposed in this paper is placed on the user terminal and is called the Negotiation Individual Assistant (NIA). The NIA negotiates the QoS between the user and the service provider from one side, and between the user and the network from the other side (Figure 1). The main purpose of the assistant is the representation of the user in requesting and negotiating the desired QoS in a dynamic environment. This representation is illustrated by the following points:

- analysis of the user's work for a profile attribution;
- saving and updating of all data concerning the preferences;
- translation of the user's requests into SLS parameters;
- negotiation of the desired QoS with a service provider;
- monitoring of the real-time quality to compare it with the negotiated one;
- substitution of the user in decision-making.

As shown in Figure 1, the proposed assistant contains different layers. The first one is the profile management layer. This is the layer of direct contact with the user. Once connected, the principal task of the modules of this layer is to react autonomously in order to follow the user's work. User and terminal contexts are saved along with the used applications and their requirements in the knowledge base of the system. These data are modified systematically according to any change in the user's choices and actions. The reasoning modules will also use them in order to deduce a general profile that represents the user.

The second layer is for control. Once the user's needs and preferences are identified, the next step consists of verifying that these preferences are converted into the appropriate SLS values.

DOI: 10.1002/nem

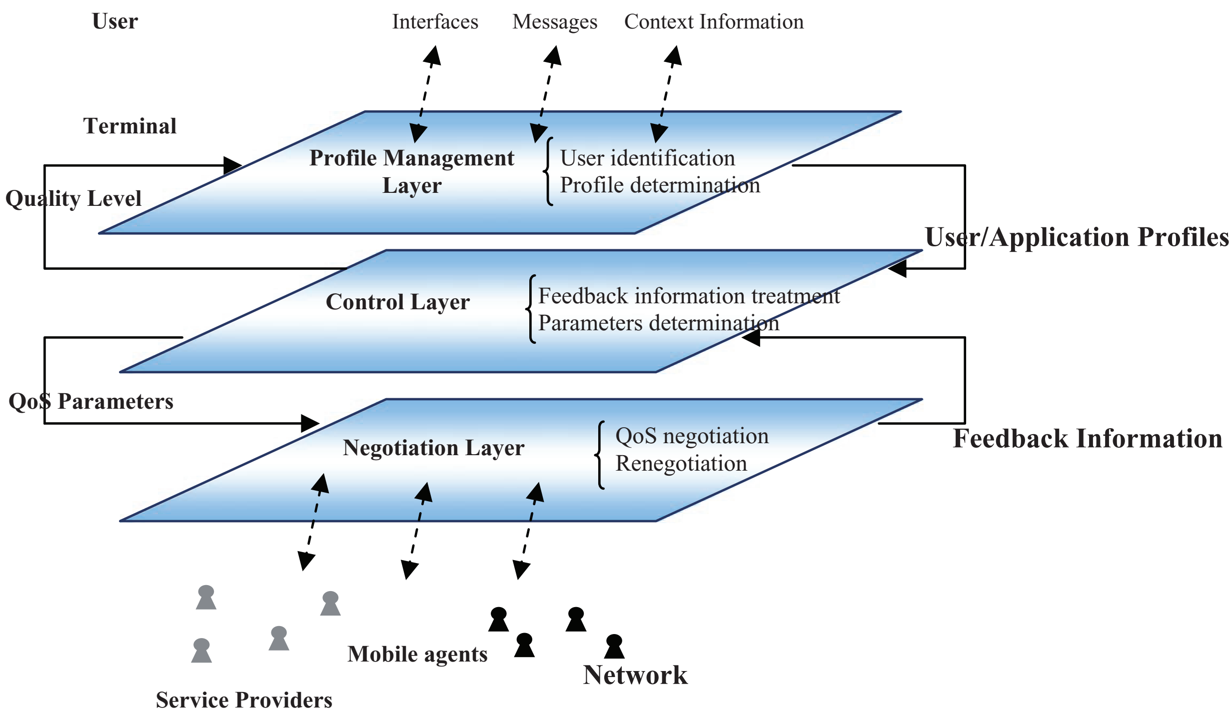

Figure 1. The layers of the proposed framework

The control made in this layer should guarantee a good adaptation of the attributed quality with the real-time user's work independently of the user's mobility or even the user's profile variation. The reasoning mechanism needed at this level is a permanent comparison of input data from the profile manager layer (profiles, characteristics) and those from the negotiation layer (degradation or modification of quality, cancellation of contract between the two sides, new propositions of services). The output of this layer will be SLS parameters values sent to the negotiation layers. These values represent the user's and applications' requirements.

The third layer is the negotiation layer. This layer manages service publication, subscription, selection and negotiation.

4.1 Profile management layer

This layer is introduced to identify the user and to analyse his work. User preferences and application requirements are saved in the knowledge base of the system. These data are modified systematically according to any change in the user's choices and actions. Applications can be classified into many categories according to their needs (delay, jitter . . .) and to the type of supported information (data, voice, image, etc.). The profile of the application will then be determined according to these categories and to the requirements of the user.

As an input to this layer, we can identify two types of data:

- information collected through communication with the user via graphical interfaces or messages;
- information collected through observation of the user's behaviour.

Once the information is analysed, the result is a user profile that represents the user's preferences in terms of quality and an application profile that describes the needs of each application.

DOI: 10.1002/nem

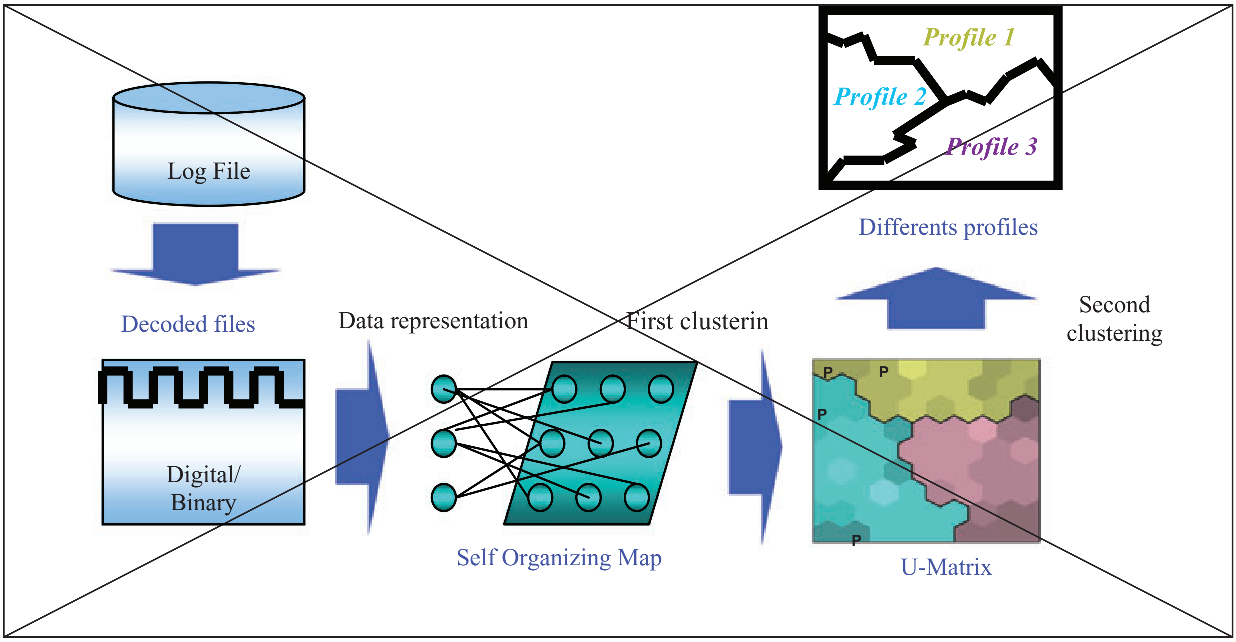

Figure 2. Profile management procedure

Thus this layer plays the role of intermediary between the user and the system. It defines the graphical interfaces needed for communication with the user. It may send him messages or questions and help him choose the answers that best match his profile or needs.

Our approach, represented in Figure 2, consists in recovering, first of all, data that represent traces of use (i.e., log files [18]). These data will be cleaned and recoded in a numerical or binary format to be easily treated. We then build a Self-Organizing Map (SOM) from the recoded file in order to extract profiles [18]. Finally, we carry out a classification to better see the cluster structure of the map, followed by a segmentation of the SOM in order to separate the different profiles.

The steps of our approach represented in Figure 2 are detailed in the following subsections.

