A Multi Agent System Approach for Self Resource Regulation in IP Networks

Gérard Nguengang[1, 2], Louis Hugues[1], and Dominique Gaiti[3]

[1] R&D Ginkgo-Networks, 8 Rue du Capitaine Scott 75015 Paris, France
[2] Laboratoire d'Informatique de Paris 6, 8 Rue du Capitaine Scott 75015 Paris, France
[3] Charles Delaunay Institute – environment of autonomous networks team,
University of Technology of Troyes, 12 rue Marie Curie, BP 2060, 10000 Troyes, France
{gnguengang, lhugues}@ginkgo-networks.com,
dominique.gaiti@utt.fr

Abstract. As Internet is becoming the global infrastructure for all media communications, ensuring the reliability and the quality of network services is far from being a secondary task. Indeed, many business activities rely on the network and any performance degradation can result in serious financial losses. The main objective of Internet Service Providers is to maintain permanently an acceptable service delivery for their subscribers and respect scrupulously the Service Level Agreements. To achieve this goal, just committing bandwidth resources is not sufficient since network services are not safe from breakdowns. The dysfunction of some of its elements or persistent overloads generated by bursts traffics can induce serious deterioration of the network performance and impact the end users' applications. Continuous service monitoring is then required and an autonomous network resource regulation mechanism is necessary in such situation to avoid the degradation and the collapse of all the applications sharing the impacted resource. In this paper, we propose a decentralized approach for the monitoring and management of the network backbone shared resource by the mean of distributed autonomous agents. The agents are deployed at the edge routers and cooperate together in order to maintain a global acceptable level of service in critical situations where the current quality of service is less than the expected one. This allows self adaptive service and resource management with an interesting abstraction from network backbone heterogeneity and complexity.

Keywords: Agents, Resource management, Network performance, Active probing.

1 Introduction

Due to the emergence of multimedia applications such as real time voice and video over Internet and the increase of business dependence on IT infrastructures, computers networks are involving to support services with diverse performance requirements. This is possible thanks to new scheduling algorithms that allow differentiated packets treatment and offer thus several forwarding alternatives. Initially conceived to provide the single best effort service, Internet is actually a multi

D. Gaiti et al. (Eds.): AN 2006, LNCS 4195, pp. 64 – 75, 2006.

service network. Internet Service Providers (ISP) are now offering a portfolio of services with guarantees in terms of end-to-end delay, jitter, and packets loss. Subscribers can choose among different levels of Quality of Service in order to best meet their applications and pricing constraints. The service and its delivery quality are negotiated through a contract, the Service Level Agreement (SLA).

However, to ensure reliability and good quality of service on these networks is not an easy task. The commitment of network resources is not sufficient to eliminate possible performance degradation. Indeed, persistent overloads in the current Internet is unavoidable [1] and can arise for several reasons: a single flow not using end-to-end congestion control and continuing to transmit despite encountering a high packets drop rate (UDP flows), the dysfunction or outage of a network element, etc… When such a situation occurs, all flows crossing the overloaded resource experienced significantly degraded service over an extended period of time. This is unacceptable since any SLA violation can lead to severe performance degradation and generate serious financial losses for those subscribers whose business activities rely on the network. Therefore, continuous performance monitoring is required to tract the ongoing QoS, compared the monitored QoS with the expected performance, detect possible QoS degradation and then take correctives actions accordingly to sustain as much as possible the agreed QoS for at least critical data. With the vastness of the current networks, the permanent monitoring of each network element in order to track any performance degradation proves to be expensive and sometimes inadequate. Our approach for solving this problem is to consider the network backbone as a shared resource among all the end users and integrate in each edge router a software agent responsible of the monitoring of the end-to-end quality of service and the execution of correctives actions in case any anomaly is detected. The agents cooperate together to coordinate their actions. Our Multi agent architecture makes it possible to ensure proactively the respect of some SLAs when, for an unforeseeable reason, a network service is unable to satisfy all the QoS requirements of its traffics.

