Decentralized task allocation mechanism applied to QoS routing in home network

Emna Ghedira Molinier, Lionel Molinier, Guy Pujolle

Decentralized Task Allocation Mechanism Applied to QoS Routing in Home Network

Emna Ghedira, Lionel Molinier, and Guy Pujolle

Paris Universitas - LIP6, Paris, France,
{emna.ghedira,lionel.molinier,guy.pujolle}@lip6.fr

Abstract. We present in this paper a task allocation mechanism constrained by a distributed, complex environment namely in our case a home network. Thus, added to multi agent systems constraints, we have restrictions in terms of communication (available throughput) and resource (bandwidth). We have a multi agent system with a knowledge base, and we have to ensure a proper quality of service in home Network adjusting the routing in real time by applying alternative route to the main ones. Our approach uses the *first price sealed bid* type auction: when an agent does not have an alternative route, it launches an auction. The agent offering the best price will be the next hop of the route.

Keywords: Home Network, Task allocation, QoS, Knowledge based system, Ontology.

1 Introduction

With the success of the Internet, its expansion in home, the adaptability, flexibility and decentralization of its application, limits had appeared. Those problems affect both users and operators. Because they are neophyte, users are unable to react properly to any event occuring in their home network. This is a direct consequence of high level control lacks. Operators are faced with the inability to properly transcribe business policies to their equipment configuration.

The task allocation problematic is a central issue in software agent systems, distributed systems but also in various areas [11, 13, 7]. This is a direct consequence of agent's hetereogenity of the multi agent system. Each agent has its own abilities, and a partial view of its environment. The multi agent systems can be assimilated to a company, where an employee does not necessarily know all its coworkers, but all the employees with their own abilities and behaviors converge to a common goal, maximizing corporation's profit [14, 13, 7].

In this paper, we propose a task allocation mechanism to adjust the routing and also the quality of service in a home network. In this environment, we have constraints in terms of communication as we are limited by bandwidths, in terms of resource which is based on the available throughput and finally because there is only a partial visibility [12].

This paper is organized as follow: we start with an overview of the problem then we are introducing the agent we are using to solve our problematic and one application. Finally we conclude and present some future works.

2 Problem overview

We call *Home Network* a set of elements that compose our high-tech environment at home with a home gateway (box) and others home devices like phones, television, PC... This type of network can not be studied separately of the Internet thus, the broadband access is the first aspect to be considered [10, 9].

No longer than 15 years ago, Internet access at home was reserved to some elite. They were using an analogical modem (56K) connected to their desktop PC, and they were painfully surfing the Internet and reading their e-mails. However, with broadband accesses, the Internet starts to widespread, based on this modem-PC architecture.

This was the first step of the *Home Network*, since triple-play offers introduce the notion of router inside the home. Nowadays there is a home gateway in mostly every home, which provides dedicated interfaces for Internet access, telephony and television. This gateway is providing interfaces for those three services: SCART or HDMI, Ethernet or WiFi, analogical telephony.

In the near future, all those services will merge IP, thus creating only one into network over the Home. The single router architecture will no longer be sufficient because it will have to provide at least 10 Mb/s in each corner of the house (HDTV flow for instance). That is why, standardisation consortiums tend to agree on the architecture illustrated in figure 1, which add several devices named HNID (Home Network Infrastructure Device) in order to improve the coverage (see [6]). Those devices also act as bridges between technologies.

Introducing HNID, the network will cover all the home with good conditions, but also enable the network to suppport more devices. However, creating a real network, with active elements, leverages routing problematic: how can we provide routing in the *Home Network*? The tricky point is that the medium (WiFi) is very perturbation sensitive and its bandwidth may collapse very quickly.

This routing problematic is quite well known in networks. *Home Networks* inherit from ad-hoc networks, but this domain only considers wireless links, excluding PLC[1] benefits. At the opposite, it also inherits from corporate networks which already have such kind of architecture, but in Home Network, the user is neophyte and there is no human network administrator to configure and maintain it. In other words, the *Home Network* has to be autonomic, which means that it has to optimize itself in order to provide the best service possible to the end-user.