The unsupervised connectionist learning

The unsupervised numerical learning, or automatic classification, consists in determining a partition of an instances space from a given set of observations, called the training set. It aims to identify potential trends of data to be gathered into classes. This kind of learning approach, called clustering, looks for regularities from a sample set without being driven by the use of the discovered knowledge. Euclidian distance is usually used by clustering algorithms to measure similarities between observations.

SOMs implement a particular form of competitive artificial neural network [19]; when an observation is recognized, activation of an output cell—competition layer—leads to inhibiting of activation of other neurons and reinforces itself. It is said to follow the 'winner takes all' rule. Actually, neurons are specialized in the recognition of one kind of observation. The learning is unsupervised because neither the classes nor their numbers are fixed a priori.

This type of neural network is organized into a two-dimensional layer of neurons [20]. Each neuron k is connected to n inputs through n exciter connections of respective weights w and to their neighbours with inhibiting links.

The training set is used to organize these maps under topological constraints of the input space. Thus, a mapping between the input space and the network space is constructed; closed observations in the input space would activate closed units of the SOM.

An optimal spatial organization is determined by information received from neural networks. When the dimension of the input space is lower than three, both the position of weight vectors and the direct neighbourhood relations between cells can be visually represented. Thus, a visual inspection of the map provides qualitative information on its architecture.

Connectionist learning is often presented as a minimization of a risk function [21]. In our case, it will be carried out by minimization of the distance between the input samples and the map prototypes (referents), weighted by a neighbourhood function h_{ij}. To do that, we use a gradient algorithm. The criterion to be minimized is defined by

$$E_{\text{SOM}} = \frac{1}{N}\sum_{k=1}^{N}\sum_{j=1}^{M} h_{j\text{NN}(x^{(k)})} \left\| w_{.j} - x^{(k)} \right\|^2$$

Where N represents the number of learning samples, M is the number of neurons in the map, $\text{NN}(x^k)$ is the neuron having the closest referent to the input form x^k, and h is the neighbourhood function. The neighbourhood function h can be defined as

$$h_{rs} = \frac{1}{\lambda(t)}\exp\left(-\frac{d_1^2(r,s)}{\lambda^2(t)}\right)$$

where $\lambda(t)$ is the temperature function modelling the neighbourhood extent, defined as

$$\lambda(t) = \lambda_i \left(\frac{\lambda_f}{\lambda_i}\right)^{\frac{t}{t_{\max}}}$$

where λ_i and λ_f are, respectively, initial and the final temperature (e.g., $\lambda_i = 2$, $\lambda_f = 0.5$), $t_{\max}$ is the maximum number allotted to the time (number of iterations for the x learning sample), and $d_1(r,s)$ is the Manhattan distance defined between two neurons r and s on the map grid, with coordinates (k, m) and (i, j) respectively.

The learning algorithm of this model proceeds essentially in three phases:

- initialization phase, where random values are assigned to the connection weights (referents or prototypes) of each neuron of the map grid;
- competition phase, during which, for any input form $x^{(k)}$, a neuron $\text{NN}(x^k)$, with neighbourhood $V_{\text{NN}(x^{(k)})}$, is selected like a winner. This neuron has the nearest weight vector by using Euclidean distance $d_1(r, s) = |i - k| + |j - m|$:

$$\text{NN}(x^{(k)}) = \underset{1 \le i \le M}{\operatorname{argmin}} \left\| w_{.i} - x^{(k)} \right\|^2$$

- adaptation phase where the weights of all the neurons are updated according to the following adaptation rules:

$$\text{If } w_{.j} \in V_{\text{NN}(x^{(k)})} \text{ then adjust the weights using}$$

$$w_{.j}(t+1) = w_{.j}(t) - \varepsilon(t)\, h_{jNN(x^{(k)})}\left(w_{.j}(t) - x^{(k)}\right):$$

$$\text{else } w_{.j}(t+1) = w_{.j}(t)$$

Repeat this adjustment until SOM stabilization.

SOM map segmentation

We segment the SOM using the K-means method (Figure 3). It is another clustering method that consists in arbitrarily choosing a partition; the samples are then treated one by one. If one of them becomes closer to the centre of another class, it is moved into this new class. We calculate the centres of new classes and we reallocate the samples to the partitions. We repeat this procedure until having a stable partition.

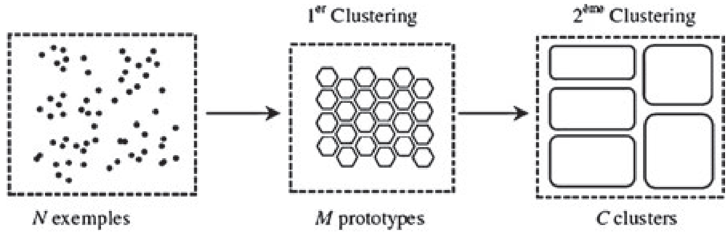

Figure 3. Two successive clusterings: SOM followed by K-means

The criterion to be minimized in this case is defined by

$$E_{K\text{-means}} = \frac{1}{C}\sum_{k=1}^{C}\sum_{x\in Q_k}\|x - c_k\|^2$$

where C represents the number of clusters, Q_k is the cluster k, C_k is the centre of the cluster Q_k or the referent.

The basic algorithm requires fixing K, the number of wished clusters. However, there is an algorithm to calculate the best value for K assuring an optimal clustering. It is based principally on the minimization of the Davies-Bouldin index [22], defined as follows:

$$I_{\text{DB}} = \frac{1}{C}\sum_{k=1}^{C}\max_{l\neq k}\left\{\frac{S_c(Q_k) + S_c(Q_l)}{d_{\text{ce}}(Q_k, Q_l)}\right\}$$

with

$$S_c(Q_k) = \frac{\sum_i \|x_i - c_k\|^2}{|Q_k|} \quad \text{and} \quad d_{\text{cl}}(Q_k, Q_l) = \|c_k - c_l\|^2$$

where C is the number of clusters, S_c is the intra-cluster dispersion, and d_{cl} is the distance (centroid linkage) between the clusters centres k and l. This clustering procedure aims to find internally compact spherical clusters which are widely separated.

There are several methods to segment the SOMs [23]. Usually, they are based on visual observations and manual assignment of map cells to clusters. Several methods use the K-means algorithm with given ranges for the K value. Our work is based on the approach of Davies–Bouldin index minimization.

We note that the K-means approach can be directly applied to the data instead of the SOMs approach. In our work, we applied it to the SOMs results. The idea is to use SOMs as a preliminary phase in order to set a sort of data pretreatment (dimension reduction, regrouping, visualization, etc.). This pretreatment has the advantage of reducing the cluster calculation complexity and also ensures better visualization of the automatic classification results.

Moreover, the use of SOMs for visualization is crucial, especially in the case of data multivariate: dimension >2 or 3. In this last case, the SOMs permit, on one hand, reduction of the data space dimension, and on the other hand, visualization of the clusters in the plan.

Simulation results

We applied the two algorithms described above to our data (log files describing different traces of use) in order to determine the negotiation profiles. In the simulations, we used the *SomToolbox* proposed by Helsinki University of Technology (HUT) researchers of the T. Kohonen team [24]. The results obtained are very promising (Figure 4).

Figure 4(a) is a representation of a SOM map seen as '*component planes*' that allows visualization of the partition of different variable values. The highest values of the variables are in red and the lowest values are in blue. This representation allows us to identify the clusters structure of the map.