This paper proposes a Multi Agent System approach for self adaptive resource management. The aim is to maintain an acceptable level of service delivery when a performance problem occurs in the network with an interesting abstraction from the network backbone heterogeneity and complexity. The organization of the paper is as follows. Section 2 provides a survey of related research in the area of adaptive network resource management. Section 3 gives an overview of our approach. Section 4 describes in details the agent architecture. The experimental setup used to evaluate the concept and some results are presented in section 5. Section 6 summarizes our proposal and addresses future works.

2 Related Works

Performance management is nowadays a key issue in the network management process. Network management systems must evolve to allow fast anomaly detection and rapid problems recovery in order to maintain a good level of service and therefore the respect of the SLA for all the applications using the network. Several researches were undertaken during these last years to propose solutions for an adaptive performance and resource management in IP Networks. The aim of this section is to present existing works in the field of adaptive resource management.

The current tendency in network management is the self-aware management, that is, enabling the management processes and the subjacent infrastructure to organize themselves and operate without external assistance [2]. It includes self-configuration, self-optimization, self-healing and self-protection. In terms of resource management, increasingly sophisticated techniques are needed to improve the overall service quality and minimize the effects of potentially damaging periods of poor service [3]. Vilà and al [4] have proposed a dynamic bandwidth management scheme in logical network such as ATM or MPLS based on distributed agents. The system consists in two types of agents: The Network Monitoring agents (M) and the Network Performance agents (P). Each node has one P agents and several M agents. The M agent monitors and controls a single logical path (LP). P agents are responsible of the supervision and the collaboration of the M agents of the same node and cooperate with its peers to perform bandwidth management and re-routing of the LPs. For instance, if a LP is congested, its M agent will detect the congestion (threshold violation) and warn the P agent. The affected P agent may send a message to one of its neighbors requesting some action. The P agent that receives the message firstly merges the partial network view and then, it uses its actualized partial network view to make a decision or take an action. [5] also proposed a decentralized architecture of autonomous, collaborative agents that can model normal network operations, detect departures from expected behaviors, and take remedial actions to avoid performance degradations. Each agent creates a thumbprint of network behavior under normal circumstances. Agents compare these thumbprints to observe performance and detect deviations from normal activities. Although the aforementioned proposals enhance the control and the management of network resources and allow fast recovery of network performance when network anomalies occur, their implementation requires the “agentification” of all the network elements since all the nodes are involved in the management process. Also, the network monitoring is not service-oriented. Even if no performance degradation is detected, these systems cannot ensure that the network services QoS requirements are respected.

[6, 7, 8] have proposed measurement based admission control (AC) algorithms for efficient network resource management. A new flow is admitted in the network if the current network load permits the provision of its QoS requirements. In [9] a distributed service-oriented traffic control mechanism based on online active monitoring is proposed. AC decisions are made based on feedback from edge-to-edge on-line measurements of service specific QoS parameters. These proposals present a very attractive solution because there is a real possibility of making reasonable estimates of the properties of end-to-end paths from edge-based measurements. However, if a network service experiences suddenly performance degradation, all the flows using it will be impacted.

3 Overview of the Approach

In an ideal world, the network should be able to proactively detect service disruption or performance degradation and freely take corrective actions to overcome or minimize their effects on network end users applications without any human intervention and in the respect of the SLAs. This is impossible with the classic centralized network management paradigm based on the periodical polling of network devices. The amount of information and computation needed to estimate in real time the network state is huge and the process of data aggregation, treatment and decision can take too much time. It is thus important to divide the problem for better solving it. Multi agent system paradigm provides a decentralized approach to solve difficult problems in complex environments. One of the main ideas of multi agent system is to generate approximate solutions to hard problems by distributing them to autonomous rational problem solvers (agents) that have local problem solving capabilities and are able to find a solution for the whole problem by cooperating with each other [10].

