

Artificial Intelligence techniques in the dynamic negotiation of QoS: A user interface for the Internet New Generation

Zeina Jrad¹, Francine Krief², Lahcene Dehni¹ and Younès Bennani¹

¹LIPN Laboratory, University of Paris13
99 Avenue J-B Clément, 93430 Villetaneuse, France
{zj, ldehni, younes.bennani}@lipn.univ-paris13.fr

²LaBRI Laboratory, UMR CNRS 5800
University of Bordeaux I, cours de la Libération, 33400 Talence, France
francine.krief@labri.fr

Abstract. The Internet New Generation provides a service adapted to the needs of the applications (particularly real time) including a Quality of Service (QoS) guaranteed. It is based on the DiffServ architecture and the policy-based networking management. It also uses the agent technology to solve problems and control infrastructures and flows. Access to the Internet New Generation is more difficult. The user has to choose the best service provider and indicate the technical parameters that he needs.

In this paper, we investigate the use of some techniques of the AI (Artificial Intelligence) domain to implement a user interface in order to help the user access to the Internet New Generation. We use the connectionist clustering in the management of the negotiation profiles. Then we use the agent technology to help the user to choose the best service provider, dynamically negotiate the SLS on the user's behalf, follow the users behavior to be able to anticipate the negotiations and manage the re-negotiations.

1. Introduction

In recent years, the development and the apparition 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's why we need services adapted to specific application needs with a guaranteed Quality of Service (QoS) [1]. The Internet New Generation has to provide these services particularly for real time applications.

To provide QoS in best-effort IP networks, the IETF developed the IntServ and DiffServ architectures. The IntServ model was created to transport audio, video, real time data as well as traditional data traffic in a way similar to the one of an integrated services network [2]. The DiffServ model was created because of the difficulties of deployment of IntServ [3],[4]. The Internet New Generation uses the DiffServ architecture to provide QoS.

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 [5] for the control and management of these IP networks. The Internet New Generation is based on the policy-based networking management.

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 [6].

There are some protocols that allow the dynamic negotiation [7], [8] and [9] 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 quality of service. The difficulty of the process can be reduced by replacing the user in finding applications needs in terms of QoS parameters according to the context of use.

In this context, the user needs to be assisted, first in choosing the best provider, and then in dynamically negotiating the technical parameters of the SLA (the SLS).

In this paper, we propose an intelligent user interface in order to help the user access to the Internet New Generation. We investigate the use of some techniques of the AI (Artificial Intelligence) domain to implement this interface. We use the connectionist clustering in the management of the user's profiles. Then we use the agent technology to help the user in the negotiation of SLS parameters according to the current profile.

The paper is organized as follows. Section 2 describes the framework of the proposed interface. The third section describes the first layer of the proposed model (the profile management layer). Sections 4 and 5 describe the control and negotiation layer respectively. Finally we present future work in section 6.

2. The proposed framework

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

- The analysis of the user's work for a profile attribution.

- The save and update of all data concerning the user preferences.
- The translation of the user's requests in SLS parameters.
- The negotiation of the desired QoS with a service provider.
- The monitoring of the real-time quality to compare it with the negotiated one.
- The substitution of the user in the decisions-making.