DOI: 10.1002/nem

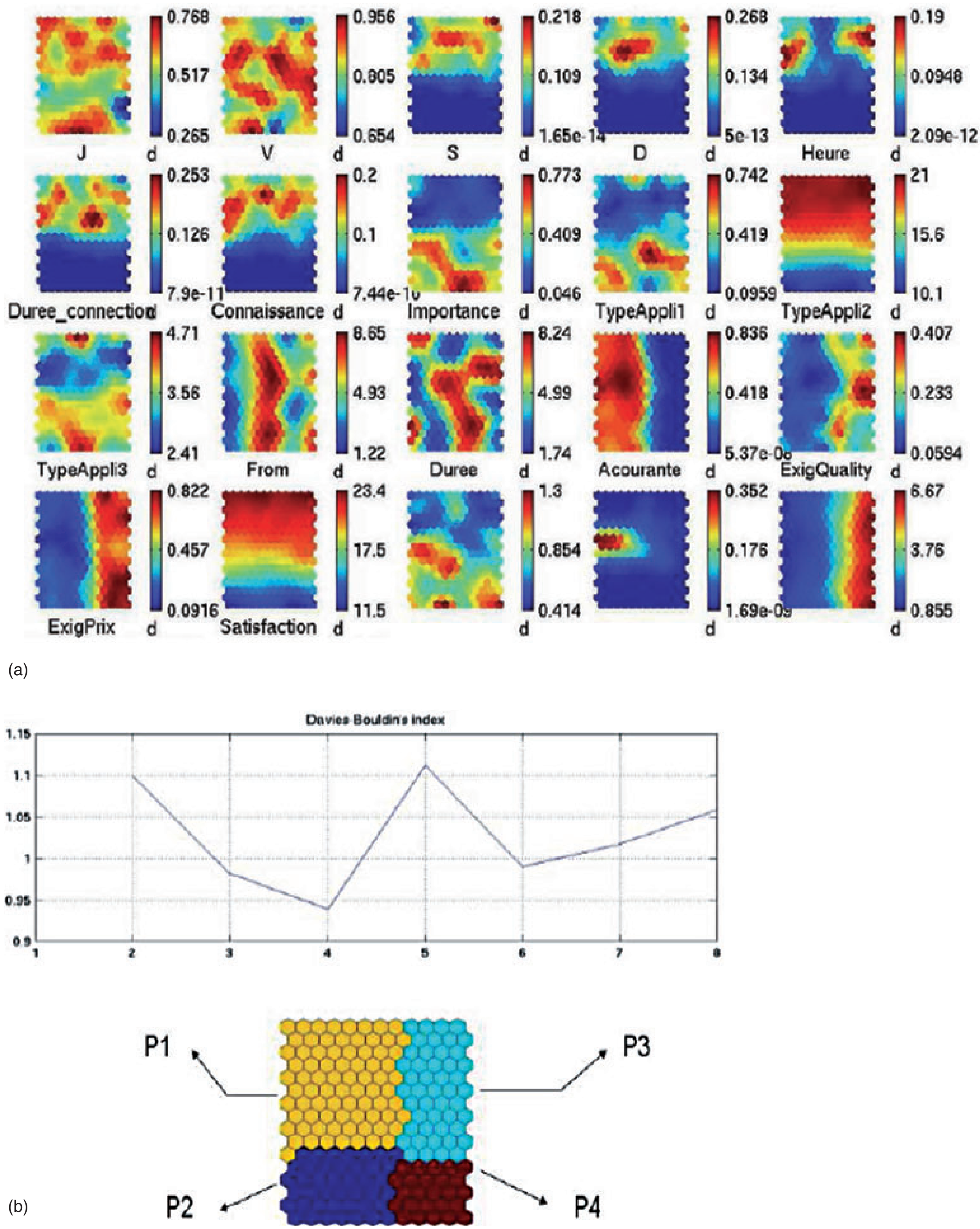

Figure 4. (a) SOM clustering. (b) Resulting profiles. A colour version of this figure is available online at www.interscience.wiley.com

Figure 4(b) represents neuron segmentation of the SOM map (second clustering). The curve shows that the minimal value of the Davies–Boudin index corresponds to an optimal clustering resulting in four profiles.

The classes finally obtained are coherent although they result from an unsupervised classification without any pre-established class before treatment. These results are thus encouraging because they make it possible to interpret the obtained profiles.

4.2 Control layer

Once the user's needs and preferences are identified in a negotiation profile, the next step consists in verifying that these preferences are converted into the appropriate parameters values [25].

The control made in this layer should also guarantee a good adaptation of the attributed quality with real-time user's work independently of the user's mobility or even the user's profile variation. The reasoning mechanism needed at this level is a permanent comparison of input data from the profile manager layer (profiles, characteristics) and those from the negotiation layer (degradation or modification of quality, cancellation of contract between the two sides, new propositions of services). The output of this layer will be QoS parameters values sent to the negotiation layer. These values are related to the negotiation profile determined in the previous layer and represent the user's and applications' requirements. The control layer accomplishes these functionalities:

- It determines the values that should be attributed to all of the negotiation parameters depending on the constraints of the system, service providers and negotiation profile.
- It establishes the link between the user and the service provider. It verifies that both of the two parts respect the negotiated services. The satisfaction of the user is deduced from analysis of his behaviour.
- It analyses the feedback information collected by the responsive agents in order to take decisions concerning renegotiation and to evaluate the state of the link and the video packet transmission.

4.3 Negotiation layer

The negotiation layer is responsible for negotiation of the QoS parameters received from the control layer. This negotiation is accomplished in two ways. The first is done using a negotiation protocol [40] (see also Section 6 for further details) between the user terminal and network entities according to the QoS profile determined by the NIA. The second one is the negotiation of service level with the different service providers currently available. The following paragraphs detail this negotiation process.

The negotiation layer may also ask for a change in the required services according to the needs of the user and applications. This corresponds to a request for renegotiation between the two parts. Mobile agents [26,27], are sent to the service providers in order to bring new offers. As mentioned in Figure 5, three types of agents are proposed [28]:

- *User negotiator (UN)*. This is a mobile agent that is sent by the user overseer on the platform of the provider (access mediator). Its first task is to survey the offers made by the providers. This is the discovery phase. The second phase is the negotiation which takes place when the user has a specific need.
- *Access negotiator (AN)*. This is created by the provider to each mobile agent (UN) that arrives on its platform in order to negotiate services in favour of the provider.
- *User overseer (UO)*. This manages the entire negotiation process on behalf of the user. It sends a UN for the service survey and negotiation. It collects the results of the different negotiation threads and then makes the final decision.

A possible solution would be to create a marketplace where all providers could propose their offers and negotiate with interested customers. However, this solution asks a high degree of cooperation between providers. From the telecommunications point of view, this is not the preferred solution. We based our solution on the assumption that providers are mostly competitors.

DOI: 10.1002/nem

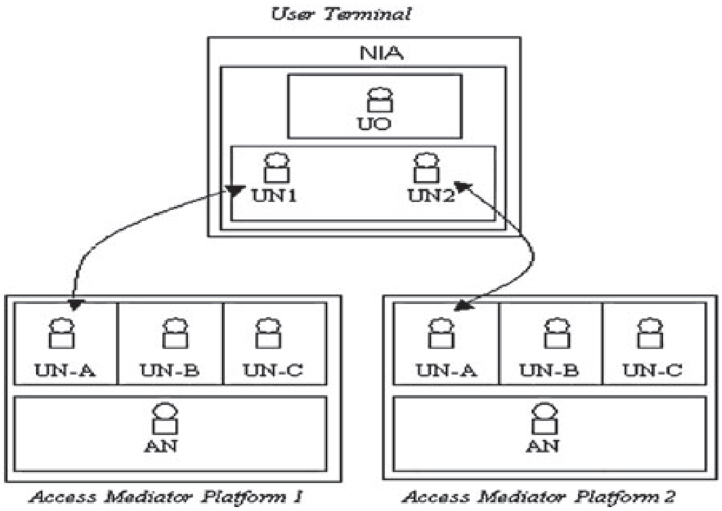

Figure 5. Framework for the dynamic negotiation of SLA/SLS

The negotiation process

The proposed solution has two phases: a survey phase when the user agent discovers the new offers made by the providers, and a negotiation phase which takes place when the client has specific QoS needs.

In the first phase, every provider opens access to a multi-agent platform on its site that would welcome customers' UN agents. The UO sends a UN to the provider's platform with which he has a predefined contract. The UN takes the user profile that corresponds to the QoS characteristics of the applications the user often uses. Access to the platform requires an authentication phase. The UN then starts to survey the offers made by the provider. When new offers are proposed, the UN filters them and sends them to his UO that matches the user's profile. The mobile agents stay on the provider's platforms and continuously survey the publication of new offers.

The second phase starts when a customer has a specific need. The UO filters the offers sent by the UN and sends its needs to the UNs concerned. The UNs then open negotiations with their AN counterparts to obtain the best rates and QoS. When the UN and AN reach an agreement (or after a certain period of time where no agreement is reached), the UN sends to the overseer the description of the best proposal made by the AN. The overseer then compares the different offers it received from the different ANs and sends back an agreement to the best one. Every agent concerned by the different concurrent negotiations then returns to its normal state.

To implement our model and protocol of negotiation, we used the platform JADE (Java Agent Development Framework). JADE [29] is a software framework fully implemented in the Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with FIPA specifications and through a set of graphical tools that supports the debugging and deployment phases. The agent platform can be distributed across machines (which do not even need to share the same OS) and the configuration can be controlled via a remote GUI.

Further details concerning the implementation of our model and the negotiation protocol can be found elsewhere [31,30].

5. MOBILITY AND QoS PROFILE

The user's mobility profile is built on the basis of its behaviour/movement after **m** associations with the system. The goal of this profile is to build a user's behaviour model.

DOI: 10.1002/nem

Our scheme is applied within the heterogeneous network, so like cellular networks we think that it is easy for the system to know the localization of the user. In this context, a cell is the geographic area covered by an access point or base station.