[1] PLC stands for PowerLine Communications. In other words, it means that we use power outlet as a network medium

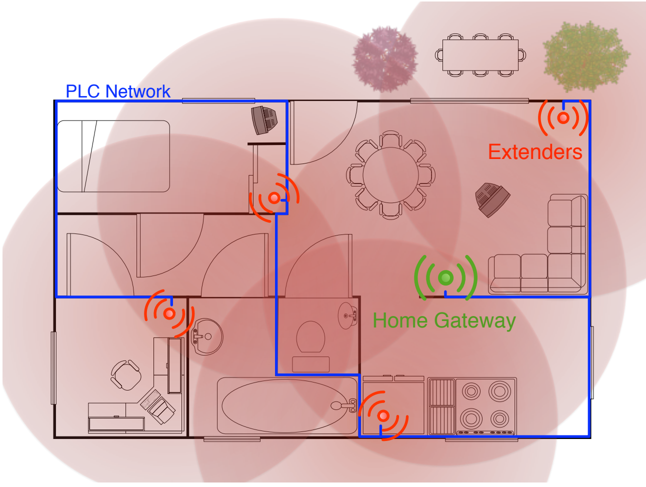


Fig. 1. Architecture of the Home Network

3 Agent-based solution

Introducing autonomy in networks is an emerging and widely recognized idea in the telecomunication world. One way to autonomy is to work with a lot of knowledge (see 3.1) within the network but also to take high-level decisions as the network complexity is increasing.

3.1 Knowledge plane

The knowledge plane has been added upon applications to mutualize information. It has been introduced by [3], a network reseacher, and defined as:

> [. . .] a distributed and decentralized construct within the network that gathers, aggregates, and manages information about network behavior and operation.

We grasp this concept through a multi-agent system based on knowledge.

3.2 Ginkgo Multi agent system

What is the Ginkgo Platform? The agent platform can be considered as a middle-ware for Autonomic Networking. It was designed to run onto network equipments: that means the platform is distributed over the network. The platform architecture is presented on figure 2.

Each agent is embedded on a network device and has a partial view of its environment which is defined by the application designer. It communicates only exchanging knowledge with neighbours.

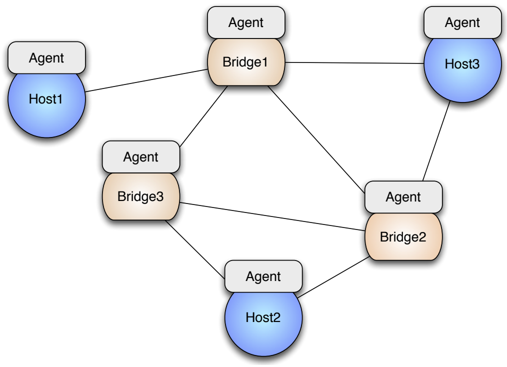


Fig. 2. An agent is embedded each network device

What is an agent in the Ginkgo Platform? The figure 3 presents the structure of ginkgo agent, thereafter the explanation of each component.