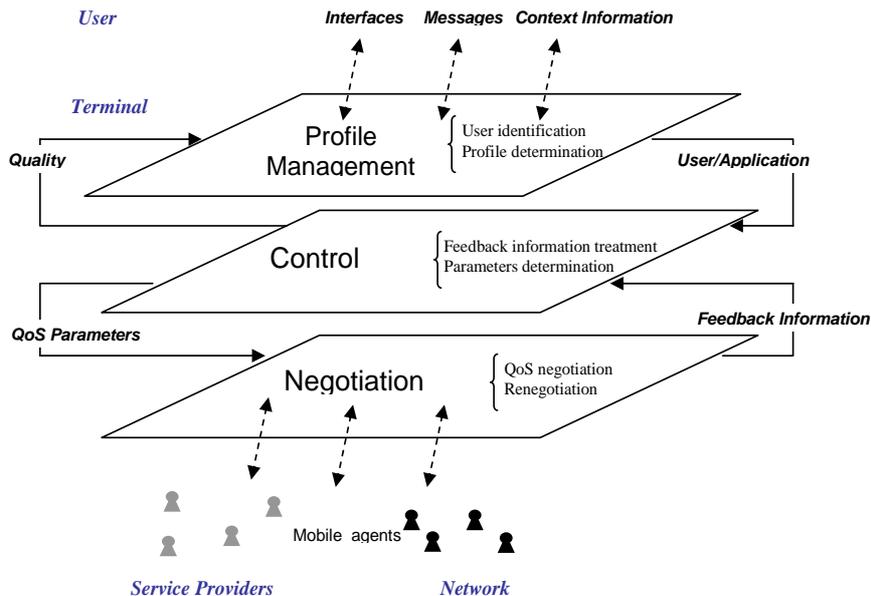


Fig. 1. The layers of the proposed framework

As shown in Fig. 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 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 the 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.

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 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 users and applications requirements.

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

3. Profile Management Layer

This layer is introduced to identify the user and to analyze 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 choices and actions. Applications can be classified in many categories according to their needs (delay, jitter...) and to the type of the supported information (data, voice, image...). 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 a communication with the user via graphical interfaces or messages.

Information collected through an observation of the user's behaviour.

Once the information is analyzed, 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.

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

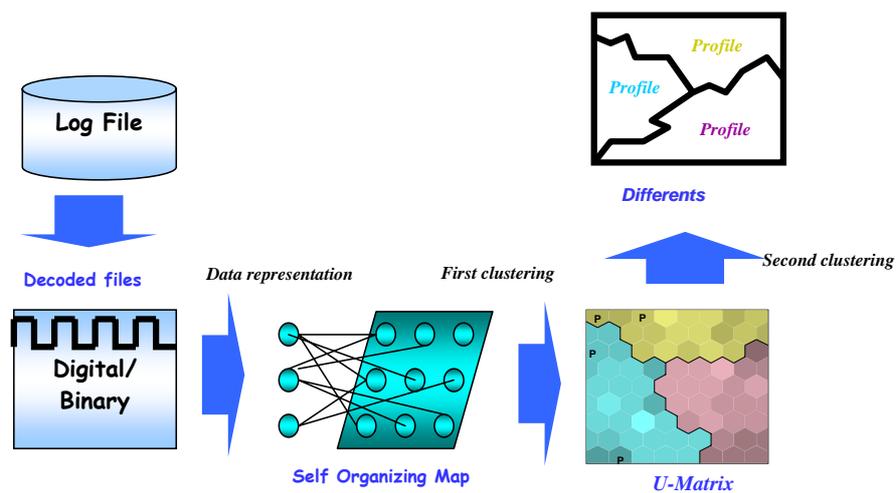


Fig. 2. Profile management procedure

Our approach, represented in Fig. 2, consists in recovering, first of all, data that represent traces of use (i.e. log files [10]). These data will be cleaned and re-coded in a numerical or binary format to be easily treated. We build then a Self Organizing Map from the re-coded file in order to extract profiles [10]. Finally, we carry out a classification to better see the clusters 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 Fig. 2 are detailed in the following subsections.

3.1. 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 training set. It aims to identify potential trend of data to be gathered into classes. This kind of learning approach, called clustering, seeks 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.

Self-Organizing Maps (SOM) implement a particular form of competitive artificial neural networks [11]; when an observation is recognized, activation of an output cell – competition layer – leads to inhibit activation of other neurons and reinforces itself. It is said that it follows the so called “Winner Takes All” rule. Actually, neurons are specialized in the recognition of one kind of observations. The learning is unsupervised because neither the classes nor their numbers are fixed a priori.

This type of neural networks is organized in a two dimensional layer of neurons [12]. Each neuron k is connected to n inputs through n exciter connections of respective weights w and to their neighbors 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 the neural networks. When the dimension of the input space is lower than three, both of the position of weights vectors and the direct neighborhood relations between cells can be visually represented. Thus, a visual inspection of the map provides qualitative information on its architecture.

