On Using Multi-agent Systems in End to End Adaptive Monitoring

Leila Merghem [1], Dominique Gaiti [1,2], and Guy Pujolle [1]

[1] LIP6-Université de Paris 6, 8 rue du Capitaine Scott – 75015 Paris, France,
{Leila.Merghem,Guy.Pujolle}@lip6.fr
[2] LM2S: Université de Technologie de Troyes, 12 rue Marie Curie – 10010 Troyes Cedex, France.
Dominique.Gaiti@utt.fr

Abstract. The complexity (in terms of services and multimedia streaming) and dynamicity of telecommunication networks are continually growing, making network management and control more and more difficult. Such a management must be adaptive, dynamic and smart. In order to be sure of getting the best management mechanisms' choice, a monitoring operation becomes fundamental. This paper presents a new adaptive monitoring approach based on a multi-agent system, taking into consideration the features of the networks namely flexibility, dynamicity, heterogeneity of users generated traffic, etc. A two-layers monitoring architecture is proposed in this paper. The first layer is responsible for dealing with monitoring of some local parameters, which are useful for the second layer. Indeed, this latter is adaptive, autonomous and able to make new decisions- about the data to monitor and the management mechanisms to activate- based on the current node's state.

1 Introduction

Telecommunication networks represent a very dynamic and complex area, in which every day managers are facing new problems and challenges. Increases in network complexity and information volume make resources and network control more and more difficult. Users ask for more and more services and multi media applications where the network is not conceived to carry these data. We argue that in such an unpredictable, changing and open environment, intelligent agents give the opportunity to obtain an optimized network management and monitoring. In fact, the main features of agents, namely the autonomy, the ability to communicate with the others, the ability to solve common problems in a decentralized manner, in addition to learning aptitudes, allow agents to operate in telecommunication network dynamic environment.

In this paper, we will present a new adaptive monitoring approach of telecommunication networks based on agents which monitor the network state and control its different components. Our aim is to include some intelligent and dynamic control, thanks to agents, allowing us to guarantee a QoS and to give a better management and global performance of the network.

A. Marshall and N. Agoulmine: MMNS 2003, LNCS 2839, pp. 422–435, 2003.

The remainder of this paper is organized as follows. We first present the features of the monitoring approach we propose. Then, we describe Multi-Agent Systems and explain why we have chosen this technique. Section 4 introduces the two monitoring levels within a network's node, followed by a detailed description of the different kinds of events and actions in the model. In section 6, we give examples of adaptive monitoring. Section 7 presents related work and section 8 concludes the paper.

2 Usefulness of Adaptive Monitoring

The monitoring we perform is:

- scalable: our monitoring approach is scalable because it is based on a multi-agent system which scales well with the growing size of the monitored network. For that, one has to integrate an agent (or a group of agents) on the new node to be controlled and the monitoring of this node is realized;
- distributed: each agent is responsible for a local monitoring. There is no centralization of the information collected by the different agents, and the decisions the agent performs are in no way based on global parameters. This feature is very important as it avoids having bottlenecks around a central monitoring entity;
- adaptive: the agent adapts its actions depending on the monitored data and entities according to the incoming events and the vision of the current system state. The monitoring we adopt is adaptive because of the following: (1) the agent modifies the monitored parameters: the agent decides, at every moment, which parameters must be monitored and which ones are no longer important under the current conditions; (2) the agent adapts the current management mechanisms (MM) and the actions undertaken when a certain event occurs. The actions the monitoring process executes may become no longer valid and must therefore be replaced by other actions. These new actions are considered more suitable to the current observed state;
- local: the agent monitors only local parameters. However, the agent can use information sent by its neighbors (from other nodes) to adapt the monitoring process;
- selective: the agent filters the received events and reacts only to those it recognizes. Event classification (see section 5) is used to trigger the appropriate actions.

3 Multi-agent Approach

A multi-agent system is composed of a set of agents which solve problems that are beyond their individual capabilities. Ferber [9] defines an agent as being an entity which: (1) can communicate directly with other agents, (2) possesses its own resources, (3) is capable of perceiving its environment (but to a limited extent), (4) has only a partial representation of its environment (and perhaps none at all), (5) has a behavior which tends towards satisfying its objectives, taking account of the resources and skills available to him and depending on its perception, its representation and the communications it receives.