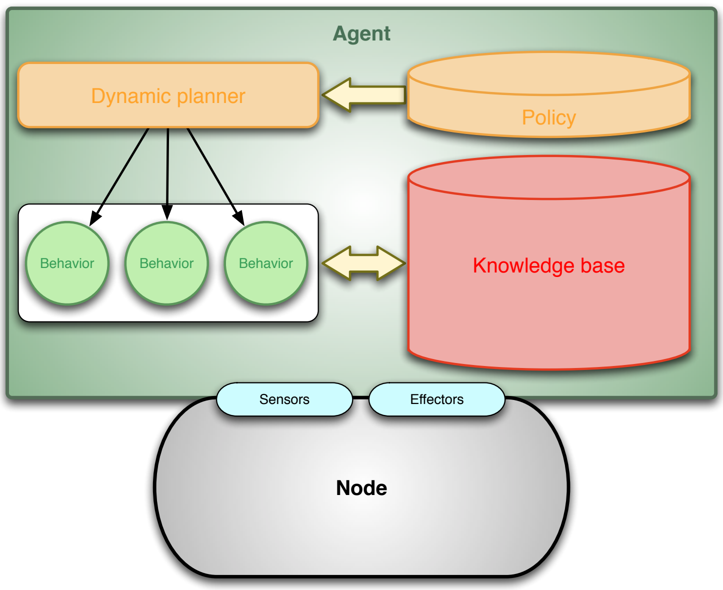


Fig. 3. Ginkgo agent architecture

- **Behavior**: Agent abilities are implemented by the application developer and permanently adapting themselves to environment changes. It can read and write in the knowledge base, sense and act from the node.
- **Policy**: Rule set by operator.
- **Dynamic Planner**: It orchestrates the behaviors. It can start, stop, configure even modify according to the policy.
- **Knowledge Base**: The Knowledge base (KB) is a central functionality of the agent. It stores the data used by an agent in an homogeneous and structured way and provides diffusion mechanisms of this knowledge between the agents.
- **Situated View**:It is the partial view that an agent has of its environment [2]. It gathers local knowledge acquired by examining the device on which is embedded but also the knowledge collected by its peers.

Preliminary work: Ontology, a way to structure the KB Classicaly, in a network device, there are many of algorithms running simultaneously, each one using its own data. This lead to an important overlap, mandatory, since there is no common way to handle data. KB stores not only information manipulated by the behavior and the dynamic planner but also supplies a representation.

Thus, it stands to reason to define an ontology which allows us to have a common vocabulary for the knowledge representation. This facilitates the communication between agents. The figure 4 represents a subpart of the ontology using Unified Modeling Language (UML).

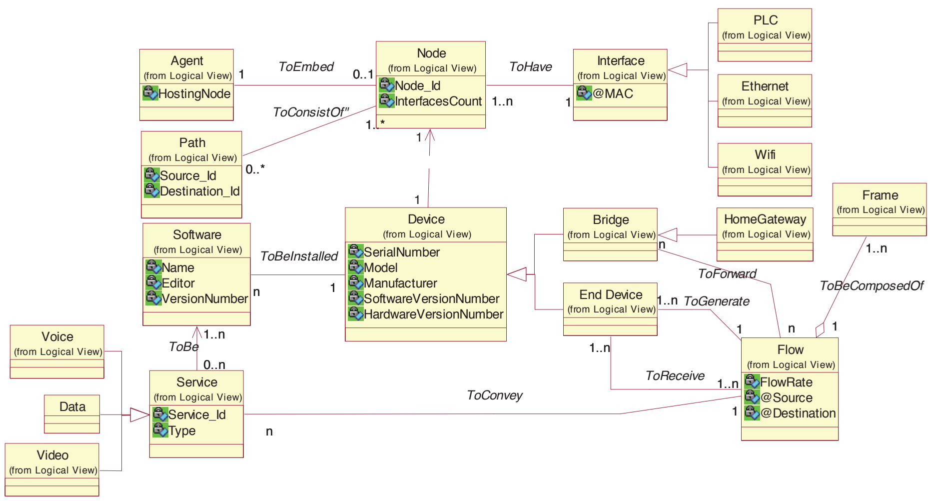


Fig. 4. Extract of the ontology

From this ontology we have derivated a data model and implemented it in the KB. For instance, an agent is *embedded* into a node which *has* interfaces connected to other nodes ...

Now, the KB is structured, we have to design the algorithm that relies on this knowledge.

4 Task Allocation in Home Network

4.1 Motivation

In classical approach of routing concerns, we try to tune algorithms to have the lowest convergence time. This can be done by reducing the amount of information taken into account. However, in such a complex environment this may lead to poor routing performances as links quality may fluctuate very quickly.

Our way to handle this problem is to consider that the routing algorithm is performing well in most cases, but need to be by-passed in critical situations, by applying pre-computed alternative routes. This two stages routing assure the quickest response time in case of link failures or link capacity collapse due to external interferences.

Even if the alternative route principle is well-known in the litterature, it is hard to implement, since we first need to monitor the normal operation of the network, and secondly compute (and maintain) in background alternative routes.

When a flow comes to a node, the classical routing process routes it. However, the agent embedded detects this new flow and tries to find out an alternative route. This flow has some specific properties such as type, bandwidth and destination. This make the finding complex (several criteria) and relying on neighbour abilities.