In our approach, we combine the use of distributed agents and the active service monitoring to regulate the network utilization. We make the assumption that the network backbone is a shared resource among all the end users. This has the advantage of providing an abstraction of the network core complexity. Figure 1 gives an overview of the system.

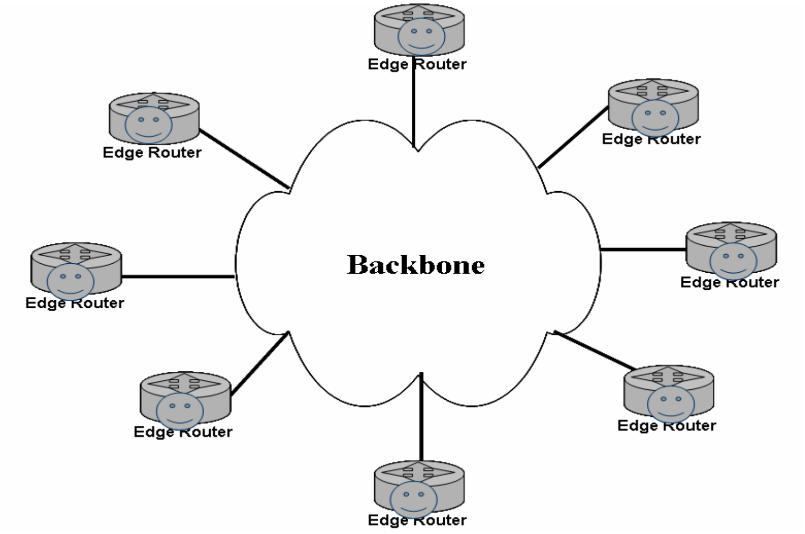


Fig. 1. An Overview of the approach

Each edge router is equipped with an agent. The agent carries out the on-line monitoring of the network services performance. Periodically, on the basis of its measurements, each agent computes a local indicator. This indicator represents the agent perception of the network backbone health, a situated representation of the network core state. This information is thereafter broadcasted to the other agents of the multi agent system. Once the indicators are collected, the agents individually start the diagnosis process and evaluate if an action must be taken or not. If the current network performance requires an adaptation, the agents cooperate together to define the action to undertake and elect the agent responsible of its execution. This is to avoid unilateral agents' answers to a problem which could lead the network to a state of under utilization.

4 The Agent Architecture

Jennings & al [11] define an agent as an entity which is:

- *Situated* in some environment.
- *Autonomous*, in the sense that the system can act without direct intervention from others (humans or other software processes).
- *Flexible*, which is further broken down into three properties: *responsive* (perceives its environment and responds to changes in a timely fashion), *pro-active* (exhibits opportunistic, goal-directed behaviour) and *social* (able to interact with humans or other artificial agents).

This definition corresponds to the capacities we intend to equip our agents. They must be able to act on the control plane of their router in a coordinated way. To achieve this goal, each agent is made up of three fundamentals elements: the knowledge base, the situated network view, and the behaviors. Figure 2 summarizes our agent architecture.

