Active Network Modelling and Simulation: A Behavioural Approach

Dominique Gaïti and Leïla Merghem
[1] *Laboratoire de Modélisation et de Sûreté des Systèmes (LM2S)*
University of Technology - Troyes (UTT) – France
12 rue Marie Curie, BP 2060, 10010 Troyes Cedex
Email : dominique.gaiti@utt.fr
[2] *LIP6*
University of Paris 6
8 rue du Capitaine Scott
75015 Paris – France
Email :{Leila.Merghem, dominique.gaiti} @lip6.fr

Abstract Network environments become more and more complex in terms of new services, quality of service to handle and user demand for increasing throughput. All these facts have to be performed in real time (or almost), and consequently, network management and control are difficult to realise in this continuously changing environment. The network is now unpredictable and there is a real lack in the modelisation and simulation tools to handle this dynamicity. In this context, agents and multi-agent systems seem to be good candidates to provide a way to model and to manage the network dynamicity. This is due to the fact that agents can perform tasks in an autonomous, distributed, adaptable and cooperative manner. Our goal here is to model an active network by using a multi-agent approach and more precisely a behaviour based agent approach. Instead of representing the actions of the network elements by complex algorithms, we rather describe them by their behaviours which are usually simple and familiar words, inspired in part from human behaviour. On this basis, we have made simulations of an active network environment, and implemented different combinations of four individual behaviours by using a multi-agent platform called Swarm. Results of these simulations are significant and show that we do not loose important packets (with priority) any more when we introduce a behaviour in the nodes.

Keywords: Multi-agent system, Behavioural modelling and simulation, Adaptive systems.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35584-9_19

O. Martikainen et al. (eds.), *Smart Networks*

1. INTRODUCTION

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. The domain of modern telecommunications is characterised by its continuous changes with decentralised data and control, and mobility. The agent technology allows us to manage such a dynamic world, due to its main characteristics namely the autonomy, the ability to communicate with the others in order to solve some common problems in a decentralised manner as well as learning aptitudes.

Our aim is to model an active network by using a multi-agent approach, in order to include some intelligent and dynamic control to avoid congestion and lost packets in order to get a better decentralised management task and global performance of the network.

In order to obtain an agent-based model, several steps have to be followed [5], (1) determine what we have to model, i.e. what are the entities we need to represent, (2) decide what must be active in a network: will it be nodes? (a certain percentage or all the nodes?), will it be packets? (which type of packets?) or the both of them?, (3) choose an agent model (number of agents, their nature (cognitive, reactive, hybrid, adaptive,...), the relations with the other agents (co-operation, negotiation, competition,....), (4) choose the software platform that will be used for the implementation of our model, (5) implement, verify and validate the model.

In this paper, we will not develop all the aspects of an agent-based modelling, but we would rather present the behavioural approach and benefits that are generated. We have implemented this approach with the help of a software platform called Swarm.

The remainder of this paper is organised as follows. We first present the target system of our modelling, i.e. active networks in section 2. Then, we describe some behavioural modelling in other domains than network in section 3. Some behaviours that could be applied in networks are proposed in section 4, followed by the description of the simulations realised up to now and the results obtained (section 5). In section 6 we conclude the paper.

2. ACTIVE NETWORKS – THE TARGET SYSTEM

An active network is a network in which some components are dynamically programmable by third entities (operator, service provider, applications, users) [8].

Active networks are classified by the following approaches:

- active packets: where the deployment of services in nodes is conceptually integrated to the user flow, and in which life span of a service depends conceptually on the traffic in question;
- active nodes: where services are dynamically deployed in nodes, but not in the same flow as the data processed by these services.

Several problems have occurred in current networks like the difficulty to integrate new standards and technologies into the shared network infrastructure, the poor performance due to redundant operations at several protocol layers, and the difficult integration of new services in the existing architectural model [13]. In order to solve these problems or at least a part of them, many projects related to active networks (Smart Packets [12], SwitchWare [1], COMPOSE [14], ANTS and Active IP (MIT) [16]...) are currently being or are already developed in different universities.