That is why we have chosen to use a task allocation mechanism based on auctions. So, the neighbours can help a node to find out an alternative route offering their ressources. The auction issuer can choose the best offer.

4.2 Formalization

The task allocation problematic is represented by a *triplet*:

$$< T, Ag, c >$$

T : set of tasks

$Ag = 1, \ldots, n$: All agents participating in the execution of these tasks

$C : P(T) \longrightarrow \mathbb{R}^{+}$: Function defines the execution cost of each task subset and have two constraints:

- Monotony : If we add a task to a set of tasks, the cost of their execution must be superior to the execution cost of the initially set.

$$c(T_1, .., T_n) < c(T_1, \ldots, T_n, T_{n+1})$$

- If a task is not executed the cost must be null.

4.3 Environment

We consider a set of agents $Ag = 1, \ldots, n$ organized according to a network topology as represented in the figure 5. Each agent must handle requests placed in a schedule, knowing that any agent can be the recipient of a query. There are T types of tasks: $T = 1, \ldots, n$.

Agents of the network do not have the same skills, they can not perform all types of tasks. In addition, for a given type of task, they are able to execute only x number of tasks Ti (ie corresponds to the structure that is under the Agent d for instance). The figure 5 illustrates the problem.

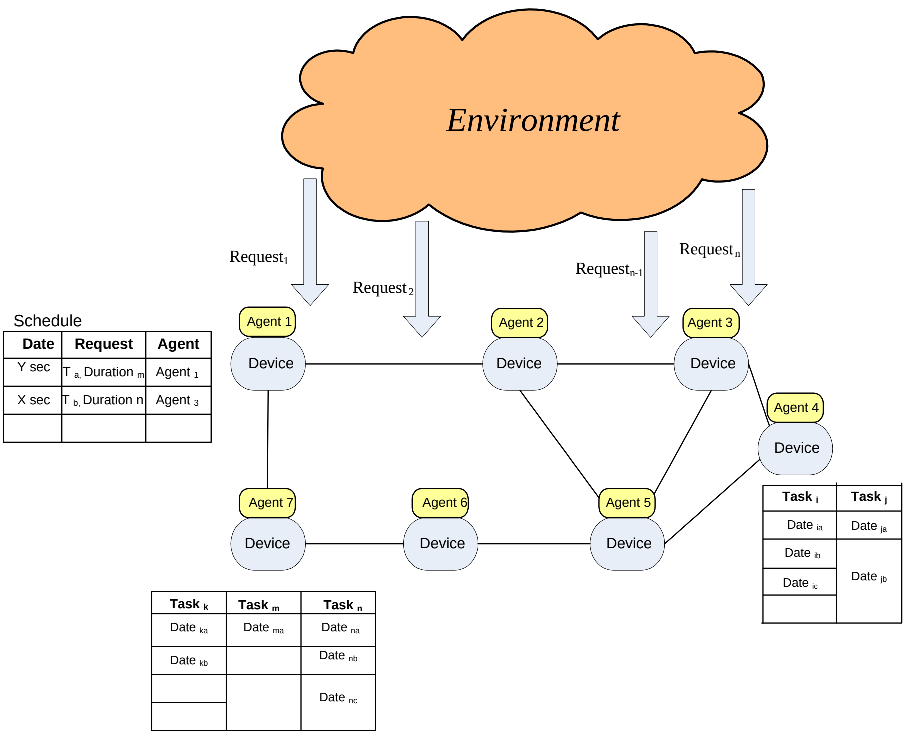


Fig. 5. Illustration of the problem

Given these constraints, there are two reasons for which a reallocation is necessary:

- The task type of the request received is not within the skill of the receiving agent.
- The agent has reached the maximum number of tasks it can execute.

4.4 Approach and general principles

The proposed mechanism is based on the theory auction mechanism and inspired by auction *first-price sealed-bid* presented below:

> The initiator starts the auction and each participant submits a bid in one turn, without knowing other participants offer. The one which has made the largest offer wins the subject of the auction and pays the amount its offer [5].

Before we define the general principles, we present the structures that compose the situated view.