The system model is based on the continuous-time Markov chain (CTMC).

Our system can evolve between N states defined by the following set: C = ($C_1, C_2, \ldots, C_i, \ldots, C_n$).

The system is in state i = the terminal mobile in cell C_i. P_{ij} is the probability of transition from cell C_i to cell C_j. $P_i(t_r)$ defines the location probability of the mobile terminal in cell C_i at time t_r.

The user's mobility profile contains the following information:

- the user's identifier;
- user preferences: User_P.

This attribute represents the set of user's preferences and is determined by the NIA.

The user's preferences are determined after an observation phase in which the system observes the user's behaviour.

Figure 6 represents the determination of User_P. The proposed format for User_P is as follows:

User_P = <Preference ID> <Duration_P> <Cell_P> <QoS_level>
<Preference ID> identifies the preference (the system can detect several preferences for the user);
<Duration_P>: <start_P> <end_P> determines the period of time in which the user's preference is satisfied;
<Cell_P> determines the cell in which the user's preference is satisfied;
<QoS_level> is the QoS level needed by the user for the preference.

- **M** = [P_{ij}] [N*N]: the matrix of transition, which contains P_{ij}; before **m** associations, P_{ij} are random.

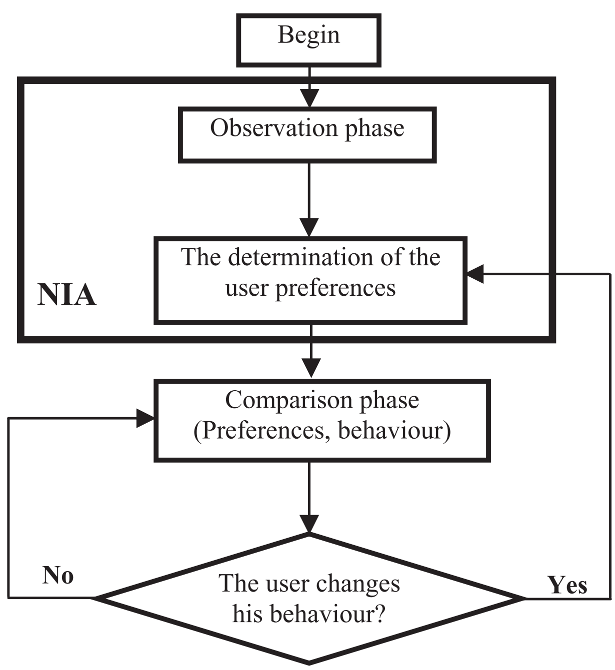

Figure 6. Determination of user preferences

DOI: 10.1002/nem

We note **t** $[i, j]$: the number of transitions from cell i to cell j during **m** associations with the system; $\mathbf{g}(i)$: the number of transitions outgoing from the cell i during the **m** associations with the system. We calculate it as follows: $\mathbf{g}(i) = \sum_{j=1}^{n} t[i, j]$.

After **m** associations, the probability of transition from the cell i to cell j is calculated as follows:

$$P_{ij} = t[i, j]/\mathbf{g}(i)$$

- $\mathbf{V} = [P_i(t_o)]\ [N]$: this vector contains $P_i(t_o)$.

t_o corresponds to the beginning of each communication; $P_i(t_o)$ defines the location probability of the mobile terminal in cell C_i at time t_o.

Where $k(i)$ is the number of associations with cell i during the **m** associations at time t_o, we have

$$\sum_{i=1}^{n} k(i) = \mathbf{m} \quad \text{and} \quad P_i(t_o) = k(i)/\mathbf{m}$$

- The MSpec (mobility specification): MSpec determines the future locations of the mobile terminal.

The proposed format for MSpec is as follows: MSpec = <MSpec ID> <Duration> <Cell ID>.

MSpec ID is the identifier of MSpec.
Duration is the interval of time (<start time>, <end time>) during which the future locations of the mobile terminal can be determined.
Cell ID: <cell ID1>, <cell ID2>, <cell ID3>, . . . , <cell IDn> is a set of cell identifiers. We suppose that each cell is identified by a single identifier.

We have $P_j(t_{r+1})$: the probability of the mobile terminal's location in cell C_j at time t_{r+1}. We can calculate this probability using the following formula:

$$P_j(t_{r+1}) = \sum_{i=1}^{n} P_i(t_r) * P_{ij}$$

We define $\theta(0 \leq \theta \leq 1)$, which is a fixed or variable threshold. It is used to select cells according to their probabilities. The MSpec is defined as follows: $\text{MSpec}(t_r) = \{C_j / P_j(t_{r+1}) \geq \theta\}$.

Before **m** associations, the system does not calculate MSpec because the user is new and the system does not have the necessary information to calculate MSpec; it has no information concerning the **M** matrix and the **V** vector (observation phase).

6. EXAMPLE

In this section we present a scenario of communication between two mobile terminals where MH1 is the entity which generates the flow and the mode is *sender-initiated reservation.*

In this scenario the two entities which communicate are mobile and MH1 generates flow with the mode *sender-initiated reservation.* Therefore, MH1 represents the NI (NSIS initiator: the signalling entity that makes the resource request, usually as a result of a user application request), MH2 represents the NR (NSIS responder: the signalling entity that acts as the endpoint for the signalling and can optionally interact with applications as well) and the ARs as well as the MAPs represent the NF (NSIS forwarder: the signalling entity between an NI and NR that propagates NSIS signalling through the network).

We note MSpec1 and MSpec2, respectively, are the sets of future localisations of MH1 and MH2 during the communication.

Figure 7 shows the Advance resource reservation procedure, the negotiation of the QoS level between MAP1 and MAP2 is done using the negotiation protocol SLN NSLP [40] according to the user's QoS

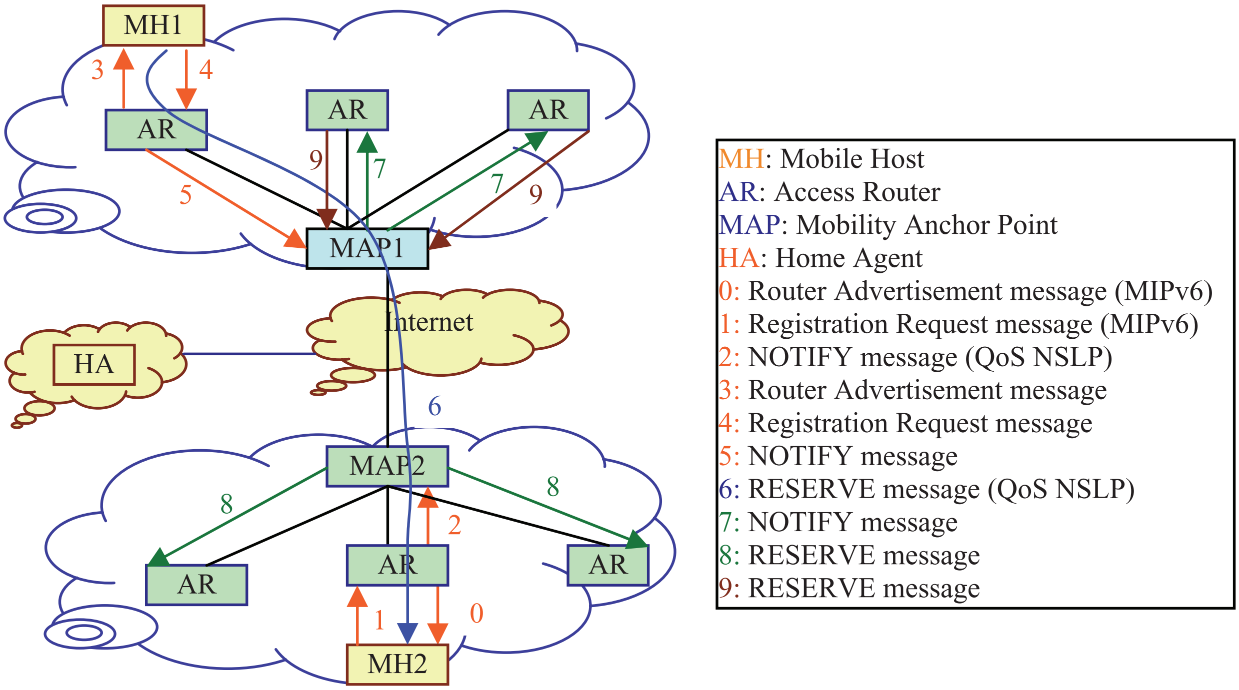

Figure 7. Advance resource reservation procedure

profile determined by the NIA. This QoS profile contains the user's preferences in term of QoS for each application used. Then, MQoS NSLP is used for advance resources reservation.