Multi-agent systems have been used in numerous areas like: (1) the road traffic control ([4], [14]); (2) biologic phenomena simulation like the study of eco-systems

[7] or that of ant-colonies [8], for example; (3) social phenomena simulation like the study of consumer behaviors in a competitive market [5]; (4) industrial applications like the control of electrical power distribution systems, the negotiation of brands, etc.; (5) etc. Multi-agent approach is well suited to control distributed systems. Telecommunication networks are good examples of such distributed systems. This explains partly the considerable contribution of agent technology when introduced in this area. The aim was mainly to solve a particular problem or a set of problems in networks like: the discovery of topology in a dynamic networks by mobile agents ([13],[17]), the optimization of routing process in a constellation of satellites [19], the fault location by ant agents [20], and even the maximization of channel assignment in a cellular network [6].

Our approach consists in integrating agents in the different network nodes. These agents optimize the networks QoS parameters (delay, jitter, loss percentage of a class of traffic, etc.), by adaptively monitoring the network elements, the traffic nature and volume, and the user profile and his (her) habits. Agents can be reactive, cognitive, hybrid or adaptive [3], [7], [22]. Reactive agents are suitable for situations where we need less treatment and faster responses (actions). Cognitive agents, on the other side, allow making decisions and planning based on deliberations taking into account the knowledge of the agent about itself and the others. Adaptive agents can adapt their actions and parameters to the changing situations. Hybrid agents are composed of several concurrent layers. In INTERRAP [15], for example, three layers are present: a reactive layer, a local planning layer, and a cooperative layer. The approach we propose is different. In fact, every node has one cognitive agent that supervises, monitors, and manages a set of reactive agents. Each reactive agent has a specific functioning aiming to optimize some monitored parameters. The cognitive agent (we call it Master Agent) is responsible for the MMs' selection of the different reactive agents it monitors, regarding the current situation and the occurring events.

4 The Two Levels Monitoring Process

Two kinds of agents are defined in our model: (1) Master agent: which monitors the other agents in addition to what is happening at the level of the node; (2) the other agents: which monitor some local parameters like loss percentage, etc.

We can distinguish two levels of monitoring within a network node. These two levels are the following:

Management Mechanisms' Level (Level 0)

This level is composed of the different node's management mechanisms, which are currently activated. Each management mechanism has its own parameters, conditions and actions, which can be monitored and manipulated by the entity lying at the level 1 (the Master Agent). The functioning of a MM is limited to the execution of the loop (conditions → actions); therefore, this is represented by a method of a reactive agent. We can find different agents at this level (Scheduler Agent, Queue Manager Agent, Admission Controller Agent, Routing Agent, etc.). Each one of these agents is responsible for a specific task within the node. So each agent responds to a limited set of events and performs actions ignoring the treatments handled by other agents lying in the same node or in the neighborhood.

Adaptation Level (Level 1)

This level supervises, monitors, and manipulates the entities of level 0. A Master agent is lying at this level and is responsible for the different interactions with the other agents like the cooperation, negotiation, messages processing, etc. This agent possesses a model of its local environment (its neighbors) that helps him to take its own decisions. It chooses the actions to undertake by consulting the current system's state (neighbors nodes state, percentage of loss, percentage of the queue load, etc.) and the meta-rules it has at its disposal in order to have only the most relevant management mechanisms activated with the appropriated parameters. The node, thanks to the two monitoring levels (which are in the same time its decision levels), responds to internal events (loss percentage for a class of traffic, load percentage of a queue, etc.) and to external ones (a message sent by a neighbor node, reception of a new packet, etc.).

Actions of the Master Agent

The actions of the Master Agent adapt the node's MMs and may consist in:

- inhibiting the MM (action (3) on C in Figure 1): the inhibition happens when this MM becomes useless regarding the current node's situation and rules;
- modifying the internal functioning of the MM (action (2) on the MM B): this modification appears by updating the parameters on which the MM depends like the thresholds (MinTH and MaxTH) for the RED management mechanism or the weight of each queue for the WFQ management mechanism, for example;
- activating the MM (action (4) on the MM D): the activation takes place if the Master Agent considers that this management mechanism is appropriate to the current node conditions. This activation may be accompanied by the inhibition of other MMs to avoid the coexistence of contradictory MMs;
- letting the active behavior running (action (1) on MM A): this occurs when the MM is still relevant to the current conditions. So, the Master agent can continue to make the same actions and monitor the same set of data and parameters.