- $\{T_{ex}\}$: Queue of task to execute

$Task_i$	$Task_j$
$Task_i$, $Date_a$	$Task_j$, $Date_a$
$Task_i$, $Date_b$	...
...	...
...	

- $\{E_{local}\}$: List of local auctions. This structure brings together all auctions initiated by the agent.

Id_{task}	Reservation Price
$Id(T_i)$	X_i
$Id(T_j)$	X_j
...	...

- $\{E_{global}\}$: List of global published auctions to the network

Editor agent	Id_{task}	Reservation price	Assigned agent	Offers
Ag_1	$Id(T_i)$	X_i		(Ag_4,P_i) ...
...	...	...	...	...

The offer column contains only its own offer except for the editor agent that contains all offers.

- $\{O\}$: List of the offers for a bid published

IdE_{global}	Editor agent	Offer price
$Id(Eglobal_i)$	Ag_1	X_m
$Id(Eglobal_j)$	Ag_4	X_k
...	...	...

These structures are used in algorithms presented below. They are implemented as behaviors and are based on these points:

- A task becomes a bid only when it can not be executed by the receiving agent
- The situated view is only partially broadcasted
- The situated view is transmitted only when there is a change

When an agent receives a task, it checks two conditions:

1. **Condition1:** If the type of task belongs to its skills
2. **Condition2:** If the maximum number of execution for this type of task is reached.

If **condition 1** is false or **condition 2** is true, then the agent initiate an auction about this task following the algorithm described in figure 6.

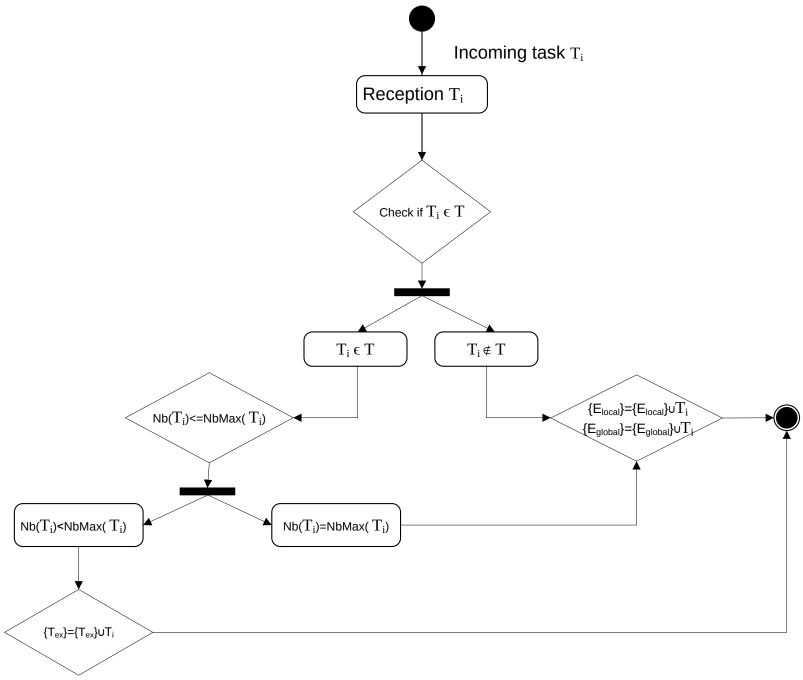


Fig. 6. Behavior that treats a received task

Bidding management by an agent: When an agent receives the structure $\{E_{global}\}$, it processes the bids one by one executing these steps:

▷ **Check** if the auction has already been treated
- **If** true
  Go to next auction.
- **Else**
  ▷ **Check** if it belongs to $\{E_{local}\}$ (it means that I am the editor agent)
  - **If** true
    1- It collects all offers and select the best one

2- It deletes the auction from $\{E_{local}\}$
3- Fill informations of the auction in $\{E_{global}\}$ with the assigned agent
4- Go to next auction.