Supervision and control of networks are certainly the predilection fields of the active networks. Active networks are also used to make multicast protocols more reliable. It allows to gracefully solve problems that result of the implosion of the negative acknowledgement (NACK) and the useless retransmissions. Active networks can be also used to avoid congestion that may occur during the transmission of multimedia data, by allowing the routers to ask sources to reduce their flows in case of congestion.

However, this new approach generates new problems, due to this increment of complexity. There are new variables to be taken into account and also new resources to be managed and protected.

We are mainly interested in the dynamicity and complexity introduced by the active network concept through the active packets or active nodes. Conceptually, nothing in that kind of network will remain static. This dynamicity needs to be modelled and simulated and we propose the behaviour-oriented approach to do it, because no tool in the telecommunication area provides a real answer.

3. BEHAVIOURAL MODELLING

We propose to use behaviours to model the activities of the different network's elements. Thus, each element will have one or more behaviours, and its interactions with the others will be based on that behaviour. Moreover, instead of representing actions of network elements by complex algorithms, we rather describe them by simple and familiar words, inspired from human behaviour. We say for example that a node is faithful instead of saying that it always sends its packets to the same node, by following the same way, whatever the conditions of this route are. Thus, if a node A says to a node B that its behaviour is faithful, it must not send it the actions that it is performing concerning the routing for example, because B can easily predict them. The profile of a node will be the association of its elementary behaviours, and interactions between the network's nodes will be the result of the interactions between their behaviours. The dynamicity of an active network will be provided by the change of behaviour in response to the local element's environment.

We are mainly interested in behavioural modelling because there are several successful behavioural simulations in other fields than networks (road traffic simulation ([3], [10], [15])), biologic and social phenomena simulation ([4], [6]). These fields are clearly as complex as the network area and contain factors that are generally out of human control. Despite these difficulties, scientists have obtain very good results, and have proven the reliability of multi-agent simulations, and specially their utility in understanding, explaining, and discovering new phenomena. For example, road traffic simulations can be a good source of inspiration for us. After all, in a telecommunication network, we face lots of problems that are also handled in a road network such as traffic management, priority management.

A good example of these systems is Trafficopter [10], a multi-agent system that provides fast and reliable way of assisting the driver of a vehicle in deciding the most convenient route he has to take to reach his destination. The behaviour of drivers is affected by the information on the traffic conditions that lay ahead sent by the cars that are there now, or have been there recently (helpful behaviour). This information can also be collected as a result of a request sent by the driver (needy behaviour).

In the OLSIM project [3], [15], real time information on traffic is sent, in order to affect the behaviour of drivers and consequently reduce the traffic.

All these behaviours model different entities in the system whatever this system is.

4. BEHAVIOURS TO MODEL

4.1 What do we need to model?

The network entities that should be modelled are: nodes, links, and packets.

According to our point of view, an entity is active if it has a behaviour, which could change according to the state of the network (state of traffic and neighbour nodes, etc...).

For the moment, packets do not have a behaviour; they are routed from one node to another to carry data that might affect the behaviour and decisions of covered nodes. This is also valid to links that are passive entities. Only nodes have one or several behaviours which modify the actions that they are performing on data (packets). Data can be more or less important depending on the quality the user is asking for its flows.

In our system, we distinguish three Qualities of Service [18]:

1. Premium packets have the highest priority. They have a total guarantee for the required service : they are never lost, and are not delayed ;
2. Olympic packets belong to the second class of priority. Packets must reach their destination (no lost) but with no guarantee of time;
3. Best Effort packets have the lowest priority. In fact, these packets are routed only if the bandwidth is available, otherwise they are lost.

These classes are under study in the IETF (Internet Engineering Task Force) normalisation group [18]. Each node has at least one behaviour that reflects its actions and the set of conditions to which it responds.