The connectionist learning is often presented as a minimization of a risk function [13]. In our case, it will be carried out by the minimization of the distance between the input samples and the map prototypes (referents), weighted by a neighborhood function h_{ij} . To do that, we use a gradient algorithm. The criterion to be minimized is defined by:

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

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

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

$\lambda(t)$ is the temperature function modelling the neighborhood extent, defined as:

$$\lambda(t) = \lambda_i \left(\frac{\lambda_f}{\lambda_i} \right)^{\frac{t}{t_{max}}} \quad (3)$$

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

$$d_1(r,s) = |i - k| + |j - m| \quad (4)$$

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

- Initialization phase where random values are assigned to the connections weights (referents or prototypes) of each neuron of the map grid.
- Competition phase during which, for any input form $x^{(k)}$, a neuron $NN(x^k)$, with neighborhood, $V_{NN(x^{(k)})}$ is selected like a winner. This neuron has the nearest weight vector by using Euclidean distance:

$$NN(x^{(k)}) = \underset{1 \leq i \leq M}{\operatorname{argmin}} \|w_i - x^{(k)}\|^2 \quad (5)$$

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

If $w_j \in V_{NN(x^{(k)})}$ then adjust the weights using:

$$w_j(t+1) = w_j(t) - \varepsilon(t) h_{jNN(x^{(k)})} (w_j(t) - x^{(k)}) \quad (6)$$

else $w_j(t+1) = w_j(t) \quad (7)$

Repeat this adjustment until the SOM stabilization.

3.2. SOM map segmentation

We segment the SOM using the K-means method (Fig. 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 center of another class, it is moved into this new class. We calculate the centers of new classes and we reallocate the samples to the partitions. We repeat this procedure until having a stable partition.

The criterion to be minimized in this case is defined by:

$$E_{K-means} = \frac{1}{C} \sum_{k=1}^C \sum_{x \in Q_k} \|x - c_k\|^2 \quad (8)$$

Where C represents the number of clusters, Q_k is the cluster k , c_k is the center 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 Davies-Bouldin index [14], defined as follows:

$$I_{DB} = \frac{1}{C} \sum_{k=1}^C \max_{l \neq k} \left\{ \frac{S_c(Q_k) + S_c(Q_l)}{d_{cl}(Q_k, Q_l)} \right\} \quad (9)$$

With,

$$S_c(Q_k) = \frac{\sum_i \|x_i - c_k\|^2}{|Q_k|} \quad (10)$$

$$d_{cl}(Q_k, Q_l) = \|c_k - c_l\|^2 \quad (11)$$

C is the number of clusters, S_c is the intra-cluster dispersion, and d_{cl} is the distance (centroid linkage) between the clusters centers 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 [15]. Usually, they are based on the visual observations and the manual assignment of the map cells to the clusters. Several methods use the K-means algorithm with given ranges for K value. Our work is based on the approach of Davies-Bouldin index minimization.

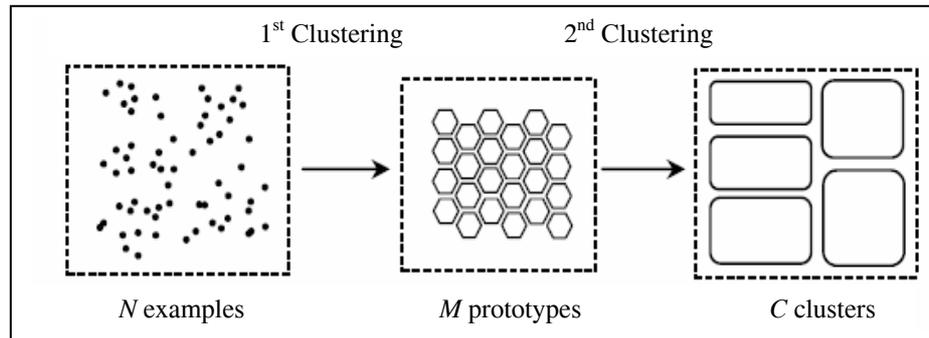


Fig. 3. Two successive clusterings: SOM followed by K-means

We note that the K-means approach can be directly applied to the data instead of 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 pre-treatment (dimension reduction, regrouping, visualization...). This pre-treatment has the advantage to reduce the clusters calculation complexity and also ensures a 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, to reduce the data space dimension, and on the other hand, to visualize the clusters in the plan.

3.3. Simulations results

We applied the two algorithms described above on 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 the researchers of the HUT (Helsinki University of Technology) of the T. Kohonen team [16]. The results obtained are very promising (Fig. 4).