- **Else**
  - ▷ **Check** if an agent has been assigned
    - **If** true and the assigned agent is myself
      1- Add task to $\{T_{ex}\}$ and execute it
      2- Go to next auction.
    - **If** true and the agent assigned is not myself
      Go to next auction.
    - **If** false
      - ▷ **Check** if I have already made an offer
        - **If** true
          Go to next auction.
        - **Else**
          - ▷ **Check** Condition1
            - **If** true
              - ▷ **Check** Condition2
                - **If** true
                  1- $\{O\} = \{O\} \cup \{O\}_{Agi}$ with $\{O\}_{Agi} = null$
                  2- Go to next auction.
                - **Else**
                  1- The determinated offer amount=$\{O\}_{Agi}$
                  2- $\{O\} = \{O\} \cup \{O\}_{Agi}$ with $\{O\}_{Agi} \neq null$
                  3- Go to next auction.
            - **Else**
              1- $\{O\} = \{O\} \cup \{O\}_{Agi}$ with $\{O\}_{Agi} = null$
              2- Go to next auction.

The initiator agent waits to collect all offers before selecting the best one. Indeed, all neighbours must make an offer, eventually a null one if it can not execute the task.

We suppose that for each task type there is at least two agents can execute it. So, if an agent can not execute a received task and initiate an auction, we assure that there is an other agent can makes an offer.

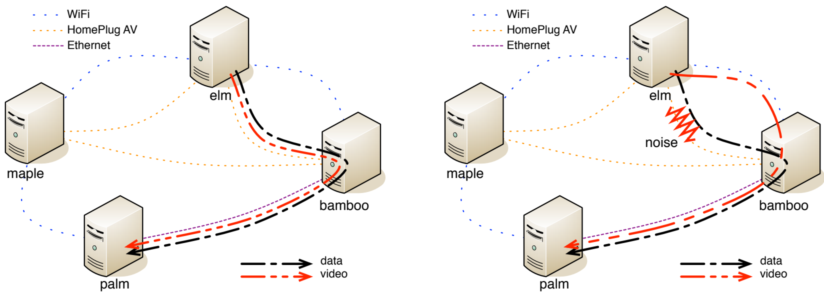
Moreover, the offers amount is determinated using an algorithm defined by the application designer based on its constraints (here the available throughput).

An agent deletes an auction of its structure $\{E_{local}\}$ and the referenced offers when the auction is completed (this means that an agent was assigned and it added the task to its structure $\{T_{ex}\}$). As for the structure $\{E_{global}\}$, an auction will be deleted by the assigned agent after having added it to its $\{T_{ex}\}$.

5 Application

Previous algorithms have been implemented and tested on the Ginkgo Platform using a set of unitary tests. Futhermore, the mechanism has been used by some members of the Ginkgo research team working on Home Networks. In this project, it was used to improve alternative routes computation as explained in [8].

In this context, a testbed has been implemented which support the scenario described with figures 7(a) and 7(b). HNID are connected using Ethernet, WiFi and PLC, and the network convey data, tv and voice flows.

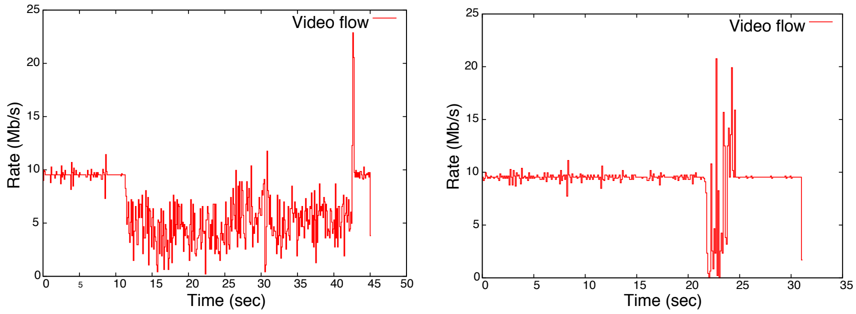


(a) Initial state (b) Network action

At the beginning (figure 7(a)), there are, one data flow over the PLC, and one video flow going from *elm* to *palm* over the same medium. In the next step, we generate a perturbation of the PLC link[2]. The agent detects it and applies the alternative route which uses the WiFi link: this change is immediately applied and the user does not suffer from any scrambling. However, the data flow is not redirected since it allows easily some bandwidth reduction. The figure 7 illustrates the situation for the video flow with and without our agents.