4.2 What kind of agents ?

Agents can be reactive, cognitive, hybrid or adaptive depending of the autonomy level, the capacity to reason or the ability to know their environment. We have opted for adaptive ones because they support, by definition, our goals. Adaptive agents are indeed entities that change their behaviour in the light of changing circumstances; they can sense their environment and act upon what they sense, hence the term adaptive [2], [5], [7]. They can be therefore a good candidate to represent active entities, that change their behaviour following the network states.

Adaptive agent simulations are often used in research to study issues that are too complex to be addressed by any other way. They routinely are used to model life and other complex, non-linear systems [9]. In this context, we will use adaptive agent simulations to evaluate and to understand complexity issues of the network environment.

4.3 Some behaviours in the network area

The notion of behaviour of a node is important because it allows the other nodes to acquire the profile of this node, and consequently be able to reason and take new decisions on the basis of what should that node do, and what should be its proper reactions.

In this paper, we will focus on the two following behaviours:

a) ***Careful***: this behaviour consists to observe the state of the buffers with the help of two thresholds (figure 1). When the buffer load is below 40%, the node has no particular action to do. But when the first threshold is reached, it starts to reject Best Effort packets coming from its own sources. From 70% of buffer load (2nd threshold), the node rejects all

Best Effort packets (even those coming from other nodes), and sends control messages to its nearest neighbours;

b) ***Careless***: the node starts to reject Best Effort packets coming from its sources when the first threshold (60%) is reached, and rejects Best Effort packets coming from its own sources and from one neighbour (router) if the second threshold (90%) is reached. In this case, it sends control messages to its nearest neighbours.

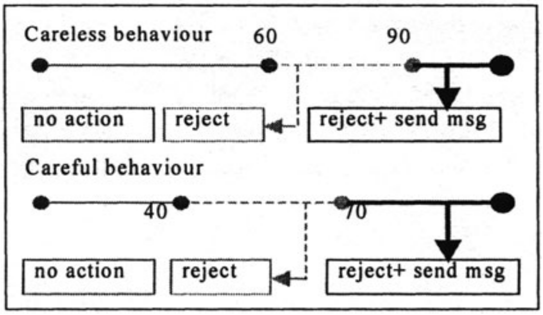

Figure1. Careful and careless behaviours

These two behaviours have been implemented and tested in order to demonstrate the benefit of the concept of behaviour in the network management. Examples of other behaviours that have been defined but not yet implemented are given in the following:

- ***Faithful***: is a node that never changes its routing tables; when an ***Unfaithful*** node will change it according to the current state of the environment;
- ***Nationalist***: if node A's buffers are λ% occupied, then it rejects packets from the source which behaves in a different way than node A. The priority is given to packets coming from nodes having the same behaviour than A;
- ***Self-seeking***: is the opposite of the nationalist behaviour. In fact, this behaviour can be applied if there is a hierarchy between behaviours and it will consist to reject packets coming from nodes having behaviours that belong to a less important class;
- ***Thrifty***: is an important behaviour because it avoids the existence of useless packets. A is thrifty if it does not keep useless packets in its buffers. In fact, if a packet P that must be routed to a node B has λ as a lifetime, and the minimum necessary time to achieve B is γ>λ, then A will kill P. We can imagine also that when a Premium packet is lost the node can reject packets belonging to the same flow;
- ***Equitable***: is a node that manages its buffers with equality vs ***inequitable*** node that starts to process the one with the highest priority.

4.4 Coexistence of behaviours

Including behavioural entities in the network is beneficial because these entities can adapt (i.e. change) their behaviour to the current situation, taking into account the profile of their neighbours, in order to make the most convenient decision. But what can be more advantageous is the use of entities that have more than one behaviour that tends to fill the gap of an incomplete behaviour. In fact, each behaviour responds to a certain number of situations and conditions but not to the whole condition. Including a multitude of behaviours in the same node will make this one more watchful to the changes in its environment, and therefore more adaptive, improving thereby system performance.