Concerning the advance resources reservation using QoS NSLP, the procedure is as follows (the registration can start with MH1 or the MH2; the following scenario considers that MH2 is the first mobile, which makes the registration. The mode used is the *sender-initiated reservation* mode):

0. The AR informs MH2 with the message *router advertisement* of the availability of resources. For that, we propose to add a bit Q in this message. If $Q = 0$ then the AR does not have resources and in this case MH2 can be connected in BE.
1. During registration, MH2 asks its AR for a certain QoS. In this case, we propose to add the MSpec2 object to the *registration request* message. (Here, we are interested only in the interactions between MIPv6 and the QoS NSLP messages; other MIPv6 messages are necessary in order to continue the registration.)
2. After registration with MH2, the AR sends the QoS request to MAP2. For that, we use the NOTIFY message with the MSpec2 object included in it. After reception of the NOTIFY message, MAP2 analyses the MSpec2 object.
3. The AR informs MH1 with the *router advertisement* message of the availability of resources using the bit Q. If $Q = 0$ then the AR does not have resources and in this case MH2 can only be connected in BE.
4. During registration, MH1 asks its AR for a certain QoS. The MSpec1 object is added to the *registration request* message.
5. After registration with MH1, the AR sends the QoS request to MAP1, for which we use the NOTIFY message and the MSpec1 object. After reception of the NOTIFY message, MAP1 analyses the MSpec1 object.
6. To reserve resources between MH1 and MH2, MH1 (NI) sends the RESERVE message, which must contain the QSpec object. This message is transported by GIMPS to MAP1, sent to MAP2, to the AR and finally to MH2 (NR).

DOI: 10.1002/nem

7. After reception of the RESERVE message, MAP1 sends the NOTIFY message to all the ARs, which are in MSpec1 in order to receive the RESERVE message.
8. The RESERVE message is forwarded after its reception by MAP2, in all ARs which are in MSpec2.
9. The ARs which are in MSpec1 send the RESERVE message to MAP1.

6.1 The handover procedure

The stages of the handover procedure are as follows:

a. Registration of MH2 with its new AR (MIPv6 protocol).
b. Establishment of the new path and update of the resources reservation:
b1. The new AR sends the RESERVE message to MH2 (message 1 in Figure 8).
b2. MH2 sends the RESPONSE message with the new MSpec2 (message 2 in Figure 8).
b3. After reception of the RESPONSE message, the new AR sends the NOTIFY message to MAP2 with the new MSpec2 (message 3 in Figure 8).
b4. MAP2 analyses the new MSpec2 and sends the corresponding RESERVE message (message 4 in Figure 8):
c. Registration of MH1 with its new AR (MIPv6 protocol).
d. Establishment of the new path and update of the resources reservation:
d1. MH1 sends the RESERVE message to the new AR with the new MSpec1; it will be forwarded to MAP1 (message 5 in Figure 8).
d2. MAP1 includes the old and new MSpec1 in a NOTIFY message. It then sends this message to all the ARs whose identification is in the new and old MSpec1 (message 6 in Figure 8).

Each AR analyses the two MSpec1 objects and sends the corresponding RESERVE message (message 7 in Figure 8).

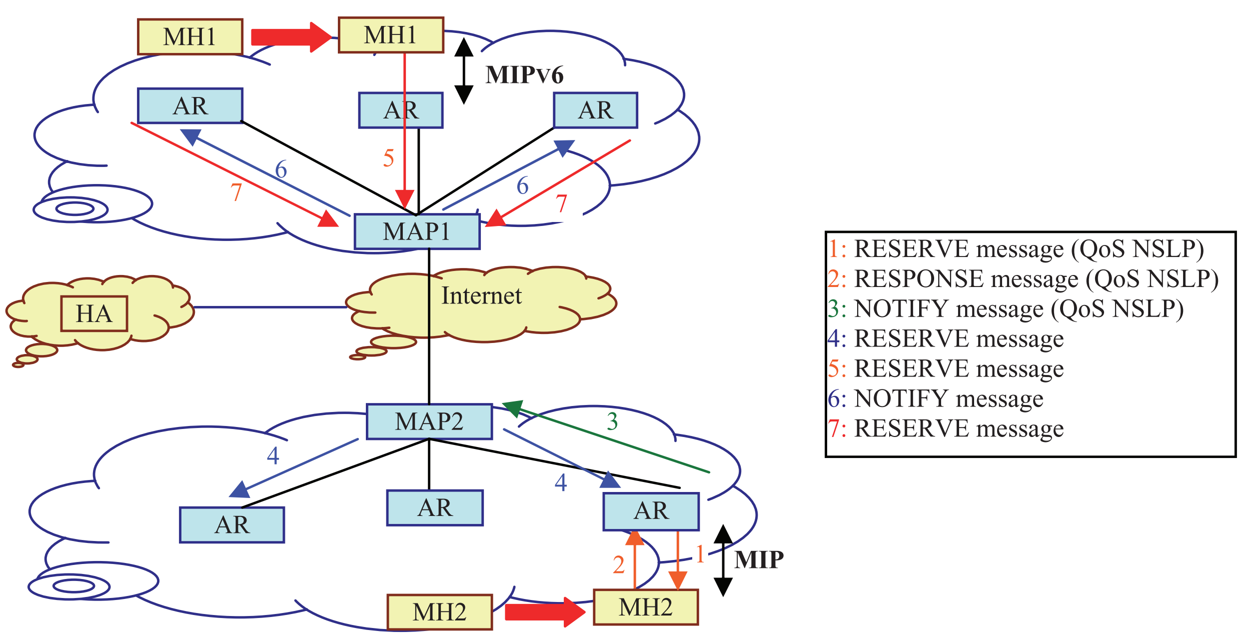

Figure 8. The handover procedure

7. PERFORMANCE EVALUATION

The network configuration used for the simulation is shown in Figure 9. The studied scenario represents a communication between a fixed correspondent node (CN) and a mobile host. The simulation contains two stages: the first stage consists in seeking a good value for θ; for this, we use MATLAB mathematical software. The second stage consists in validating our approach of resources management using OMNeT++ simulation tools.

7.1 The first stage (MATLAB tools)

For the observation phase, we take nb = 40 (the observed communications). During these 40 communications, we will follow the different locations of the mobile terminal in order to determine the **M** matrix and the **V** vector. After the observation phase, the **M** matrix and the **V** vector are fixed.

The system calculates the vector $\mathbf{V}_1 = \mathbf{V} * \mathbf{M}$ in order to determine MSpec1 for the first handover. For the second handover, the system calculates the vector $\mathbf{V}_2 = \mathbf{V}_1 * \mathbf{M}$ in order to determine MSpec2 and so on.

After six handovers, we have the following results:

$$\mathbf{V}_1 = [0.2712\ 0.0700\ 0.1350\ 0.0665\ 0.1900\ 0\ 0.1440\ 0\ 0.0920\ 0.0313]$$

$$\mathbf{V}_2 = [0.2015\ 0.1193\ 0.1089\ 0.1558\ 0.0504\ 0.1404\ 0.0271\ 0.0940\ 0.0343\ 0.0683]$$

$$\mathbf{V}_3 = [0.2217\ 0.1119\ 0.1531\ 0.0873\ 0.1308\ 0.0406\ 0.0876\ 0.0331\ 0.0791\ 0.0548]$$

$$\mathbf{V}_4 = [0.2147\ 0.1275\ 0.1263\ 0.1350\ 0.0689\ 0.0941\ 0.0433\ 0.0693\ 0.0500\ 0.0709]$$

$$\mathbf{V}_5 = [0.2176\ 0.1225\ 0.1509\ 0.1022\ 0.1064\ 0.0543\ 0.0684\ 0.0446\ 0.0696\ 0.0635]$$

$$\mathbf{V}_6 = [0.2159\ 0.1296\ 0.1361\ 0.1250\ 0.0792\ 0.0777\ 0.0501\ 0.0597\ 0.0569\ 0.0698]$$

Figure 10 shows the impact of θ on MSpec size. If $\theta \geq 0.28$, MSpec is empty, so, it is not interesting to take a value of θ which is greater than 0.28. We remark that $\theta = 0.1$ is a good value for the simulation. In this case, MSpec contains four or five cells.