The Master Agent uses meta-rules in order to decide on actions and monitoring to perform. Each Master Agent possesses a set of rules allowing to select the appropriate management mechanisms to activate, and therefore to select the best actions to execute. These rules respond to a set of events and trigger the actions which affect the MMs supervised by that Master Agent. Their role is to manage a set of management mechanisms in order to provide the best functioning of the node and to avoid incoherent decisions within the same node. These rules give the node the means to guarantee that the set of actions executed, at every moment, by its agents are coherent, in addition to be the most relevant to the current situation. The Master Agent owns some modules and each of them is responsible for a particular task (figure 2).

The actions undertaken by the node have local consequences but may influence the decisions of the other nodes. In fact, by sending messages bringing new information on the sender node's state, a receiver's Master Agent rule may be triggered. This can

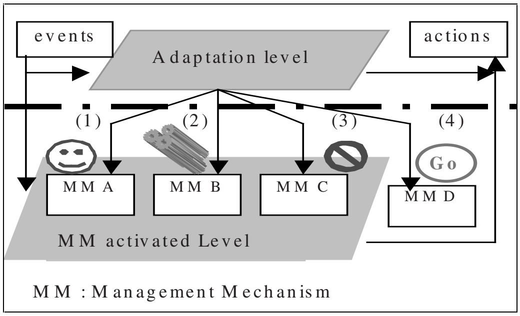

Fig. 1. Operations on management mechanisms

involve a change within the receiver node (the inhibition of an activated management mechanism, or the activation of another one, etc.). This change may have repercussions on other nodes, and so forth until the entire network be affected.

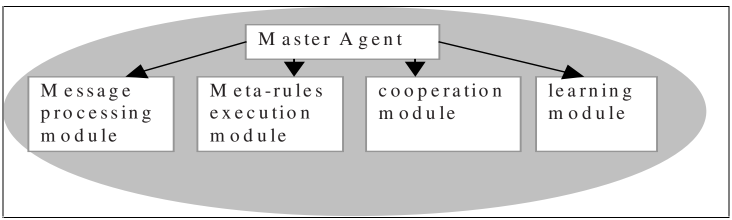

Fig. 2. Master Agent's main modules

This dynamic process aims to adapt the network to new conditions and takes advantage of the agents' abilities to alleviate the global system. We argue that these agents will achieve an optimal adaptive monitoring process because of the following two points: (1) each agent holds different processes (management mechanisms and adaptive monitoring of these mechanisms) allowing to take the most relevant decision at every moment; (2) the agents are implicitly cooperative in the sense that they possess rules that take account of the neighbors' state in the process of management mechanisms' selection and monitoring (section 5, example 1).

The Master Agent meta-rules and the dynamic they create are represented by an Augmented Transition Network (ATN) [21]. In an ATN, two concepts are fundamental: states and transitions. For us, a state represents the current activated management mechanism, while the transition represents a rule, that is, a set of events and the actions they cause. Each Meta-Behavior has its own ATN. In figure 3, we can see the queue management rule-based ATN. Three rules, and consequently three transitions, are represented. Each transition is labeled by the rule engendering it. The state PRIOR means that the activated queue MM is PRIOR with the parameters given by the rule action. The Master Agent has dedicated ATNs to represent the management mechanisms dynamic. One can argue that it is unnecessary to associate an ATN to each management task and that we can have only one ATN. In this case, a state will look like: FIFO_StaticRouting_PQScheduling. This implies an ATN with a very important size and we are not favorable for such a solution.

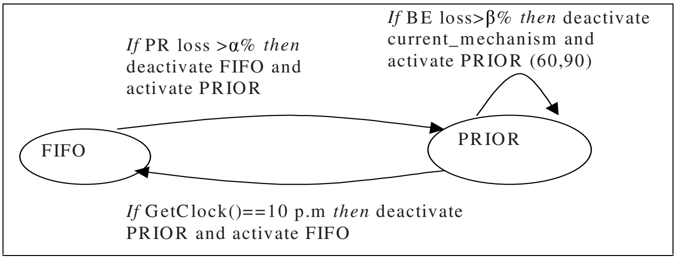

Fig. 3. Queue Management Meta-behavior ATN

The current ATN are provided by the system's designer but it is possible to enrich them by a learning process. This issue is beyond the paper purpose. The Meta-rules-Execution_Module is responsible for deciding when the meta-rule must be executed regarding the events it receives.