4.5 Relations between behaviours

A node can adopt more than one behaviour at the same time; it may have both careful and faithful behaviours for example. But in order to be coherent, the node must not adopt contradictory behaviours simultaneously, and in case this should happen, it will be considered as a node failure.

The relations between the different behaviours can be:

a) Contradictory: careful and careless behaviours are a good example of this relation. Two contradictory behaviours can not exist in the same node at the same time; if this case appears it will be considered as a failure node;
b) Complementary: two behaviours are complementary if the existence of one of them is strengthened by the existence of the other. Thrifty behaviour for example is strengthened by careful or careless behaviour;
c) Indifference: this relation exists between two behaviours that affect different parameters and are triggered by different factors. Nationalist and faithful are indifferent behaviours.

4.6 Change of behaviours

An entity changes its behaviour as a result of the following events:

- reception of a packet from a node which asks it to adopt a certain behaviour (distributed decision);
- change of the conditions of the current behaviour; the node must decide what is the most convenient behaviour to the current network conditions (local decision).

This entity can inform the other nodes of this change and especially the nodes lying in the nearest neighbourhood but it is not mandatory. The other nodes can reason on what they really know or on what they suppose to be true. This behaviour change can be followed by other changes at the neighbourhood level and even at the entire network level.

5. SIMULATIONS

The aim of the present simulations is to measure the benefits introduced by the behaviour concept in a node in terms of loss of Premium or Olympic packets. There are of course other parameters that should be optimised like the response time, the confidence interval, etc. According to the network environment and the chosen parameter to optimise, one of the different tested configurations fits best the current conditions.

The simulations are realised in a multi-agent platform called Swarm [17], that provides a framework to simulate a large set of complex problems. Swarm is a multi-agent, discrete event simulation, domain independent, that is used in different projects. In the Swarm system, the basic unit of simulation is the *swarm*, a collection of agents executing a schedule of actions. Swarm provides object libraries of reusable components to build models and analyse, display and control experiments on those models.

5.1 Parameters of the simulations

Our network topology (figure 2) is composed of six routers (1 to 6) and eight clients (sources: 1a, 1b, 2a, 3a, 3b, 4a, 5a, 6a). A screen catch of that topology is shown figure 3. The simulation environment is a two-dimension grid, on which we place the network's elements. Two kinds of entities are represented: agents and objects.

5.1.1 Objects

The different objects of the simulation are: links (between two nodes), packets (which move autonomously according to the decision taken by the node; a packet has no goal and does not respond to any stimulus), queues or buffers (contain packets waiting to be treated by the node).

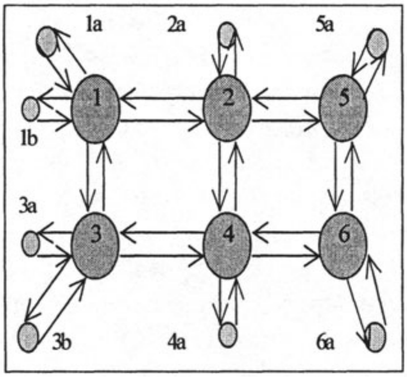

Figure 2. Topology

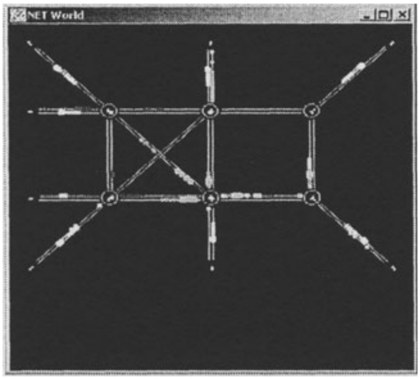

Figure 3. Network topology on screen

5.1.2 Agents

We have defined the following classes of agents:

- Service agent: processes packets lying in the node's buffer. There are two kinds of these agents: router service agent and client service agent. The first one sends packets to other nodes according to the routing table and their destination, while the second one is responsible of flow generation;
- Queue manager: processes the packet arrival and places them in the queue according to the current behaviour (taking into account the type of the packet when the behaviour is selective, careful or careless, or without caring about the priorities of packets (FIFO behaviour, C.f. 5.2);
- Master agent: monitors service agents and queue managers;
- Node agent: is the association of a service agent, a master agent and several queue managers. It represents the entity in its environment.

