CONFIGURABLE MULTI-AGENT SYSTEM FOR QOS CONTROL IN WATM

Youssef Iraqi[1], Raouf Boutaba[2] and Ahmed Mehaoua[3]

[1] University of Montreal, *DIRO*
C.P. 6128, Succ. A, Montreal, QC H3C 3J7, Canada
[2] University of Toronto, *ECE Department*
10 Kings College Road, Toronto, Ont. M5S 3G4, Canada
[3] University of Cambridge, *CCSR*
10 Downing Street, Cambridge, CB2 3DS, UK

Abstract

This paper introduces a configurable multi-agents architecture for QoS control in WATM. The ultimate aim of the proposed architecture is to provide a self-regulating network congestion control management by means of global network state awareness and agents interactions. The agents dynamically manage the buffer space at the level of a switch and interact to reduce the cell loss ratio while guaranteeing a bounded transit delay. We particularly address video transmission over UBR services using a per-VP queuing approach and an adaptive cell discarding congestion control scheme. Furthermore, a dynamic reconfiguration of the agents is performed during handoffs in order to continue meeting user end-to-end QoS requirements.

1. Introduction

Like Wired ATM, Wireless ATM (WATM) [1] aims at supporting multimedia communications which are characterised by variable connection bandwidth and QoS requirements. According to the nature of the multimedia traffic, some connections require low delay and delay variation while others require very low cell loss. Unlike wired ATM, Wireless ATM requires a handoff mechanism to support user mobility. The 'goodness' of a handoff impacts on the end-to-end QoS provided to users.

This paper introduces a configurable multi-agents architecture for QoS control in WATM. The architecture aims at self-regulating network congestion control management through global network state awareness and agents' interaction. The agents dynamically manage the buffer space at the level of a switch and interact to reduce the cell loss ratio while guaranteeing a bounded transit delay. We particularly address video transmission over UBR services using a per-VP queuing approach and an adaptive cell discarding congestion control scheme. A dynamic reconfiguration of the agents is performed during handoffs in order to continue meeting user QoS requirements.

This paper contains 5 sections. Section 2 presents the adopted congestion control scheme. Section 3 introduces the multi-agent architecture describing the two agent-levels structure as well as the agents' co-operation and communication protocols. Sections 4 and 5 detail the behaviour of the agents involved in the architecture. Section 6 discusses the (re-)configuration of the multi-agents system during handoffs.

2. A Congestion Control Scheme

The ATM network conveys multiple traffic classes associated with different quality of service requirements. This paper is based on a per-(class of service) per-VP queuing approach. Particularly, it addresses a per-VP queuing in UBR services.

Incoming cells belonging to best effort video connections are stored in the UBR output buffers (see Figure 1). Similarly, other incoming cells are stored in corresponding output queues depending on the service type. The portion of unused buffer space is considered as a shared buffer.

The question arises to how allocate this unused resource among the competing services. In the following, we assume that a simple scheduling policy is applied at every switch, which allows dynamic allocation of shared buffers to the different service classes.

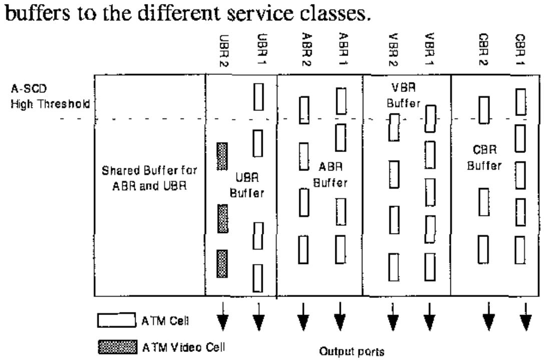

Figure 1: Switch Buffer Structure

This policy automatically allocate the buffer space to high priority CBR and VBR traffic and dynamically manage the remaining buffer space for best effort ABR and UBR traffic. This means that shared buffer space is flexibly allocated to VPs on an as-needed basis.

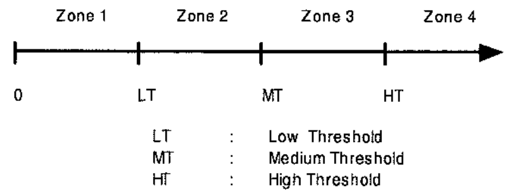

Figure 2: Buffer thresholds

The congestion control scheme used here and presented in [2] is based on a three-thresholds approach for managing buffer space allocated to UBR virtual paths as shown in figure 2. A constant method is applied to determine the values of the two other thresholds (e.g. Medium Threshold and Low Threshold). They are respectively set to 0.8 and 0.6 fraction of the High Threshold. The HT value is dynamically evaluated and set-up by the multi-agents system in order to reduce cell loss ratio while guaranteeing the end-to-end transit delay.