5 Events and Actions Model

Each agent holds a set of rules defining the manner of realizing the tasks. A rule is of the following form: *if Events then Actions* and its formal definition is given in figure 4. An event can be: a message, the value of a parameter or time-dependant.

```
Rule::=<Events>; <Actions>
Event::=<message>|<parameter_value>|<time_based>
Message::=<Simple_message>|<Normalized_message>
Simple_message::=<Source>; <Destination>; <Msg_Type>
Normalized_message::=<KQML_message>|<ACL_message>
parameter_value::=<attribute>;<operator>;<value>
time_based::=<agent.current_clock>;<operator>;<value>
```

Fig. 4. Rule description

Message-Based Event

This message carries information likely to be interesting to the receiver. It can be simple (and light, by the way) carrying only its type (Message M1 in Rule 1 means that a congestion has occurred or will soon occur) as it can be more developed. In that case the message is enclosing an order, an advice, etc. and is respecting a formalism like KQML [23], ACL [24], etc. A message can be broadcasted to all agents or to a given group of agents, or sent in an end-to-end manner.

Rule 1: *if the message received is of type M1 then*
add the message source to congested_nodes_list
use dynamic routing avoiding this node until it recovers from its congestion

Each Master Agent owns a message base containing the messages to which it responds. When the message arrives, it is inqueued before being removed from the messages queue by the get_msg method. The match_msg method (figure 5) has the role of deciding if the event belongs really to those events the agent deals with. In this case, the Meta_rule_Execution_Module will be notified. It is neglected and the treatment of the next event starts otherwise.

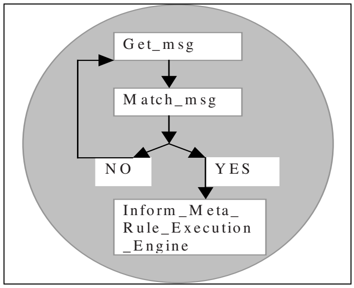

Fig. 5. Message treatment process

Parameter-Value Based Event

This event can be a single value or an interval. Therefore, the rule can look like: *if X= α then actions* (Rule 2) or *if X in [α, β] then actions* (Rule 3).

Rule 2: *if all the received packets are Best Effort (their percentage = 100%) then*
deactivate Prior queue management mechanism
activate FIFO queue management mechanism

Rule 3: *if Premium loss percentage > 0.0001% then*
deactivate FIFO queue management mechanism
activate Prior queue management mechanism with parameters (α, β)

Time-Based Event

This parameter becomes more and more important, especially because of the personalization of the users contracts allowing predicting -to some extent-, the nature and the amount of traffic within the network. Therefore, decisions may be achieved based on the current time (in addition to the date). In Rule 4, the node knows that, from 10 p.m., the clients (which have already negotiated their QoS), requiring a Premium class [16] in rush hours are no longer Premium after 10 p.m. It is consequently obvious that a simple FIFO mechanism has to be activated.

Rule 4: *if getClock() = 10 p.m. then*
deactivate Prior queue management mechanism
activate FIFO queue management mechanism

Reactive Agents Events and Actions

Events to which the reactive agents respond are of type 2: they depend on a value parameter. CAREFUL (see [10] for a more detailed description) for example, a PRIOR queue MM, depends on the current queue load parameter. As we can see in figure 6, following the current queue load, the agent state changes causing a different response to a particular event. Let consider the event we are interested in is a best effort packet arrival. If the current agent's state is *stable queue*, the action to execute will consist in putting the packet in the queue according to its priority. But if the current state is *close congestion* then the packet will be dropped. This proves that the context in which an event occurs influences a lot the system's response to this event.

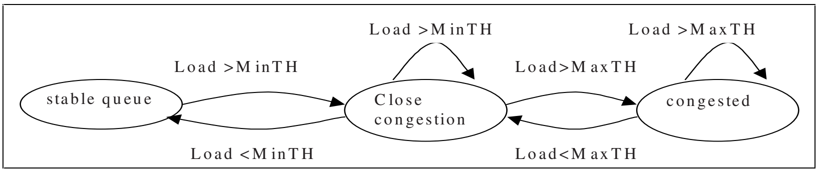

Fig. 6. PRIOR queue management mechanism

6 Examples of Adaptive Monitoring

In order to illustrate our monitoring approach, we propose some examples and the results of their simulations with the oRis Multi-Agent simulator [25].