Each node possesses a static routing table. Packets are sent by a client to another client. The time between two flows is 200 simulation steps. We simulate the different flows in that proportion: 50% Best Effort flow, 30% Olympic flow and 20% Premium flow. In the future we will adopt a 20% Best Effort flow, 60% Olympic flow and 20% Premium flow proportion which will be more realistic. Our choice here was to prove that no behaviour will lead to an important loss of Premium and Olympic packets even if they do not constitute an important percentage of the whole packets in the system.

The size of each queue depends on the number of input links (number of input links *20), so queues of nodes 1, 2, 3 and 4 have 100 places, where queues of nodes 5 and 6 have 60 places.

5.2 Implemented behaviours

Four behaviours have been implemented in the current simulations. A node can have one of the following behaviours:

- ***FIFO:*** in that case, the node processes the packets in a FIFO manner; it represents the current routers;
- ***Selective:*** the node places packets in the queue according to their priority, but when its buffer becomes full, it will reject all packets even the Premium and Olympic ones. Client agents have this behaviour by default;
- ***Careful and careless***: have been explained above (see 4.3).

The two last behaviours are different from the others by the fact that they never loose Premium packets. These two behaviours reject Best Effort packets in advance in order to gain time and buffer load. In fact, by starting to loose Best Effort packets before the buffer becomes full, we let more slots for Premium packets (and eventually Olympic ones) that should arrive later. An important gain of time can be reached when we avoid to treat Best Effort packets in the queue and to discard them (if the buffer is full) when Premium packets arrive.

5.3 Results

We have made simulations with several configurations in order to see the impact of the use of each behaviour on the node performances, and consequently the global system. In this paper, we will focus on seven configurations that will illustrate the main results that we got. These results are given in Table 1 (I: careless, P: careful, F: FIFO, S: selective)

When all the nodes have queues which proceed in a FIFO manner (we will call that nodes FIFO nodes) (figure 4), we have the worst results in term of global loss (35,16 %, Table 1). This result is decomposed into 42,66% of Best Effort lost, 34,37% of Olympic lost and 22,97 of Premium lost.

In fact, by using selective nodes (figure 5), the global loss drops from 35,16% with the FIFO nodes to 30,55% with the selective ones. But we cannot be satisfied with these results because the percentage of Premium and Olympic lost packets remain important (Olympic packets represent 35,61% of the lost packets, and the Premium ones 25%).

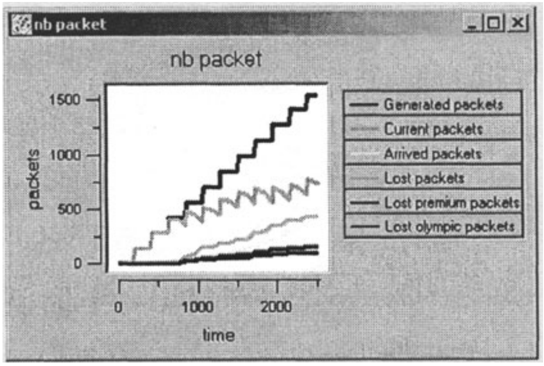

Figure 4. Configuration with only FIFO nodes

Table 1. Results of some simulations

	Global loss (GL) in %	Best Effort loss/GL	Olympic loss/GL	Premium loss/GL
F F F F F F	35,16	42,66	34,37	22,97
S S S S S S	30,55	39,39	35,61	25
I I I I I I	29,17	98,87	1,13	0
P P P P P P	31,48	100	0	0
S P I I S P	8,44	92,31	3,84	3,85
S P I S P I	8,44	38,46	40,77	20,77
I S P I S P	10,06	100	0	0