3. Multi-Agents Architecture for WATM Congestion Control Parameters regulation

The lack of most existing video QoS control frameworks is the use of static resource management and congestion control parameters. For instance, sources parameters (e.g. grouping mode, Drop tolerance) are negotiated at connection establishment time and cannot dynamically adjust to Quality of Service (QoS) variations. Similarly, switch parameters (e.g. buffer thresholds) are initialised for the virtual path life duration and don't take benefit of network load changes.

This static approach is not optimal and can be improved using an intelligent multi-agent system. The latter ensures self-regulating network management through global network state awareness and agents' interaction.

As presented in [3, 4], the primary task of these intelligent agents is to relieve the network operator from the adjustment of resource allocation and congestion control parameters (e.g. bandwidth usage control, buffer allocation, resource re-negotiation, etc).

An agent is a self-contained software element responsible for performing part of a programmatic process [5]. It contains some level of intelligence, ranging from simple predefined rules to self-learning artificial intelligence (AI) inference machines. It acts typically on behalf of a user or a process enabling task automation. Agents operate rather autonomously and may communicate with the user, system resources and other agents as required to perform their task. Moreover, more advanced agents may co-operate with other agents to carry out tasks beyond the capability of a single agent.

Among the action to be performed by the agents is the continuous monitoring of network state and the use of this knowledge to make decisions based on predefined rules and policies and with respect to user goals.

This work emphasises the automatic adjustment of the A-SCD [2] parameters (e.g. Thresholds, Drop Tolerance) to ensure a low cell loss ratio with a bounded end-to-end cell transfer delay in a WATM network.

3.1 Multi-Agents Architecture

Let us define a Managed Domain (MD) as the association of two adjacent ATM switches along the virtual path connection (VPC). Each Managed Domain is under control of a high level intelligent agent (IA), referred to as the Domain Agent (DA).

As depicted in Figure 3, the lower architecture layer is controlled by a set of intelligent agents, referred to as Switch Agent (SA). SAs are located at every ATM switch and monitor the switch behaviour (e.g. buffer queue length). They automatically adjust appropriate thresholds depending on directives received from the DA. Since SAs have partial knowledge of the system (e.g. VPC), they only act on behalf of the DA to collect and filter pertinent state information. This delegation of performance management results in a minimum control information exchange within a specific Managed Domain.

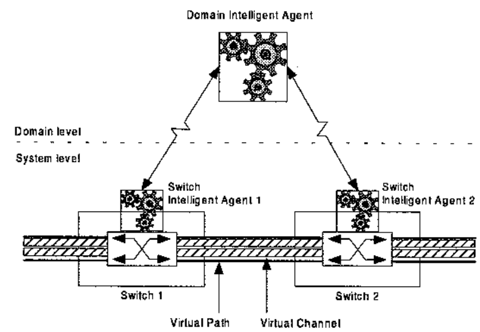

Figure 3: A Managed Domain

SAs are responsible of the syntactical aspects of the management information (e.g. collection, and representation), while the DA focuses on the semantic aspects (intelligent processing, decision, etc).

A DA aggregates information reported by its subordinate SAs. Since it embeds a wider knowledge of the system's state, it is able to make management decisions leading to the invocation of management actions to be executed by the underlying SAs.

At the system level, an SA may execute its tasks totally decoupled from its neighbours. By using such a multi-level agent architecture, we have divided the global space into domains with manageable complexity. The induced partial knowledge faced by the system level is compensated by the reactivity/responsiveness of the overall control. Indeed, to ensure accurate and efficient control/management decisions, reactions have to be in the order of magnitude of cell switching. To meet this temporal requirement, inter-agent distances and control data amount should be as small as possible. For example, the DA is physically co-located with one of the

two SA partners. Finally, the proposed architecture is sufficiently generic and system-independent to be extended to a higher number of abstraction levels.
In terms of agents activities, we will focus in this paper on the modification of the switch output buffer thresholds and the reconfiguration process of the control agents during a handoff.