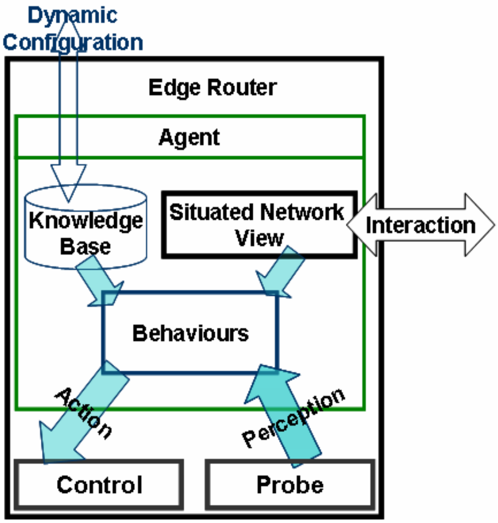
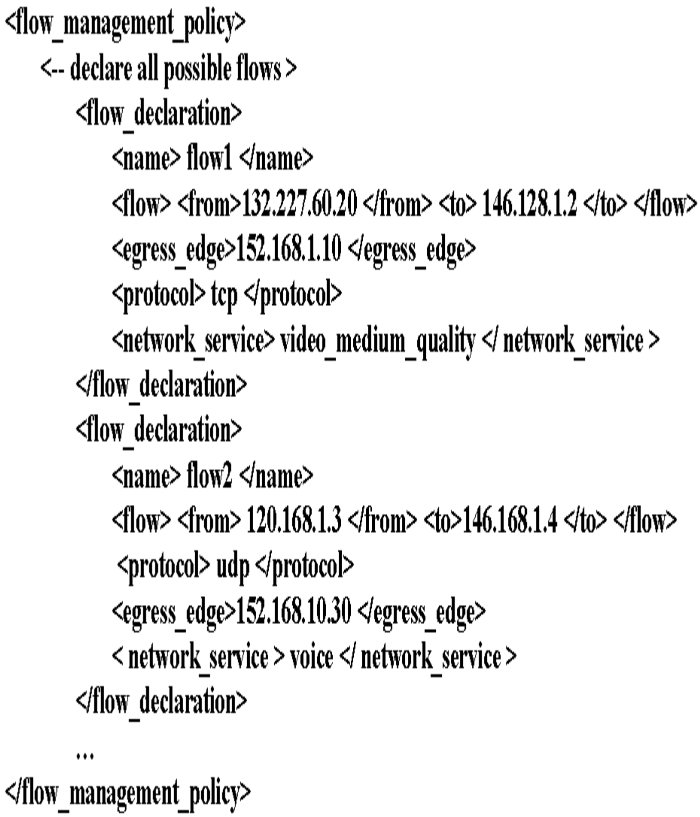


Fig. 2. Agent Architecture

4.1 The Agent Knowledge Base

The goal of the agent is to monitor the services provided by its edge router and cooperate with its peers to take corrective actions when a performance anomaly occurs. The agent knowledge base contains a description of all the flows entering the network by its edge router and a specification of the services offer by the network. A flow is characterized by the IP source address, the transport protocol, the destination port, the egress edge IP address and the network service used for its transport. Figure 3 shows the XML description of a flow.



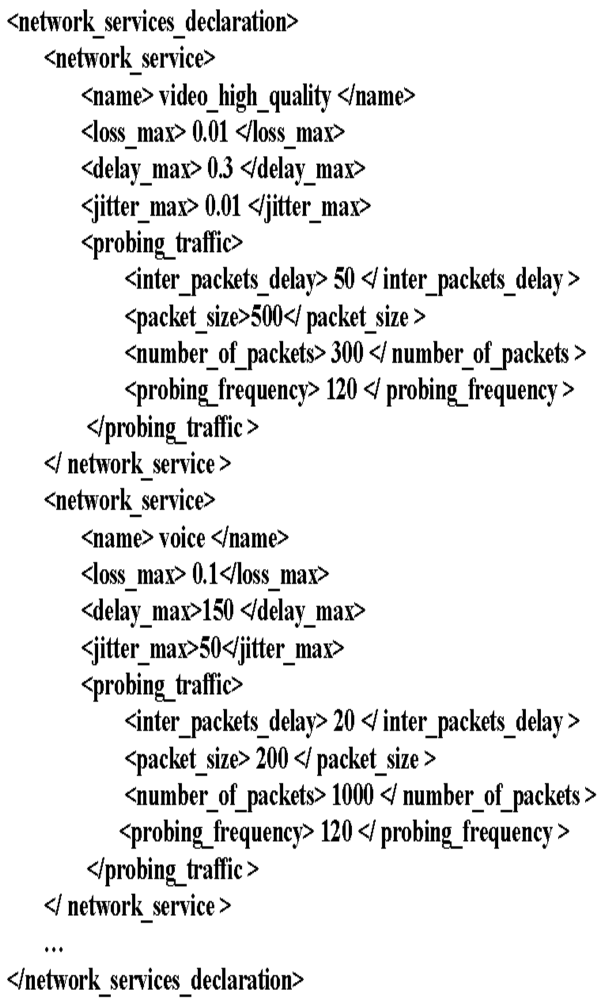
```
<flow_management_policy>
    <-- declare all possible flows >
        <flow_declaration>
            <name> flow1 </name>
            <flow> <from>132.227.60.20 </from> <to> 146.128.1.2 </to> </flow>
            <egress_edge>152.168.1.10 </egress_edge>
            <protocol> tcp </protocol>
            <network_service> video_medium_quality </ network_service >
        </flow_declaration>
        <flow_declaration>
            <name> flow2 </name>
            <flow> <from> 120.168.1.3 </from> <to>146.168.1.4 </to> </flow>
             <protocol> udp </protocol>
            <egress_edge>152.168.10.30 </egress_edge>
            < network_service > voice </ network_service >
        </flow_declaration>
        ...
</flow_management_policy>
```