Example 1

The management mechanism we are interested in is the queue management one with three rules (Rules 3 and 4 of the section 5, in addition to the rule 5 below). We suppose that each node has one queue and that the simulation starts with the activation of a FIFO mechanism.

Rule 5: *if Best Effort loss percentage > 35% then*
deactivate current_QueueManagement_mechanism
activate Prior queue management mechanism with parameters (60, 90)

The node, by monitoring the loss percentage of Premium packets, comes to notice that this percentage is very important (Figure 7), and because it can not tolerate such a bad performance, it will execute the action consisting in activating PRIOR (30,50) (rule 3 triggered, with (α= 30,β=50). This mechanism, detailed in [10, 11, 12] and consisting in monitoring two thresholds of the queue load and changing the actions concerning the incoming packets according to the current queue load, is able to guarantee a better treatment of these packets. By adopting PRIOR mechanism, the node begins to monitor a different parameter (in addition to those already monitored). This parameter is the Best Effort packets' loss. After a certain time, the node notices that there is an important loss of Best Effort packets.

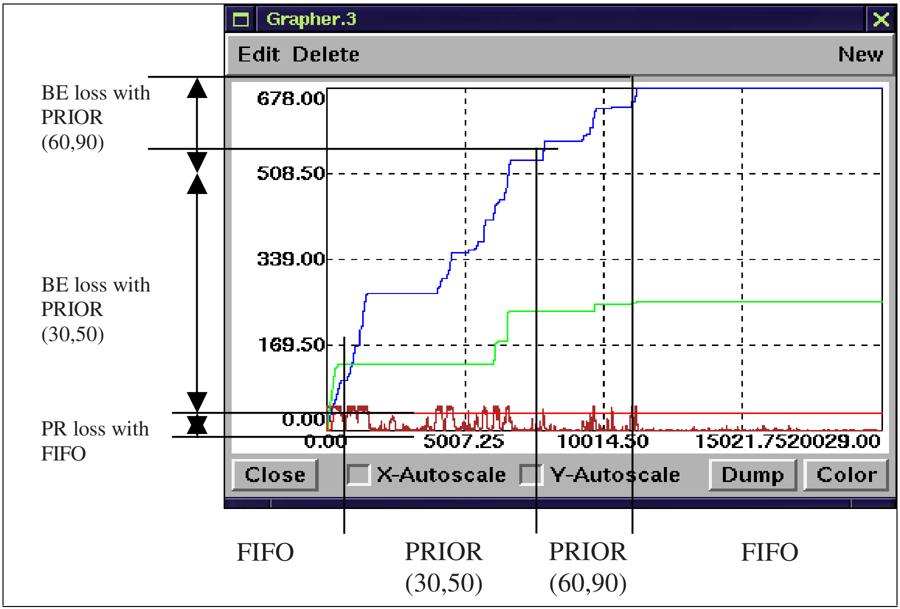

Fig. 7. Loss with an adaptive queue management

In order to respond to such an event, the node executes the action consisting in adapting the parameters of the currently activated queue management mechanism. As a result, it will use PRIOR (60,90) (Rule 5) which will permit to more Best Effort packets to be accepted in the queue. But at 10 p.m., all the packets are considered to belong to the same class of traffic: Best Effort one. Subsequently, it is unnecessary to keep the current queue MM because it is no more suitable; only one class is, from now on, present in addition to the fact that PRIOR mechanism is more complex and, consequently more time-consuming, than a simple FIFO mechanism.

One can notice that we used fixed parameters for PRIOR activation. This is done only in order to give simple and clear examples. The action may look like ”activate PRIOR with (α,β) α is randomly chosen in [0,95], and β in [3,100]” or “activate PRIOR (current_MinTH- α, current_MaxTH – β).

Example 2

In this example, we will demonstrate the gains introduced by adaptive monitoring of scheduling mechanisms. We suppose that the scheduling algorithm used at the simulation beginning is Round Robin, each class of traffic has its own queue managed by FIFO mechanism and the nodes possess the following rule:

Rule 6: *if (Premium loss >0.0001% OR Olympic Loss is >3%) then*
deactivate Round Robin
activate Priority Queuing

Figure 8 shows that the Olympic packets loss becomes almost insignificant from the moment that Rule 6 is applied. An important point, we would like to attract your attention on, is the fact that by monitoring the loss Olympic parameter, we could optimize another important parameter which is the Olympic delay (Figure 9). This proves that by carefully choosing the rules, one can optimize many parameters at the same time.