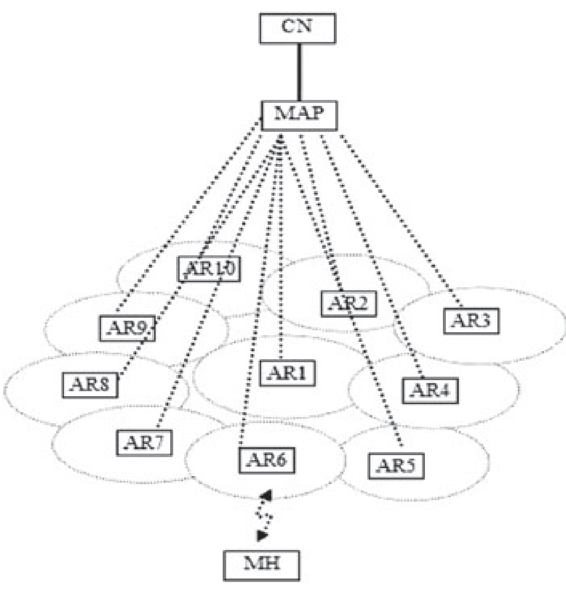

Figure 9. The network configuration used for the simulation

DOI: 10.1002/nem

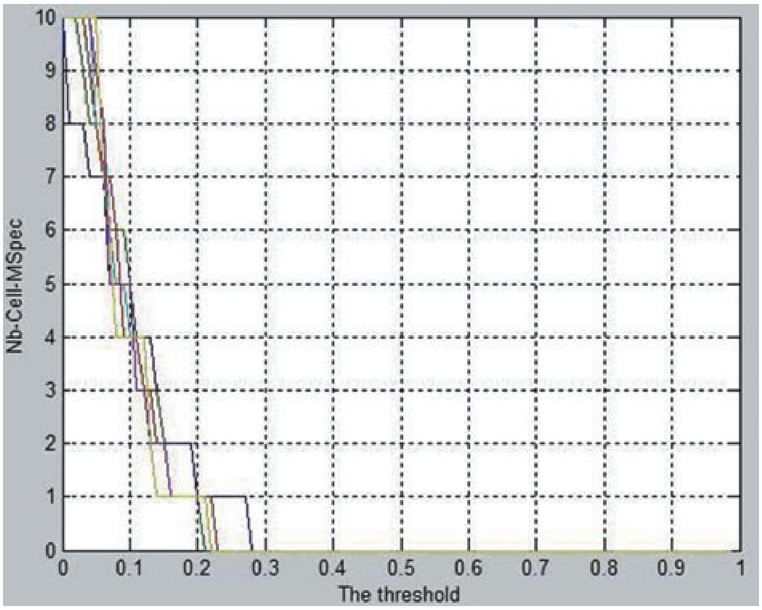

Figure 10. Impact of θ on MSpec size

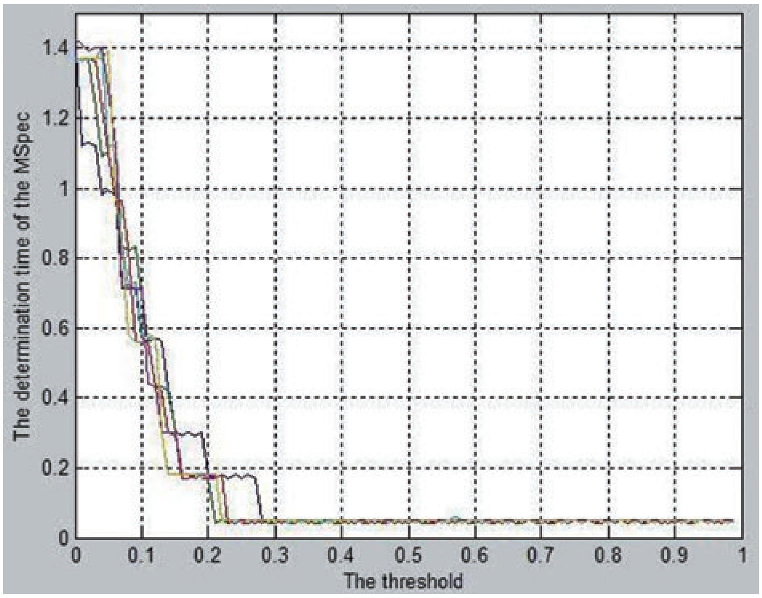

Figure 11. Impact of θ on the determination time of MSpec

With $\theta = 0.1$, we have the following results:

$$\text{MSpec } 1 = \{C_1, C_3, C_5, C_7\}, \text{MSpec } 2 = \{C_1, C_2, C_3, C_4, C_6\}, \text{MSpec } 3 = \{C_1, C_2, C_3, C_5\},$$
$$\text{MSpec } 4 = \{C_1, C_2, C_3, C_4\}, \text{MSpec } 5 = \{C_1, C_2, C_3, C_4, C_5\}, \text{MSpec } 6 = \{C_1, C_2, C_3, C_4\}$$

Figure 11 shows the impact of θ on the determination time of MSpec (the time is calculated in milliseconds).

DOI: 10.1002/nem

7.2 The second stage (OMNeT++ tools)

The simulation is based on the modelling of passive and active reservations: an active reservation is made by the mobile host in the current cell and a passive reservation is made by the MAP entity in the neighbouring cells identified by the MSpec. So, in order to model the reservations, we define three different kinds of service class with different priorities:

- The *C0 class* represents best-effort calls.
- The *C1 class* represents guaranteed calls.
- The *C2 class* represents handover calls.

In order to improve the QoS during handover, handover calls have the highest priority and best effort calls the lowest priority.

We have: Priority (*C0 class*) < Priority (*C1 class*) < Priority (*C2 class*).

For modelling, a *C1 class* request models an active reservation and a *C2 class* request models a passive reservation; a *C0 class* request does not require a resources reservation.

In our strategy of resources management, the capacity of the cell is shared between the three service classes; the resources of the passive reservation can be used by the best effort calls.

An admission control for each access router is used to decide if a new call is accepted or not. This decision is based on the available resources in the cell.

For the simulation model, we have several resources reservation requests which are generated by the two service classes (*C1 class* and *C2 class*), for which we use the OMNeT++ simulation tools in order to measure the influence of these resources reservations on network performance (handover blocking probability, signalling load and reservation re-establishment delay).

Thus, the resources reservation mechanism (with passive and active reservations) will be translated into a channel reservation mechanism in a cellular network (for *C1 class* and *C2 class*).

Traffic characterization

Each cell supports 24 traffic channels (a fixed channel allocation).

- The new traffic is generated according to a Poisson process with parameter γ.
- Call partitioning between the *C0 class*, the *C1 class* and the *C2 class* is (3 : 1 : 1) calls.
- Call durations are modelled as independent random variables, following an exponential distribution with the parameter μ ($1/\mu = 120$ s).
- The session duration follows an exponential distribution with the parameter μ_s ($1/\mu_s = 40$ s).
- The resource partitioning between the *C0 class*, *the C1 class* and the *C2 class* is (1 : 3 : 3) channels.

For the simulation, we consider the following parameters:

- The arrived calls rate $\gamma = [400, 1400]$ calls/hour.
- The MSpec failure rate = [0%, 100%].

In order to measure the performance, we consider the following parameters:

- The handover blocking probability (Ph).
- The signalling load.
- The reservation re-establishment delay.

Figure 12 shows the impacts of the arrived calls rate on the handover blocking probability measured with our resources management approach and a classical approach for resources reservation. The classical approach represents a simple way to make a resources reservation without the advance resources reservation scheme.

In this case, the failure rate of MSpec = 0%, so, for each AR, we have three traffic channels reserved for the first *C2 class* call. Thereafter, each arrived call of the *C2 class* reserves three traffic channels for the next arrived call for the same class.