Fig. 4.a. is a representation of a SOM map seen as “Component Planes” that allows the visualization of the partition of the 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.

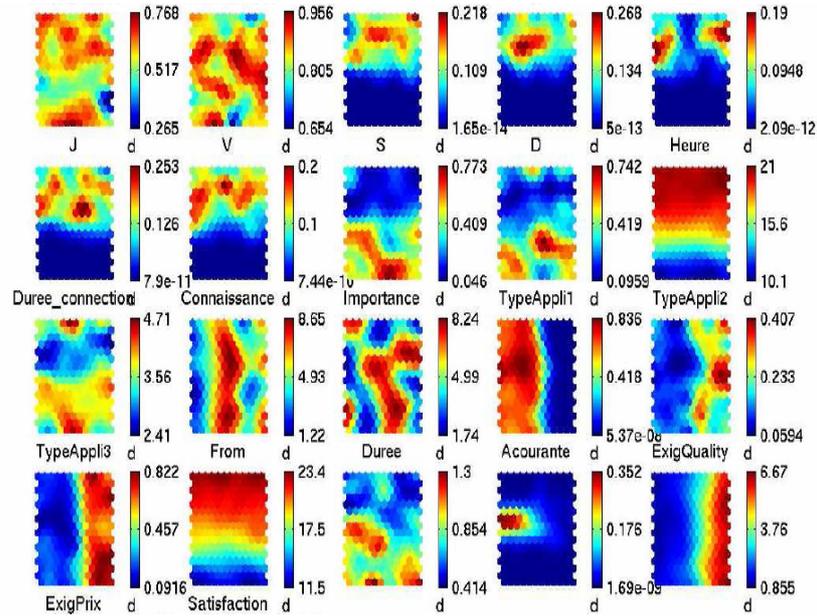


Fig. 4. a. SOM clustering

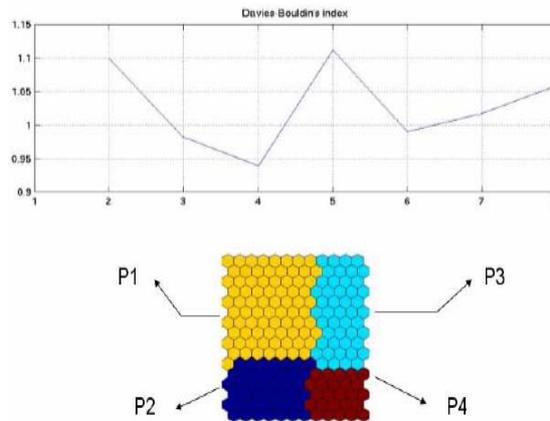


Fig. 4.b. Resulting profiles

Fig. 4.b. represents the neurons segmentation of SOM map (second clustering). The curve shows that the minimal value of the index of Davies-Boudin corresponds to an optimal clustering resulting in four profiles. The various colors represent the different clusters or the identified negotiation profiles.

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

4. 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 [17].

The control made in this layer should also 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 QoS parameters values sent to the negotiation layer. These values are related to the negotiation profile determined in the previous layer and represent the users 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, the service providers and the profile of negotiation.

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 the analysis of his behavior.

It analyses the feedback information collected by the responsive agents in order to take decisions concerning the re-negotiation and to evaluate the state of the link and the video packet transmission.

5. Negotiation Layer

The Negotiation layer is responsible for the negotiation of the QoS parameters received from the control layer. It sends the QoS parameters to the network entities and starts the process of negotiation. It may also ask for a change in the required services according to the needs of the user and the applications. This corresponds to a request of re-negotiation between the two parts. Mobile agents, [18] and [19], are sent to the service providers in order to bring new offers. Three types of agents are proposed [20]:

1) User Negotiator (UN): This is a mobile agent that is sent by the User Overseer (UO) 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.

2) Access Negotiator (AN): It is created by the provider to each mobile agent (UN) that arrives on its platform in order to negotiate services in favor of the provider.

3) User Overseer (UO): It manages the entire negotiation process on behalf of the user. It sends UN for the service survey and negotiation. It collects the results of the different negotiation threads and then makes the final decision.