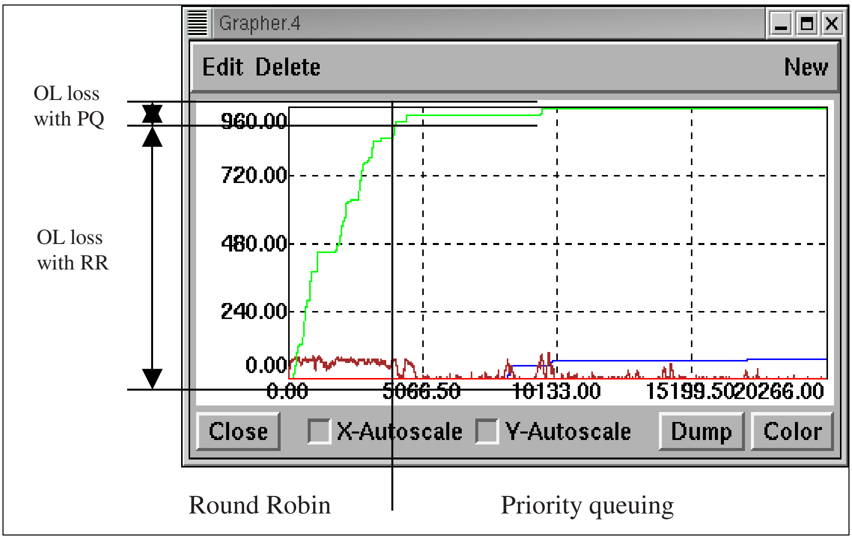

Fig. 8. Loss with adaptive scheduling

Example 3

Routing is another management mechanism that we think adaptive monitoring can optimize. In this simulation (Figure 10), PRIOR (60,90) is used as the queue MM (each node has one queue for the 3 classes of traffic) and the Rule 1 (section 5) is the one we are interested in. We can see that, after the reception of the message M1 by the neighbors of the node in question, this one minimizes Olympic packets loss. This demonstrates to what extent an adaptive routing can be advantageous.

These examples show the following aspects: (1) the importance of the monitored parameters is influenced by the current node condition and its current management mechanisms. Best effort packets loss was not important before the activation of PRIOR mechanism, for example; (2) the adaptive selection of the mechanisms controlling the node tasks depends on the occurring events; (3) the adaptive selection of the parameters that must be optimized is very important in order to achieve better end-to-end network management.

7 Related Work

An overview of the main features of on-line monitoring in addition to the description and the analysis of some monitoring systems is presented in [18]. These systems do not use a multi-agent approach. Schroeder classifies events in three classes: hardware-level events, process-level events and application-dependant events. Our system supports these three classes, even if we classify them differently. [1,2] propose a multi-agent dynamic distributed monitoring by using a set of local monitoring agents (LMA) and domain monitoring agents (DMA). The DMA delegates monitoring tasks to LMA by dynamically constituting and re-configuring groups of agents. The proposed approach uses reactive agents monitoring a particular event, while in our model, the same agents are in charge of nodes control (routing, scheduling, etc.) in addition to the monitoring task.