DOI: 10.1002/nem

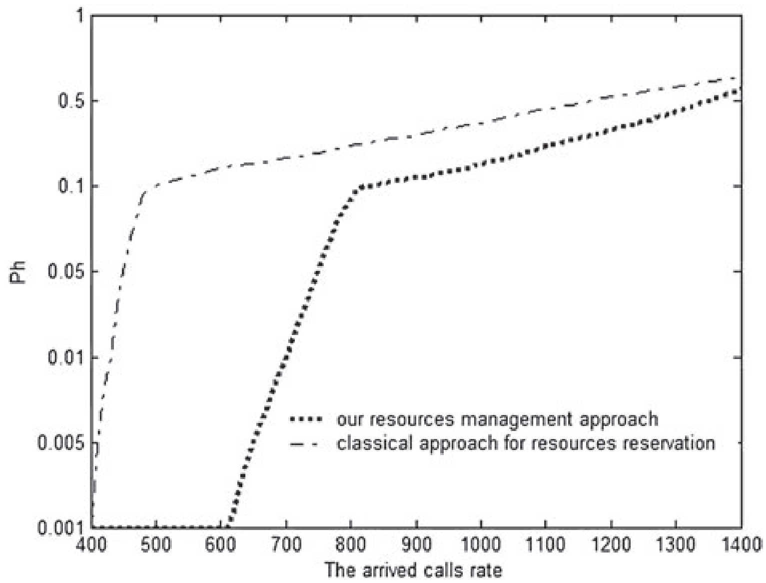

Figure 12. Impacts of the arrived calls rate on handover blocking probability

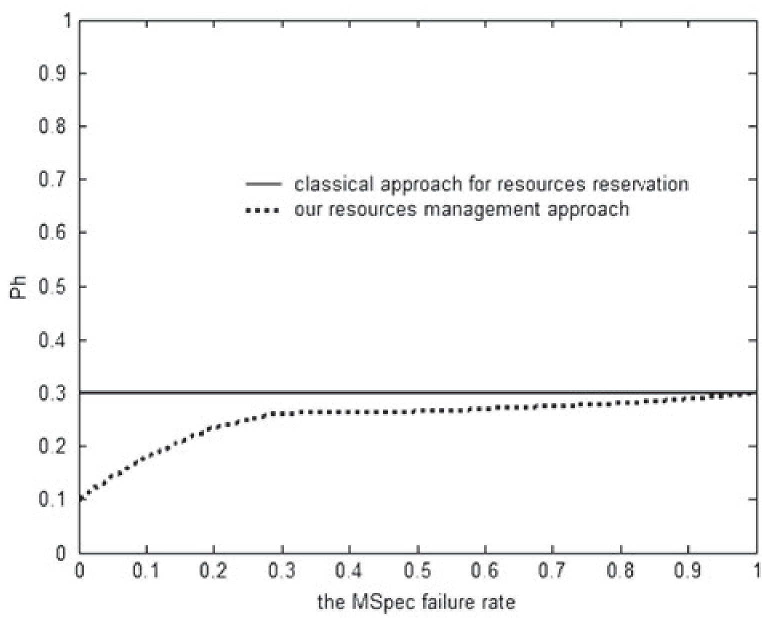

Figure 13. Impact of MSpec failure on the handover blocking probability

Our resources management approach reduces the handover blocking probability compared to a classical resources reservation approach.

In the following, we take $\gamma = 900$ calls/hour and we want to study the impact of MSpec failure rate on handover blocking probability as well as on the signalling load on the end-to-end link (between CN and MH).

MSpec failure rate modelling is as follows: if the MSpec failure rate = x%, on 100 handovers, x handovers do not have a passive resources reservation. Figure 13 shows the impact of MSpec failure rate on handover blocking probability. With a good MSpec, we can clearly reduce the handover blocking

DOI: 10.1002/nem

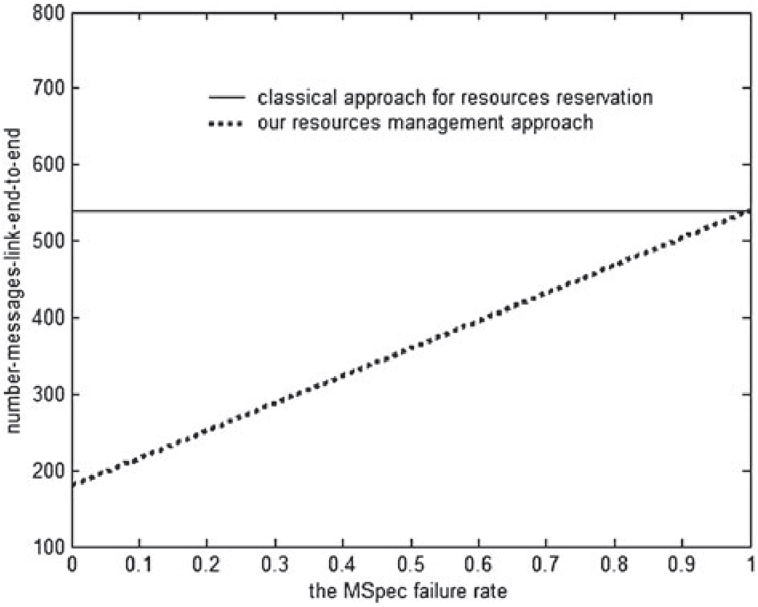

Figure 14. Impact of MSpec failure rate on the signalling load on the end-to-end link

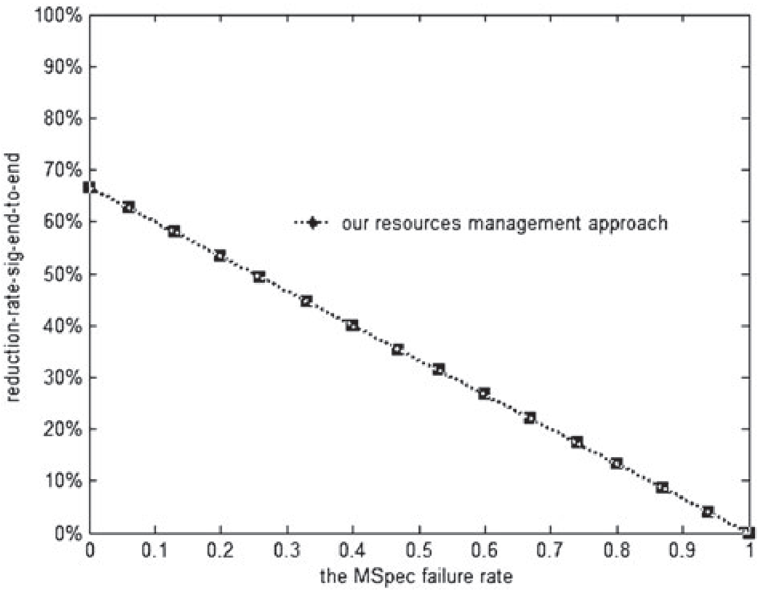

Figure 15. Impact of MSpec failure rate on the signalling delay

probability. Figure 14 shows the advantage of our resources management approach from a signalling point of view. Our approach minimizes the number of messages on the end-to-end link. Figure 15 shows the impact of MSpec failure rate on the reduction rate of the end-to-end signalling delay of our approach compared to a classical resources reservation approach. The end-to-end signalling delay is defined as the period of time it takes a RESERVE message to traverse from CN to MH.

Thus, our resource management approach in a mobile environment reduces the handover blocking probability, minimizes the reservation establishment delay and reduces the end-to-end signalling load

DOI: 10.1002/nem

(between CN and MH). The good performance of our scheme is related to the dynamic management of resources through active and passive reservations.

Compared to the research into extending the RSVP protocol to mobile integrated services networks, our approach of resource management can be applied in both DiffServ and IntServ networks since QoS NSLP is independent of the QoS model used (DiffServ, IntServ). Since we specified the user's mobility specification (MSpec object) as well as the choice of mobility architecture (HMIPv6), another advantage of our approach concerns the fact that we do not need new QoS NSLP messages or a new entity in the network (like a proxy agent) in order to make an advance resources reservation compared to the majority of research which extends RSVP in the mobile integrated services networks.

8. CONCLUSION

In this paper, we propose a signalling environment for QoS negotiation and advance resource reservation in mobile IP networks. This environment is built in conformance with the generic signalling environment, which is standardized by the NSIS IETF working group. We have presented a mobility and QoS profile used for QoS negotiation and advance resource reservation in a mobile environment. This reservation is made according to the MSpec object which determines the future locations of the mobile terminal. Our objective through this approach is to minimize the degradation of services during the handover. Also, our Negotiation Individual Assistant (NIA) presented in this paper constitutes an interface between the user and the network in the context of the new-generation Internet. This interface integrates two interesting techniques of the Intelligence Artificial domain: connectionist learning and agent technology.

A clustering algorithm based on topological Self-Organizing Maps (SOMs) is used (in the first layer of the NIA) to determine a negotiation profile that represents the user's preferences and applications needs. This profile is then used in the second layer of the NIA in order to find the appropriate values for SLS parameters. On the other side, agent technology is used (in the third layer of the NIA) in order to help the user choose the best service provider, dynamically negotiate the SLS on the user's behalf, follow the user's behaviour to be able to anticipate the negotiations and manage the renegotiations. The introduction of a multi-agents system, on both the user and provider sides, has shown good performance in the choice of the best service provider. The agents of our system communicate via an FIPA protocol [32]. This approach has many advantages. The terminal charge is reduced, the system can function on a large range of terminals and service providers can more easily propose new services. Our NIA has been implemented and tested in two French national research projects: ARCADE [33,34] and IPSIG [35].