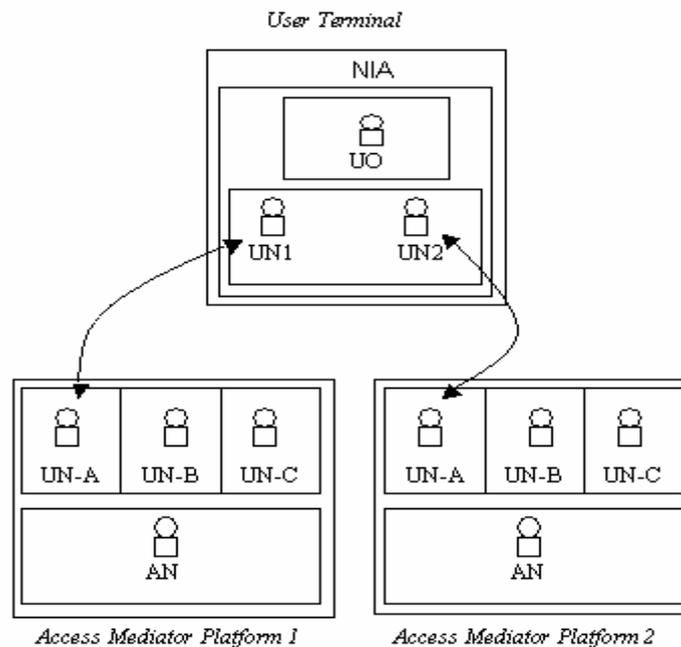


Fig. 5. Framework for the dynamic negotiation of SLA/SLS

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.

5.2. 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 customer's UN agents. The User Overseer (UO) sends a User Negotiator (UN) to the provider's platform with which he has a predefined contract. The UN takes the user profile which corresponds to the QoS characteristics of the applications the user often use. The access to the platform requires an authentication phase. UN then starts to survey the offers made by the provider. When new offers are proposed, UN filters them and sends them to his UO, the one 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 concerned UNs. Then the UNs open negotiations with their AN counterparts to obtain the best rates and quality of services. When the UN and the AN reach an agreement (or after a certain period of time where no agreement is done), the UN sends to the Overseer the description of the best proposal made by the AN. Then the Overseer compares the different offers it received from the different ANs and sends back an agreement to the best one. Then every agent concerned by the different concurrent negotiations returns to its normal state.

To implement our model and protocol of negotiation, we used the platform JADE (Java Agent Development Framework). JADE [21] is a software Framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the 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 not even need to share the same OS) and the configuration can be controlled via a remote GUI. More details concerning the implementation of our model and the protocol of negotiation can be found in [22] and [23].

6. Conclusion and future work

The Negotiation Individual Assistant (NIA) presented in this paper constitutes an interface between the user and the network in the context of the Internet of new generation. This interface integrates two interesting techniques of the Intelligence Artificial domain: The connectionist learning and the agent technology.

A clustering algorithm, based on the topological Self Organizing Maps (SOMs), is used (in the first layer of the NIA) to determine a negotiation profile that represents the user preferences and the 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, the agent technology is used (in the third layer of the NIA) in order to help the user to choose the best service provider, dynamically negotiate the SLS on the user's behalf, follow the user's behavior to be able to anticipate the negotiations and manage the re-negotiations. The Introduction of a multi-agents system, in both of the user and providers sides, has shown a good performance in the choice of the best service provider. The agents of our system communicate via a FIPA protocol [24]. This approach has many advantages. The terminal charge is reduced, the system can function on a large range of terminals and the service providers can more easily propose new services.

Our NIA has been implemented and tested in two French national research projects: ARCADE [25] [26] and IPSIG [27].