Fig. 3. Example of XML description of a flow

The network service specification includes QoS requirements in terms of authorized maximum value of the delay, jitter and packets loss as well as the characterization of the traffic which will be used for probing. The idea is to send test traffic corresponding as much as possible to the behavior of the ingress traffics in order to have the most exact estimation of the network response time. On the basis of its probing results, the agent will evaluate if the network is able to deliver the awaited services. Figure 4 shows the XML description of network services.

4.2 The Situated Network View

The IP Network backbone is not a static entity. The network state is frequently prone to fluctuations due to the high variability of the traffics generated by the users' applications. It is thus very difficult to define a model of the network behavior. That is why the agent has to build its own representation of its environment. This operation is done thanks to the agent communication module. The agent receives from its peers their perception of the current network backbone quality of service. All those informations constitute the situated network view of the agent. They are crucial for the agent decision-making process.



```
<network_services_declaration>
    <network_service>
        <name> video_high_quality </name>
        <loss_max> 0.01 </loss_max>
        <delay_max> 0.3 </delay_max>
        <jitter_max> 0.01 </jitter_max>
        <probing_traffic>
            <inter_packets_delay> 50 </ inter_packets_delay >
            <packet_size>500</ packet_size >
            <number_of_packets> 300 </ number_of_packets >
            <probing_frequency> 120 </ probing_frequency >
         </probing_traffic >
    </ network_service >
    <network_service>
        <name> voice </name>
        <loss_max> 0.1</loss_max>
        <delay_max>150 </delay_max>
        <jitter_max>50</jitter_max>
        <probing_traffic>
            <inter_packets_delay> 20 </ inter_packets_delay >
            <packet_size> 200 </ packet_size >
            <number_of_packets> 1000 </ number_of_packets >
           <probing_frequency> 120 </ probing_frequency >
         </probing_traffic >
    </ network_service >
    ...
</network_services_declaration>
```

Fig. 4. Example of XML description of networks services

The knowledge base can be updated dynamically by a remote system.

4.3 The Agent Behaviors

Successively, the agent carries out two behaviors:

- the Service Probing behavior
- the Flow Control behavior

4.3.1 The Service Probing Behavior

The Service Probing behavior allows the monitoring of the QoS requirements for all the defined network services. The QoS monitoring involves the definition of metrics, measurement methodology and timing decision. The IETF IPPM defined a set of standard QoS and performance metrics and proposed measuring methodologies for them [12, 13]. For each service, the probe generates test traffics in direction of all the edge routers and computes the QoS metrics. This traffic is generated in conformity with the specifications contained in the network services declaration XML file. In this work, with an aim of simplification, we decide to restrict the QoS metrics to the end-to-end delay between edges routers. This end-to-end delay is computed by dividing the round trip delay by two. It is impossible without any additional information to know which direction generates the delay. It is certainly a coarse estimation of the one-way delay but it freed from the complexity of the tools necessary for an exact measurement. Hence, the measurement of the one-way delay requires the deployment of Network Time Protocol (NTP) or Global Positioning System (GPS) for the end-points clock synchronization. Once that the edge-to-edges delays are measured for each service, a local indicator for each service is then computed. The indicator varies between 0 and 1. The computing process of the local indicator i_S for service S is as follows:

```
Step one:

Extract the highest delay for the service S and the
agents involved in its measurement e.g measured delay
D, from Agent A to Agent B = (D,A,B) where D is the
highest delay measured for the service S

Step two: Indicator computation

if (measured_Delay<= Max_Athorized_Delay)

         $i_s$ = (measured_Delay / Max_Athorized_Delay)

else     $i_s$ = 1
```

When, for a service, the indicator $\boldsymbol{i_S = 0}$, nothing is wrong on it. Oppositely, when $\boldsymbol{i_S = 1}$, it implies that the flows using the service are in trouble. Something must be done to reduce to rectify the situation.

The computed indicators are broadcasted by the agents to their peers in the multi agent system.

4.3.2 The Flow Control Behavior

The flow control behavior consists in the decision-making process and the execution of the adequate actions in order to adapt the network traffic according to the available network resources and avoid the complete collapse of network services. The Flow Control behavior takes in input the set of indicators available in the network situated view and checks for each service if it is necessary to initiate a coordinated action. The possible actions are either authorizing traffic to enter the network backbone or prohibiting the access of a flow to it. This is realized thanks to the agent control module which acts directly on the routing functions of the router OS. For each service, the first step of the decision making process is to determine the edge router that appears more than the others in the destination field of the indicators. This indicates that the concerned destination is in serious trouble. If the greatest of these indicators exceeds a certain threshold, a message is sent to all the agents which some ingress flows are in direction to the previous detected destination. A decision must be taken to release the resource and avoid persistent congestion and performance degradation. It will be taken by one of the agents having received the message according to whether it is the smallest in the collating sequence, each agent being identified by an alphabetic letter.

5 Experimental Setup

A test-bed has been carried out to test our concepts and evaluate the relevance of our approach. Figure 5 shows the detailed setup. It consists of:

- 4 PC-based Linux routers
- 2 video servers. The VideoLAN software is used for the video streaming and viewing.
- 2 video clients