REFERENCES

1. QoSforum. *QoS Protocols and Architectures*. White Paper. July 1999.
2. Yavatkar R, Pendarakis D, Guerin R. A framework for policy based admission control. *RFC 2753*, January 2000.
3. Goderis D *et al*. Service level specification semantics and parameters. draft-tequila-sls-02.txt. Internet draft, January 2002.
4. Van den Bosch S, Karagiannis G, McDonald A. NSLP for quality-of-service signalling. draft-ietf-nsis-qos-nslp-10.txt. Internet draft, March 2006.
5. Talukdar AK, Badrinath BR, Acharya A. MRSVP: a resource reservation protocol for an integrated services network with mobile hosts. *ACM Journal of Wireless Networks*. 2001; **7**: 5–19.
6. Talukdar AK, Badrinath BR, Acharya A. Integrated services packet networks with mobile hosts: architecture and performance. *Wireless Networks* 1999; **5**(2): 111–124.
7. Talukdar AK, Badrinath BR, Acharya A. On accommodating mobile hosts in an integrated services packet network. In *Proceedings of the INFOCOM '97: 16th Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*. 9–11 April 1997; 1046.

DOI: 10.1002/nem

8. Kim M-S, Suh Y-J, Kim Y-J, Choi Y. A resource reservation protocol in wireless mobile networks. In *ICPP Workshops*, Valencia, Spain 3–7 September 2001.
9. Levine DA, Akyildiz IF, Naghshineh M. The shadow cluster concept for resource allocation and call admission in ATM-based wireless networks. In *Proceedings of the 1st Annual International Conference on Mobile Computing and Networking*, Berkeley, CA; 142–150.
10. Le Grand G, Ben Otman J, Horlait E. Réservation de ressources dans un environnement mobile avec MIR: Mobile IP Reservation Protocol. In *CFIP 2000 Colloque Francophone sur l'Ingénierie des Protocoles*, Toulouse, 17–20 October 2000.
11. Horlait E, Bouyer M. CLEP (Controlled Load Ethernet Protocol): bandwidth management and reservation protocol for shared media. draft-horlait-clep-00.txt. Internet draft, July 1999.
12. Ferrari D, Gupta A, Ventre G. Distributed advance reservation of real-time connections. In *Proceedings of the 5th International Workshop on Network and Operating System Support for Digital Audio and Video*, 19–21 April 1995; 16–27.
13. Samaan N, Karmouch A. An evidence-based mobility prediction agent architecture. In *MATA 2003*, Berlin; pp. 230–239.
14. Akyildiz IF, Wang W. The predictive user mobility profile framework for wireless multimedia networks. *IEEE/ACM Transactions on Networking* 2004; **12**: 1021–1035.
15. Aljadhai A, Znati TF. A framework for call admission control and QoS support in wireless environments. In *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99)*, Vol. 3, March 1999; 1019–1026.
16. Schulzrinne H, Hancock R. GIST: General Internet Signaling Transport. draft-ietf-nsis-ntlp-09. Internet draft, February 2006.
17. Ash J, Bader A, Kappler C. QoS NSLP QSpec template. IETF, draft-nsis-qspec-09. Internet draft, March 2006.
18. Benabdeslem K. Approches connexionnistes pour la visualisation et la classification des données issues d'usages de l'Internet. Thesis in Computer Science, LIPN, University of Paris 13, France, December, 2003.
19. Kohonen T, Kaski S, Lappalainen H. Self-organized formation of various invariant-feature filters in the adaptive-subspace SOM. *Neural Computation* 1997; 1321–1344.
20. Zeharoui F, Bennani Y. M-SOM-ART: Growing self organizing map for sequence clustering and classification. In *European Conference on Artificial Intelligence (ECAI)*, Valencia, Spain, August 2004.
21. Bennani Y. Réseaux de neurones artificiels, In *Encyclopedie d'Informatique et Science de l'information*, Edition Vuibert: Paris, 2005.
22. Davies L, Bouldin W. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1979; **1**(2): 224–227.
23. Juha A, Esa A. Clustering of the self-organizing map. *IEEE Transactions on Neural Networks*, 2000; **11**(3): 586–600.
24. Alhoniemi E, Himberg J, Parhankangas J, Vesanto J. SOM Toolbox 2000. http://www.cis.hut.fi/projects/somtoolbox/ [15 November 2007].
25. Jrad Z, Krief F. An intelligent interface for the dynamic negotiation of QoS in ARCADE. In *ACS/IEEE International Conference on Pervasive Services (ICPS'2004)*, Beirut, Lebanon, July 2004.
26. Briot J-P, Demazeau Y. *Principles and Architecture of MAS*. Hermès: Paris 2002; pp 235–266.
27. Agoulmine N, Fonseca M, Marshall A. *Multi-Domain Policy Based Management Using Mobile Agents*. Lecture Notes in Computer Science. Springer: Berlin, 2000.
28. Klein G, Krief F. Mobile agents for dynamic SLA negotiation. In *International Workshop on Mobile Agents for Telecommunication Applications (MATA'2003)*, Marrakech, Morocco. Lecture Notes in Computer Science. Springer: Berlin, 2003.
29. Bellifemine F, Poggi A, Rimassa G. JADE-A FIPA-compliant agent framework. In *Proceedings of the Practical Applications of Intelligent Agents and Multi-Agents*, April 1999; 97–108.
30. Zeina JRAD. Apports des techniques de l'Intelligence Artificielle dans la négociation dynamique de la qualité de service: proposition d'un assistant à l'utilisateur dans les réseaux IP de nouvelle génération. Thesis, LIPN, University of Paris 13, France, May 2006.
31. Jrad Z, Benmammar B, Correa J, Krief F, Mbarek N. A user assistant for QoS negotiation in a dynamic environment using agent technology. In *Second IFIP International Conference on Wireless and Optical Communications Networks (WOCN 2005)*, Dubai, United Arab Emirates, March 2005.
32. Foundation for Intelligent Physical Agents. 2005. http://www.fipa.org [15 November 2007].

DOI: 10.1002/nem

33. AlAgha K, Dabbous W, Ghamri Doudanea Y, Jrad Z, Krief F, Le Garrec F, Masson S, Minet P, Nguyen TMT, Pujolle G, Rizo R, Serban R. *Final Report of Projet ARCADE*, January 2003.
34. Projet Arcade. 2002. Architecture de Control Adaptative des Environments IP. http://www-rp.lip6.fr/arcade/ [15 November 2007].
35. Projet IpSig. 2004. Signalisation générique du monde IP. http://www.telecom.gouv.fr/rnrt/rnrt/projets/res_02_85.htm [15 November 2007.
36. Boyle J, Cohen R, Durham D, Herzog S, Raja R, Sastry A. The COPS (Common Open Policy Service) Protocol. *RFC 2748*, January 2000.
37. Chan K, Seligson J, Durham D, Gai S, McCloghrie K, Herzog S, Reichmeyer F, Yavatkar R, Smith A. COPS usage for policy provisioning. *RFC 3084*, March 2001.
38. Nguyen TMT, Boukhatem N, El Mghazli Y, Charton N, Hamer L-L, Pujolle G. COPS-PR usage for SLS negotiation (COPS-SLS). draft nguyen-rap-cops-sls-03.txt. Internet draft, July 2002.
39. Mbarek N, Krief F. SLN NSLP: a service level negotiation signaling protocol in the NSIS environment. In *12th IEEE International Conference on Telecommunications (ICT'05)*, 3–6 May 2005.
40. Benmammar B, Mbarek N, Krief F. Signaling environment for service level negotiation and QoS management in Mobile IP networks. In *14th International Conference on Telecommunication Systems: Modeling and Analysis (ICTSM2006)*, 5–8 October 2006.

AUTHORS' BIOGRAPHIES

Badr Benmammar received the B.Sc. in Computer Engineering, with high honors, from the USTO (Université des Sciences et de la Technologie Oran), Algeria, in 1999. He received the M.Sc. in Computer Science from Paris 13 University, France, in 2002. He received the Ph.D degrees in Computer Science from the Bordeaux 1 University, France, in 2006. His main research activities concern quality of service on mobile and wireless networks, end-to-end signaling protocols and Agent technology. His work on quality of service has led to many publications in journals and conferences (Annals of telecommunications, WiMob, NetCon, MWCN, ICT, ICTSM).

Zeina Jrad is a post-doctoral fellow at the French national institute for research in computer science and control (INRIA). She worked before as teacher assistant for two years at the university of Paris 13 where she did her thesis entitled 'The use of Artificial Intelligence techniques in the negotiation of quality of service: a user interface for the new generation of Internet'. Her domains of interest concern user modelling and user interfaces for internet services.

Francine Krief is a Professor at Bordeaux 1 University, LaBRI laboratory, France. She is in charge of the research team 'Comet' (COntext Aware ManagEment and neTworking). Her research interests include Autonomic Management, Wireless Networks and Ambient Networks.

DOI: 10.1002/nem