Configurations with careless nodes only (figure 6) is better than the two previous ones, because we loose only 1,13% of Olympic packets, the reminder of loss concerns Best Effort packets. This is due to the fact that when a node has this behaviour, it rejects Best Effort packets before that the buffer becomes full, giving more space to Premium or Olympic packets that should arrive. The loss of Olympic packets occur when the queue is only composed of Premium and Olympic packets, and in this case the Queue

Manager is obliged to remove one Olympic packet from the queue in order to place the arrived Premium packet. This lost remains insignificant.

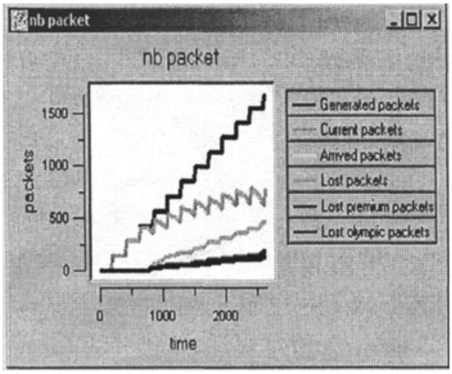

Figure 5. Selective nodes only

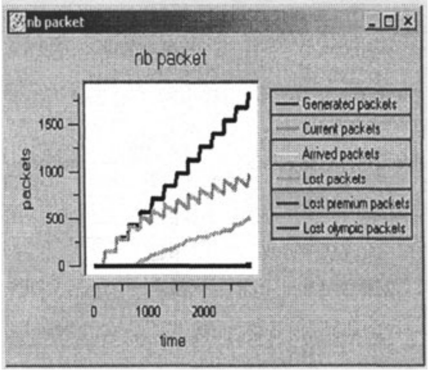

Figure 6. Careless nodes only

Better results are achieved when all the nodes are careful (figure 7). In fact, in that case, we do not loose Premium or Olympic packets, because we start to reject Best Effort packets from low thresholds of buffer load in order to let space for Premium or Olympic packets.

A configuration with the three behaviours (careless, careful and selective) confirms the results obtained above (figure 8).

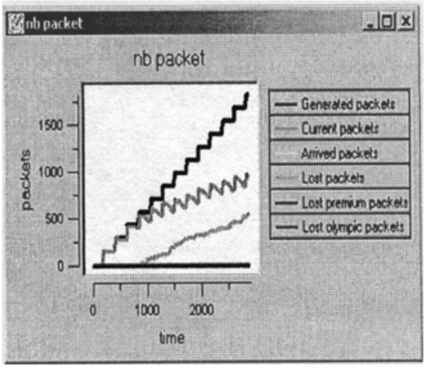

Figure 7. Careful nodes only

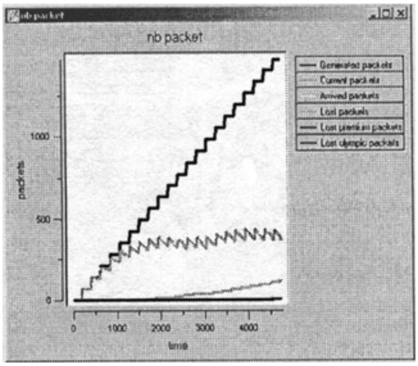

Figure 8. 2 S, 2 I and 2 P nodes

In fact, node 1 which is selective here (Table 1 row 5, figure 9) looses Premium and Olympic packets, whereas the node 3 (careless node) is responsible of the majority of Best Effort packets' loss. The number of lost packets depends on the node's behaviour and also on its position in the network. In fact, we notice that the nodes 1 and 3 are always those that cause the most loss because they are quite loaded (two sources) whereas the other nodes have only one source, and so less traffic (see figure 10 where we describe the node 4 in the same configuration than the node 1 in figure 9). This is clearly illustrated in figures 9, 10, 11, 12, 13 and 14.