3.2 Intelligent Agents Co-operation and Communication Operations

To support agents' co-operation and communication operations, we use ATM Operation And Management (OAM) flows. Three types of OAM cells are available at the ATM layer which are differentiated by the performed function: activation/deactivation, fault management and performance management [6]. The role of fault management cells is to monitor and to test virtual connections (VPC and VCC). Performance management cells are used to monitor the performance of VPCs/VCCs and report the collected performance data such as erroneous and lost cells. The activation/deactivation function performs monitoring and continuity checking of connections.
OAM cells can be routed at the virtual path (F4) or virtual channel level (F5). OAM cells of type F4 use the same virtual path (e.g. VPI) than user cells, but a separate virtual channel (e.g. VCI). The OAM cells of type F5 are carried in the same virtual path and channel than user cells. F4 and F5 OAM cells flow between endpoints or only on a segment of a connection depending on the value of the VCI field and the PTI field respectively.
In this paper, control informations between SAs and DA are carried by F4 OAM segment flow cells. More precisely, using the F4 performance management OAM cells with monitoring/reporting function type.
To collect the relevant management information, a DA inserts periodically (e.g. every fixed time interval T) a F4 OAM cell, which is looped back at switch agent within a single domain (Fig.4).

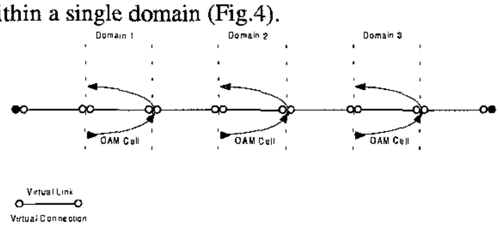

Figure 4: IA Information Exchange Mode

The length of the domain measurement interval 'T' determines the accuracy and the variance of the measures. Indeed, longer intervals provide lower variance but result in slower updating information. Alternatively, shorter intervals allow fast response but introduce greater variance in the response. The determination of the T value is out of the scope of this study. Nevertheless, a T parameter in the order of magnitude of the Round Trip Time (RTT) may be suitable and will be investigated in further work.

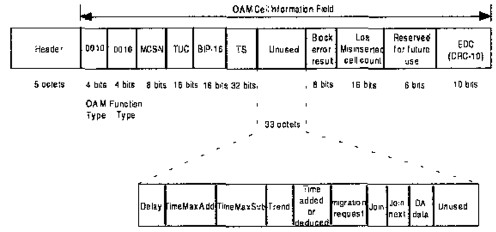

Figure 5: OAM F4 cell structure

Figure 5 shows the structure of the OAM F4 cell. The unused field contains the four parameters sent by the SA to the DA and the time sent by this latter to the SA, as well as other variables which will be introduced in section 6.1.
In order for the cells to be processed consistently at the level of the two switches, it is always the SA that is upstream of the video flow that starts making the changes. Then, it sends an OAM cell that acts as a heading for all the following user cells. Once the OAM cell arrives at the second switch, the changes are reflected on the following cells by setting the new threshold values.
According to the information flow direction, two cases are possible and are shown in figure 6.
In figure 6.a, the SA1 updates its variables and the DA sends an OAM cell to the SA2. In figure 6.b, the DA sends an OAM cell to the SA1. On reception of the OAM cell, the SA1 applies the necessary modifications and notifies its partner with an OAM cell in order for this one to change its parameters.

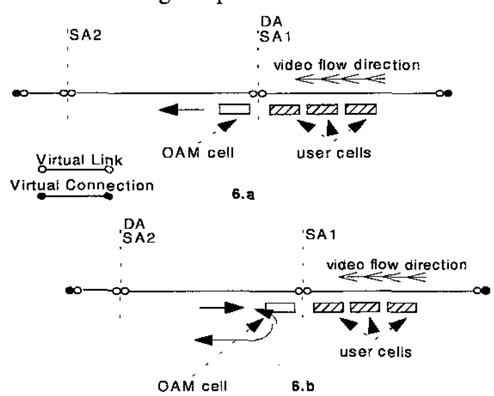

Figure 6: SA & DA communication according to information flows

4. Switch Agent Behaviour

4.1 Switch Agent Parameters

The SA maintains the following resource and control parameters: The average queue size (L) (Fig.7); the available space in the shared buffer (F); the output port rate; and the high threshold (HT).
Each SA send to its DA the following information: The mean service delay (DELAY); the maximum delay to

add at this switch (TIME_MAX_ADD) which depends on the available space in the shared buffer queue; and the maximum delay to subtract at the switch (TIME_MAX_SUB). It depends on the High Threshold (HT) and the average queue size (L) and the load trend of the switch (TREND). The first three variables are transmitted to the DA expressed in terms of temporal units (e.g. time) in order to have a homogeneous vision of the state of the two switches. This choice is justified by the fact that the switches may have different output rates with a different cell service delay. Therefore the mean queue length parameter is not sufficient to allow the DA to make accurate decisions.