In the figure 7(c), without agents, we can see, during the first 10 seconds, that the received rate is almost 9 Mb/s. At $t = 11s$, we start the perturbation, and we can see that the rate is down by 50 % : the quality perceived by the user is very poor. On the contrary, the same scenario is done with agent (figure 7(d)) : the rate is unstable during $3s$, after that the user can enjoy a good quality. Those $3s$ can be considered as important, but this is only an implementation concern, since agents are computing proactively alternative route (not adding delay) and the perturbation detection can be done in less than a second.

[2] Switch-on a lamp for instance



(c) without our solution (d) with our solution

Fig. 7. Bandwidth evolution

6 Conclusion and future works

The use of this auction based mechanism has provided some ease in its application, because the developer should in no case search a solution to its problems namely the computation of alternative route, but just determine task, and what represent a price. This prove that this proposed solution is generic and can be applied to many network problems.

As future works, we would like to investigate other applications like HandOver or, the virtualization [4, 1], even GMPLS.

The use of distributed artificial intelligence allows to overcome the limits of networks solutions. In other words, effectively manage the complexity of growing networks.

As an immediate enhancement, we will do a statistical validation studying extreme cases, for example, additional requests to determinate the limit of this mecanism and improve the algorithm or propose another one complementary.

References

1. Abid M., Ligocki M., Molinier L., Nguenguang G., Pujolle G., Gaiti D., Zimmermann H.: Practical handover optimization solution. In: IFIP Wireless Days. Dubai, UAE (2008)
2. Bullot T., Khatoun R., Hugues L., Gaïti D., Merghem-Boulahia L.: A situatedness-based knowledge plane for autonomic networking. Int. J. Netw. Manag. **18**(2), 171–193 (2008). DOI http://dx.doi.org/10.1002/nem.679
3. Clark D., Partridge C., Ramming C.J., Wroclawski J.T.: A knowledge plane for the internet. In: SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 3–10. ACM Press, New York, NY, USA (2003). DOI 10.1145/863955.863957. URL `http://portal.acm.org/citation.cfm?id=863957`
4. Fejjari I., Pujolle G.: An autonomic system for virtual network piloting. In: WNetVirt First Workshop on Network Virtualization and Intelligence for the Future Internet (2010)

5. Florea A.M.: Bucharest university online course. URL http://turing.cs.pub.ro/auf2/html/chapters/chapter5/chapter_5_4_1.html
6. Freiderikos V., Gaïti D., Hamon M.H., Insler R., Jafré P., Kortebi A., Meyer S., Molinier L., Pujolle G., Zimmermann H.: Piloting home network with intelligent agents. In: Wireless World Research Forum 21 (2008)
7. Ketchpel S.: Forming coalitions in the face of uncertain rewards. In: AAAI '94: Proceedings of the twelfth national conference on Artificial intelligence (vol. 1), pp. 414–419. American Association for Artificial Intelligence, Menlo Park, CA, USA (1994)
8. Molinier L., Ghedira E., Ligocki M., Francois R., Freiderikos V., Kortebi A.: Autonomic qos management and supervision system for home networks. In: Wireless Days (WD), 2009 2nd IFIP, pp. 1 –6 (2009). DOI 10.1109/WD.2009.5449648
9. Molinier L., Ligocki M., Pujolle G., Gaiti D.: Piloting the spanning tree protocol in home networks using a multi-agent system. In: Wireless Days, 2008. WD '08. 1st IFIP, pp. 1 –5 (2008). DOI 10.1109/WD.2008.4812910
10. Nguengang G., Molinier L., Boite J., Gaïti D., Pujolle G.: Intelligent routing scheme in home networks. In: Springer (ed.) Home Networking, pp. 179–196 (2008)
11. Nwana H.S., Lee L., Jennings N.R.: Co-ordination in software agent systems (1996)
12. Rahwan I.: Interest-based negotiation in multi-agent systems. Tech. rep. (2004)
13. Walsh W.E., Wellman M.P.: A market protocol for decentralized task allocation. In: In The Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98, pp. 325–332 (1998)
14. Wooldridge M., Jennings N.R.: Intelligent agents: Theory and practice. Knowledge Engineering Review **10**(2), 115–152 (1995). URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.2702