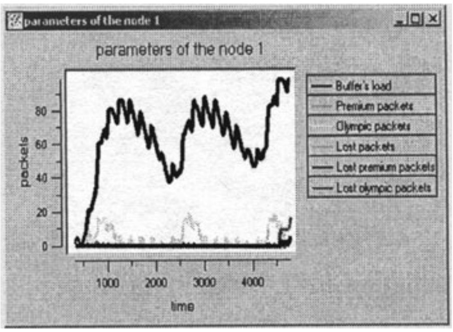

Figure 9. Node 1 in the (SPI/ISP) configuration

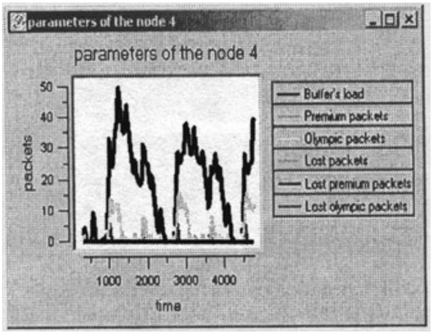

Figure 10. Node 4 in the (SPI/ISP) configuration

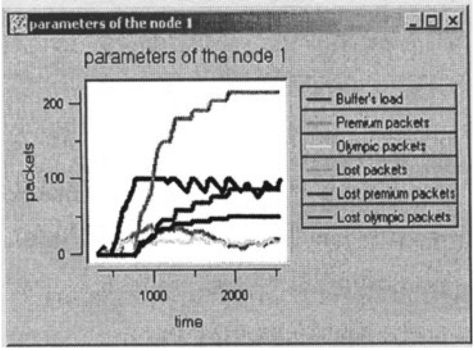

Figure 11. Node 1 in all FIFO configuration

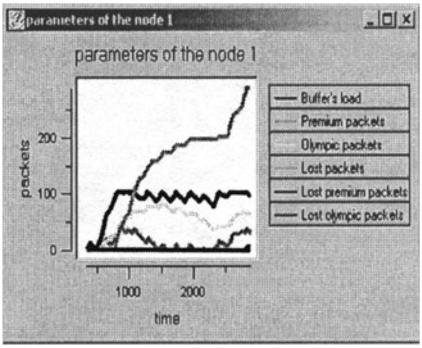

Figure 12. Node 1 in all careless configuration

The simulations show that the use of routers having behaviours gives better performance in terms of lost packets (Best Effort packets and especially Olympic and Premium ones), than when we use FIFO routers. This is due to the fact that we favour the processing of Premium and Olympic packets by putting them in the head of the queue, and moreover with careful and careless behaviours where we start to reject Best Effort packets before the buffer becomes full in order to let more space to Premium or Olympic packets that should arrive later.

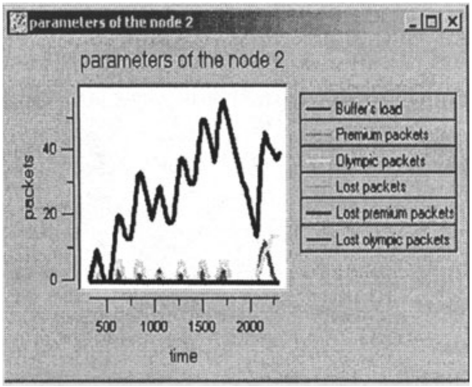

Figure 13. Node 2 in the all selective configuration

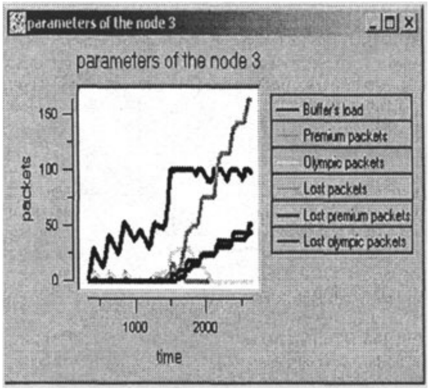

Figure 14. Node 3 in the all selective configuration

We proved through the different simulations that careful and careless behaviours represent better results than selective one concerning the loss of Premium or Olympic packets. Further simulations will be done by introducing the time parameter and by implementing and evaluating other behaviours, that should change in the light of the network change.