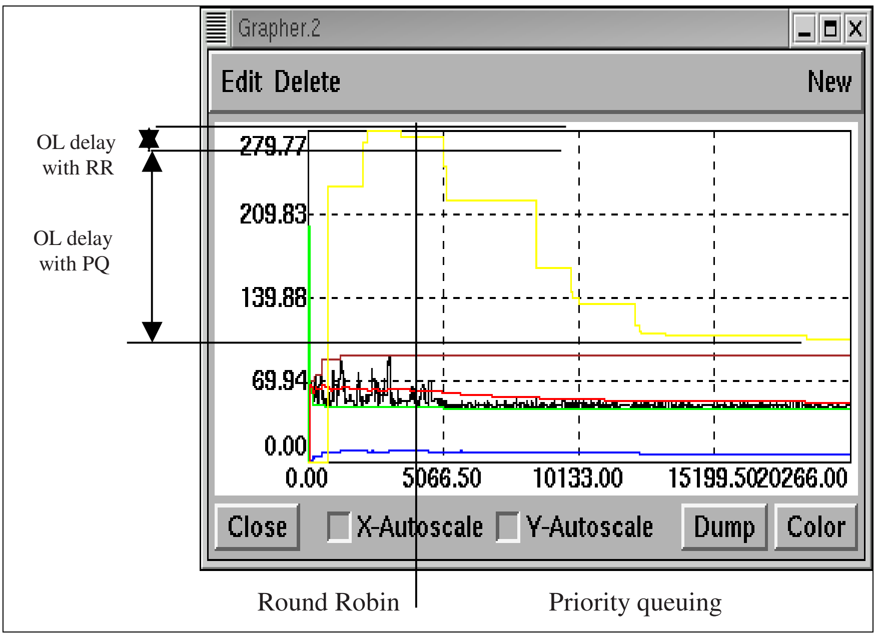

Fig. 9. Delay with adaptive scheduling

The originality of our contribution lies in the fact that we use a purely agent-oriented approach in order to guarantee an adaptive monitoring, by providing an agent-based model and an agent-based simulation. The agent model we propose respect the definition of [9] as: (1) the used agents can communicate by message sending; (2) each agent possesses it own resources (execution space, knowledge base, rules, etc.); (3) the Master Agent knows its environment: the node it belongs to, the reactive agents it is responsible for, etc. and the reactive agents also know the Master Agent and the node in which they reside; (4) the Master Agent possesses some data about the Master Agents of its nearest neighbors like their address, some of their current management mechanisms (further to a message reception), etc.; (5) each agent has an objective, even implicit. The Routing Agent objective is to affect the packet to the appropriate output queue, taking account of the packet's destination, and the current routing mechanism.

Messages transiting between agents are inspired from the agent-based approach. Even if in the current state the used messages do not appeal for the definition of advanced syntax and semantic, our system is open to extensions like KQML or FIPA-ACL. Aspects like cooperation and negotiation often used by the agent community-since the agents evolve in a common society and have common goals beyond their individual abilities- is supported by our model. We chose a multi-agent simulator for our experimentations in order to minimize the bias between the model and its simulation. In fact, the model is more respected because the multi-agent simulator offers the majority of functionalities and features that the agent-based model requires.

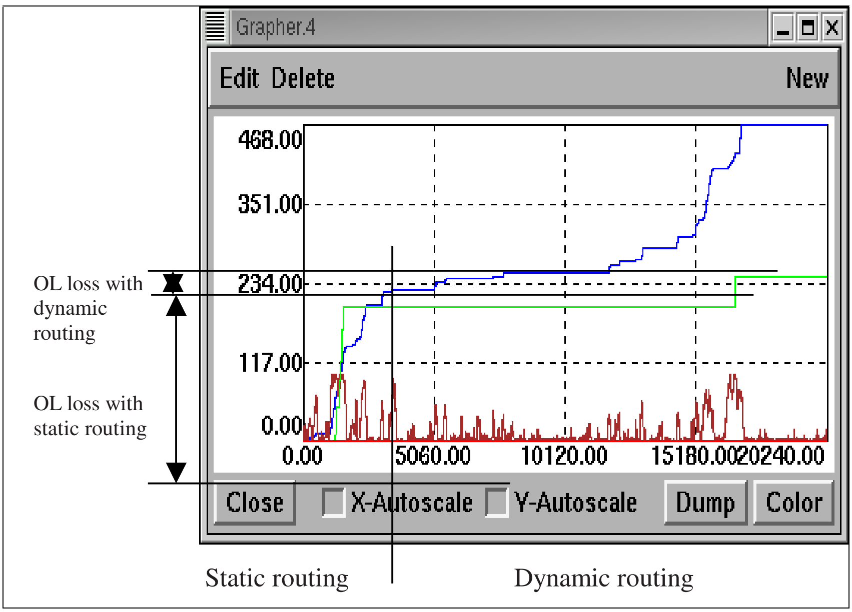

Fig. 10. Loss with adaptive routing

8 Conclusion

A new approach of adaptive monitoring in dynamic networks has been presented in this paper. This approach, based on agents, relies on their abilities like: the autonomy, the decentralization and especially, the adaptability to guarantee a continuous distributed, scalable and adaptive monitoring, aiming to optimize the network performance and guarantee the end to end QoS for the network's users.

We have described examples which clarify our model, and prove also the benefits of our adaptive rule-based selection of management mechanisms. As future work, we intend to use learning techniques to allow more adaptability in the rules' selection.