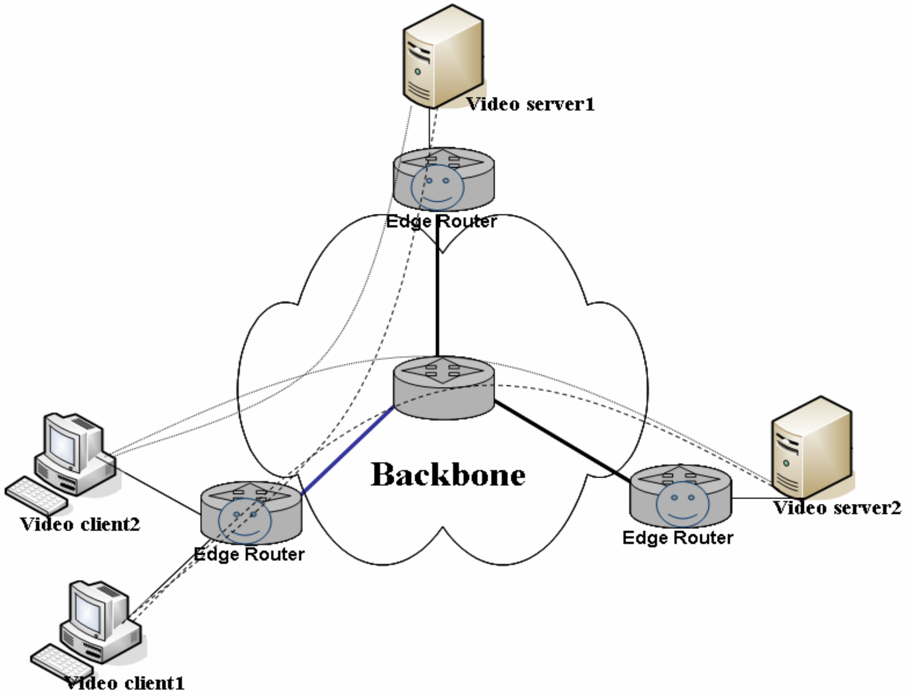


Fig. 5. Test bed

To simplify the tests, we considered that the network provides only one service, the video service. The authorized maximum delay for the video service is fixed at 500 ms. The agents are implemented in Java and the probe module in C. The test scenario is as follows:

2 video flows are streamed in loop from the servers to the clients on the udp protocol (figure 5). The incoming flows properties are declared in the knowledge base of the corresponding agent. Thus on each screen, it is possible to see two different videos. In a first stage, sufficient network resource is provided. All the links are point-to-point 100Mbit Ethernet. The videos have a perfect viewing quality. No degradation is observed on the images. Then, to simulate a performance problem on the network, the bandwidth of the link which connects the video clients' edge router to the backbone is reduced to 7.600Mbit. This is done thanks to the token bucket filter queue discipline of the Linux traffic controller module. With this bandwidth, the streams start to experience degradations. The video service is impacted by the lack of bandwidth. The images become fuzzy on the screens. The multi agent system is not yet activated. So, if nothing is done to reestablish the initial status, all the end-users' video applications will suffer of this performance degradation. Under these conditions, the agents are activated within the edge routers. The lack of sufficient network resources is detected by the system and an agent is elected to prohibit randomly the access to the network of one of its incoming flows. This has as a result the improvement of video service delivery and a good visual quality for the remaining streams. Figure 6 shows the delay variations of the video service between the video servers' edges routers and the customers' edge router some time before the agents' activation and after. We notice that before the agents' activation, the transfer delay of the video service is very high and oscillates between 700 ms and 1000 ms. Once the multi-agent system is activated, this delay falls and approaches zero. This is due to the

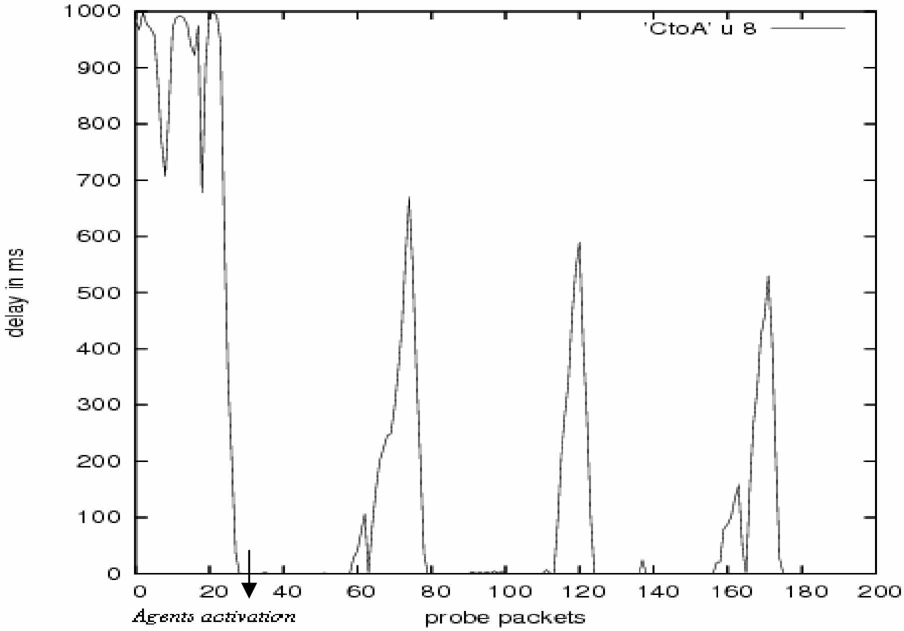


Fig. 6. Network delay variation

fact that one of the ingress streams is stopped and the shared resource is not any more congested. At the end of a variable time, since it depends on the network state, the system estimates that the anomaly having caused the performance degradation and the interruption of one of the flows is solved. This estimation is effective when the time series average of the ten last service indicator measurements is under the low threshold. The agents cooperate together and authorize suspended flows to reach again the network. Since the bandwidth limitation still exists, the system reacts and stops a video stream. This explains the jumps of delay observed thereafter.