6. CONCLUSIONS

This paper presents a multi-agent simulation of an active network realised with the Swarm platform. Using agents in telecommunication network area is not new; many mobile agents platforms are used in many projects (for more details see [11]), but our approach is however innovative for three reasons: (1) the use of behaviour to model network entities, (2) the use of behaviour to simulate the dynamicity of the network, and (3) the use of simple concept inspired from the human model to name these entities. In this light, many behaviours have been defined, but only four of them have been implemented and tested in order to find the best combination regarding the number of lost packets (Best Effort, Olympic or Premium). The first results are really encouraging.

Adaptive agents are in a position to represent active entities, and will allow the system to get better performance, because each entity will be able to adopt the best behaviour regarding its local environment conditions, and a global equilibrium will result of the different interactions of these behaviours. This will be the next step of our research because it supposes cooperation and interaction between agents.

REFERENCES

[1] Alexander D., et al.. The SwitchWare Active Network Architecture. IEEE Special Issue on Active and Controllable Networks, Vol. 12, N° 3, pp 29-36, 1998.

[2] Barber S. and Martin C. Dynamic Adaptive Autonomy in Multiagent Systems: Representation and Justification. IJPR &AI, Vol. 15, N° 3, pp 405-433, 2001.

[3] 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.

[4] Bensaid L. et Bouron T. Multi-agent Simulation of Consumer Behaviours in a Competitive Market. Pre-proceedings of MAAMAW'2001.

[5] Doran J. Agent-Based Modelling of EcoSystems for Sustainable Resource Management. EASSS'01, Prague, Czech Republic, LNAI 2086, pp 383-403, July 2001.

[6] Drogoul A., et al. MANTA : New experimental results on the emergence of (artificial) ant societies".in Artificial Societies, Nigel G. & R. Conte (Eds), UCL Press, London, 1995.

[7] Guessoum Z., Cardon A. and Ramdani A. Toward Self- Adaptive Multi-Agent Systems. MAAMAW'99, Spain, June 1999.

[8] Lazar, A., Campbell, A., Schulzinne, H., Stadler R., Building Open Programmable Multi media Networks, Computer Communications Journal, Vol. 21 No. 8, pp. 758-770, 1998.

[9] Loughran J. Island of Complexity Models and Adaptive Agent Simulations. http://www.sisostds.org/webletter/siso/iss_39/art_156.htm

[10] Moukas A., Chandrinos K. and Maes P. Trafficopter: A Distributed Collection System for Traffic Information, CIA'98, Paris, France, LNAI 1435 pp 34-43, July 1998.

[11] Perdikeas M.K., et al. Mobile Agents Standards and Available Platforms. Computer Networks Journal, Elsevier Publisher, Holland, vol. 31, n° 19, pp 1999-2016, 1999.

[12] Schwartz B., et al.. Smart Packets for Active Networks. OpenArch'99, March 1999

[13] Tennenhouse D.L., et al. A Survey of Active Network Research. IEEE Communications Magazine, volume 35, n°1, pp 80-86, January 1997.

[14] Thibault S., Marant J. and Muller J. Adapting Distributed Application Using Extensible Networks. IEEE ICDCS'99, Austin, TX, June 1999.

[15] Wahle J.,Bazzan A.L.C, Klügl F. and Schreckenberg M. Anticipatory Traffic Forecast Using Multi-Agent Techniques. http://www.traffic.uni-duisburg.de/

[16] Wetherall D., Guttag J. and Tennehouse D. ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols. IEEE OpenArch'98.

[17] http://www.swarm.org

[18] http://www.ietf.org