Anticipation on events is an important issue we want to explore in future work. In fact, activating a given management mechanism occurs only when a QoS parameter has already a poor performance. So, it will be interesting to have anticipative actions instead of corrective ones.

References

[1] Al Shaer E. Adaptive Multicast Group Management for Distributed Event Correlation. International Journal of Networking and Information Systems, pp. 75–88, June 1999.
[2] Al Shaer E. A Dynamic Group Management for Scalable Distributed Event Correlation, IEEE/IFIP Integrated Management (IM'2001), May 2001.
[3] Barber S. and Martin C. Dynamic Adaptive Autonomy in Multiagent Systems: Representation and Justification. IJPR&AI, Vol. 15, N° 3, pp 405–433, 2001.
[4] Bazzan A.L.C., Wahle J. and Klügl F. Agents in Traffic Modelling – From Reactive to Social Behaviour. KI'99, Bonn, Germany, LNAI 1701, pp 303–307 September 1999.
[5] Bensaid L., Drogoul A., and Bouron T. Agent-Based Interaction Analysis of Consumer Behavior. AAMAS'2002, Bologna, Italy, July 2002.
[6] Bodanese E.L. and Cuthbert L.G. A Multi-Agent Channel Allocation Scheme for Cellular Mobile Networks. ICMAS'2000, USA. IEEE Computer Society press, pp 63–70, July 2000.
[7] Doran J. Agent-Based Modelling of EcoSystems for Sustainable Resource Management. 3rd EASSS'01, Prague, Czech Republic, LNAI 2086, pp 383–403, July 2001.
[8] Drogoul A., Corbara B. ad Fresneau D. MANTA : New experimental results on the emergence of (artificial) ant societies".in Artificial Societies: the computer simulation of social life, Nigel Gilbert & R. Conte (Eds), UCL Press, London, 1995.
[9] Ferber J. Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison Wesley Longman, 1999.
[10] Gaïti D. and Merghem L.: Network modelling and simulation: a behavioural approach, Smartnet conference, Kluwer Academic Publishers, pp. 19–36, Finland, April 2002.
[11] Merghem L. and Lecarpentier H. Agents: A Solution for Telecommunication Network Simulation. NetCon'2002, Paris, France, Kluwer Academic publishers. October 2002.
[12] Merghem L. and Gaïti D.: Behavioural Multi-agent simulation of an Active Telecommunication Network, STAIRS 2002, France. IOS Press, pp 217–226, July 2002.
[13] Minar N., Kramer K.H. and Maes P. Cooperating Mobile Agents for Mapping Networks. http://www.media.mit.edu/~nelson/research/route-coopagents/
[14] Moukas A. *et al*. Trafficopter: A Distributed Collection System for Traffic Information. CIA'98, Paris, France, LNAI 1435 pp 34–43, July 1998.
[15] Müller J.P and Pischel M. Modelling Reactive Behaviour in Vertically Layered Agent Architecture. ECAI'94, Amsterdam, Netherlands. John Wiley & Sons, pp 709–713, 1994.
[16] Pujolle G. "Les réseaux". Eyrolles Editions, 2003.
[17] Roychoudhuri R., *et al*. Topology discovery in ad hoc Wireless Networks Using Mobile Agents. MATA'2000, Paris, France. LNAI 1931, pp 1–15. September 2000.
[18] Schroeder B. On-line Monitoring: A Tutorial. IEEE Computer, Vol. 28, pp 72–78, 1995.
[19] Sigel E., *et al*. Application of Ant Colony Optimization to Adaptive Routing in LEO Telecommunications Satellite Network. Annals of Telecommunications, vol.57, no.5-6, pp 520–539, May-June 2002.
[20] White T. *et al*. Distributed Fault Location in Networks using Learning Mobile Agents. PRIMA'99, Kyoto, Japan. LNAI 1733, pp 182–196. December 1999.
[21] Woods W. Transition Network Grammar for Natural Language Analysis. Communication of the Association of Computing Machinery, Vol. 10, N 13, pp 591–606, 1970.

[22] Wooldridge M. Intelligent Agents. In « Multiagent Systems : a Modern Approach to Distributed Artificial Intelligence » Weiss G. Press, pp 27–77, 1999.
[23] http://www.cs.umbc.edu/kqml
[24] http://www.fipa.org/specs/fipa00061/SC00061G.pdf
[25] http://www.enib.fr/~harrouet/oris.htm