6 Conclusion and Future Works

The unforeseeable breakdowns or anomalies when they occur in the network very often cause a considerable degradation of its quality of service. In these cases, all flows using the impacted services undergo the effects of the lack of sufficient network resource. That has as a consequence the possible violation of all the SLA of the customers using these services and possible financial losses for the ISP. To mitigate this perverse effect, it is necessary to implement a self-regulation system to control and regulate in near real time the amount of traffic to be admitted in the network according to the available resources. In this paper, we have specified a multi agent system in which every agent probes the network backbone and cooperates with its peers to finally authorize or prohibit the access to some ingress traffics. This allows more dynamicity in the control of network services and adaptive resource management while abstracting from network complexity and heterogeneity.

The practical experiments have permitted to prove the viability of our approach. Future works intend to extend our investigation to more complex scenario with several network services (voice, video, file transfer), estimate more precisely the impact of the test traffic on the network performance and evaluate the scalability of our solution.

References

1. R. Mahajan, S. Bellovin, S. Floyd, J. Loannidis, V. Paxson, S.Shenker, Controlling High Aggregate in the Networks, Technical report , February 2001
2. F. Krief, Self –aware management of IP networks with QoS guarantees, International Journal of Network Management 2004; 14: 351-364.
3. S. Willmott, M. Calisti, An Agent Future for Network Control? Revue des organisations suisses d'informatique, INFORMATIQUE 1/2000
4. P.Vilà J.L Marzo, E. Calle, Dynamic Bandwidth Management as part of an Integrated Network Management System based on Distributed Agents, Globecom 2002
5. L. J.Gash, Scalable autonomous monitoring and response for network communication reliabity, in proceedings of The 2004 Military Communications Conference, MILCOM '04, Monterey CA.
6. S. Jamin, P. B. Danzig, S. J. Shenker, and L. Zhang, "A measurement-based admission control algorithm for integrated services packet network", IEEE Trans. Networking., 5(1), Feb. 1997, pp. 56-70.

7. L. Breslau and S. Jamin, “Comments on the performance of measurement-based admission control algorithms”, IEEE Infocom 2000, Tel Aviv, Israel.
8. J. Qiu and E. W. Knightly, “Measurement-based admission control with aggregate traffic envelopes”, IEEE/ACM Trans. Networking, vol. 9, no. 2, April 2001.
9. S. Rito Lima, P. Carvalho, V. Freitas, Self-adaptive Distributed Management of QoS and SLSs in Multiservice Networks, 9th IEEE/IFIP International Symposium on Integrated Network Management (IM'2005) (Session 9) IEEE Press, Nice, France, May 15-19, 2005
10. K. Ficher, C. RuB, G. Vierke, Decision Theory and Coordination in Multiagent Systems, Reseach Report, September 98.
11. N. R. Jennings and M. A. Gibney, Dynamic Resource Allocation by MArket-Base Routing in Telecommunications Networks. In S. Albayrak and F. J. Garijo, editors, Proceedings Second International Workshop on Intelligent Agents for Telecommunications Applications IATA’98, pages 102-117. Springer (as LNAI-1437), 1998
12. G. Almes, S. Kalindindi, and M. Zekauskas, A One Way Delay Metric for IPPM, IETF RFC2679, 1999
13. IPPM-WG, IP Performance Measurements Working Group, http:/www.ietf.org/html.charters/ippm-charter.html.