Future work may concern different topics:

- The introduction of security levels beyond the selection criteria of the best service;
- The introduction of the mobility of the user;

- The application of the future mobile terminals. Some terminals will also be able to use different access technologies either simultaneously or one at a time. Therefore the most important property of any communication system is its ability to handle mixtures of flows and traffic characteristics in a reasonable way [28];

References

1. QoSforum, "QoS protocols & architectures", White Paper, July 1999.
2. Braden R., Clark D. and S. Shenker, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, June 1994.
3. Blake S., D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.
4. Nichols K., S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field in the IPv4 and IPv6 Headers", RFC 2474, December 1998.
5. Yavatkar R., D. Pendarakis, R. Guerin, "A Framework for Policy Based Admission Control", RFC 2753, January 2000.
6. D. Goderis and al., Service Level Specification Semantics and Parameters, draft-tequila-sls-02.txt, Internet draft, January, 2002.
7. J. Boyle, R. Cohen, D. Durham, S. Herzog, R. Raja, A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000.
8. K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, A. Smith, "COPS Usage for Policy Provisioning", March 2001.
9. T.M.T. Nguyen, N. Boukhatem, Y. El Mghazli, N. Charton, Louis-Louis Hamer, G. Pujolle, "COPS-PR usage for SLS negotiation (COPS-SLS)", draft nguyen-rap-cops-sls-03.txt, Internet Draft, July 2002.
10. K. Benabdeslem, Approches Connexionnistes pour la visualisation et la classification des données issues d'usages de l'Internet, thesis in computer science, LIPN, University of Paris13, France, December, 2003.
11. T. Kohonen and S. Kaski and H. Lappalainen, Self-Organized Formation of Various Invariant-Feature Filters in the Adaptive-Subspace SOM, pages 1321-1344, Neural Computation, 1997.
12. F. Zeharoui and Y. Bennani, M-SOM-ART : Growing Self Organizing Map for sequence clustering and classification, European Conference on Artificial Intelligence (ECAI), Valencia, Spain, August, 2004
13. Y. Bennani, Réseaux de neurones artificiels, Chapter in « Encyclopedie d'Informatique et Science de l'information », Edition Vuibert, 2005.
14. Davies, D., L., Bouldin, D., W., : A Cluster Separation Measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1(2): pp. 224-227. (1979).
15. Juha, A. , Esa, A. : Clustering of the Self-Organizing Map. IEEE Tractions On Neural Networks volume 11, n° 3, (2000).
16. E. Alhoniemi and J. Himberg and J. Parhakangas and J. Vesanto, SOM Toolbox, 2000, Copyright (C), <http://www.cis.hut.fi/projects/somtoolbox/>.
17. Z. Jrad. And F. Krief. An Intelligent Interface for the Dynamic Negotiation of QoS in ARCADE. ACS/IEEE International Conference on Pervasive Services. ICPS'2004. Beirut, Liban. Juillet 2004.
18. J.-P. Briot, Y. Demazeau, "principles and architecture of MAS", 2002.
19. N. Agoulmine, M. Fonseca, A. Marshall, "Multi-domain Policy Based Management Using Mobile Agents, Lecture Notes in Computer Science, 2000.

20. G. Klein and F. Krief. Mobile Agents for Dynamic SLA Negotiation. International Workshop on Mobile Agents for Telecommunication Applications. MATA'2003. Lecture Notes on Computer Science, Springer. Marrakech, Maroc. October 2003.
21. JADE-A FIPA-compliant agent framework, F. Bellifemine and A. Poggi and G. Rimassa, pages 97-108, Proceedings of the Practical Applications of Intelligent Agents and Multi-Agents, April, 1999.
22. Zeina JRAD, Thesis report, 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, LIPN, University of Paris 13, France, May 2006.
23. Z. Jrad., B. Benmamar, J. Correa, F. Krief , N. Mbarek. A user assistant for QoS negotiation in a dynamic environment using agent technology. Second IFIP International Conference on Wireless and Optical Communications Networks WOCN 2005. Dubai, United Arab Emirates UAE, March 2005.
24. <http://www.fipa.org>, 2005
25. K. AlAgha and W. Dabbous and Y. Ghamri Doudanea and Z. Jrad and F. Krief and F. Le Garrec and S. Masson and P. Minet and T.M.T. Nguyen and G. Pujolle and R. Rizo and R. Serban, Final report of Projet ARCADE, Janvier,2003.
26. Projet Arcade, Architecture de Control Adaptative des Environnements IP, Web Site: <http://www-rp.lip6.fr/arcade/>, 2002.
27. Projet IpSig, Signalisation générique du monde IP, http://www.telecom.gouv.fr/rnrt/rnrt/projets/res{_}02{_}85.htm, 2004.
28. Middleware for Mobile Applications Beyond 3G, Kimmo Raatikainen, SmartNet 2002.