4.2 Switch Agent Operations / Policies

Periodically, each SA calculates and inserts the following information into the OAM F4 cell at the destination of the DA:

- DELAY: The mean service delay experienced by the cells is calculated using the mean queue length and the output port rate.
- TIME_MAX_ADD: The DA may ask a SA to increase its High Threshold. It has to know the available space in the shared buffer. The SA has to express this amount in terms of temporal units, calculated as follows: γ*(number of available cell slots / output rate). γ represents the percentage of the shared buffer (F) that the switch is allowed to use (policy P.1).
- TIME_MAX_SUB: The DA can ask a SA to decrease the size of its High Threshold. It has to be aware of the available space in the switch buffer (R). The agent on the switch has also to express this information in terms of temporal units, calculated as follows: α*[(High Threshold - Average queue size) / output rate]. α represents the percentage of R (Fig.7) that the switch can free (policy P.2).
- TREND: This parameter represents the trend of the R variable (Fig.7). If R increases the load of the switch decreases. Conversely, if R decreases then the load of the switch increases. This value is computed as follows: TREND = $-\partial R / \partial t$.

The SA reconfiguration operations during handoff will be introduced in section 6.2.

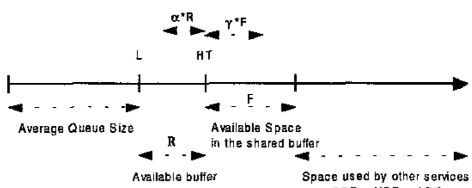

Figure 7: Switch Buffer Parameters

The management policies of the SA are:

P.1 SA can only use a percentage γ of the available space not used by the other services in addition to the maximum size allocated to the UBR service

P.2 DA can decrease the size of the switch buffer by a percentage α of the available buffer space only.

4.3 Switch Agent Pseudo Code

```
Send()
 Repeat
 {Delay = Average_Queue_Size / Output_Rate
          /* The average service delay in the switch*/
  TimeMaxAdd = γ* (nbr_free_cell / Output_Rate)
          /* The maximum time that the domain-agent can add to this
          switch */
  if Average_Queue_Size < Hight_Threshold
          then TimeMaxSub = α* [(Hight_Threshold -
          Average_Queue_Size)/ Output_Rate]
          /* The maximum time that the domain-agent can
          subtract from this switch */
          else TimeMaxSub = 0
  trend = -∂R/∂t
 }
Receive()
 Repeat
 {At reception of t do
          /* t is the time sent by the domain agent */
  Hight_Threshold = Hight_Threshold + t * Output_Rate
 }
```

5. Domain Agent Behaviour

5.1 Domain Agent Policies

Each DA maintains the following information: Current mean transit delay of the switches; the maximum delay to add to each switch; the maximum delay to subtract to each switch; and the switch load trend. The DA distributes the delay between the two switches to realise the minimum loss rate while guaranteeing the same global delay for the two switches. This is achieved by decreasing the High Threshold of the less loaded switch and increasing the High Threshold of the higher loaded switch.

The DA applies the following management policies:

P.3 The DA distributes the time credit between the switches in a pondered manner.

P.4 The most loaded switch will receive more credits than the other. If necessary, HT will be decreased to maintain the same global transit delay.

P.5 The global delay distributed between the two switches has always to be bounded.

The DA receives four parameters, from each SA, used to make a decision about time distribution between the two switches. It reduces the time allocated to the less loaded switch (S1) and adds it the most loaded switch (S2). To avoid discarding the cells already in S1 buffer, the deduced value cannot exceed TIME_MAX_SUB of S1. The added value cannot exceed TIME_MAX_ADD of S2 according to space spool allocation policy (policies P.1 and P.2).

When the two switches are loaded and the global delay does not exceed the maximum allowed delay, the DA distributes the remaining time period to the two switches (this way, increasing their buffer size). This allows the switches to decrease their cell loss rates. Such distribution of the available time is made in a pondered manner (policy P.3) according to S1 and S2 constraints (TIME_MAX_ADD and TIME_MAX_SUB).

5.2 Domain Agent Pseudo Code

```
Repeat
 {if Delay ≥ Delay_1 + Delay_2 then
  {case
      • trend_1 < 0 and trend_2 > 0 do
        {add to Switch_2 Min(TimeMaxAdd_2, TimeMaxSub_1)
        sub to Switch_1 Min(TimeMaxAdd_2, TimeMaxSub_1) }
        /* policy P.4 */
      • trend_1 > 0 and trend_2 < 0 do
        { add to Switch_1 Min(TimeMaxAdd_1, TimeMaxSub_2)
        sub to Switch_2 Min(TimeMaxAdd_1, TimeMaxSub_2) }
        /* policy P.4 */
        • trend_1 > 0 and trend_2 > 0 do
        {t=(Delay - Delay_1 - Delay_2)
              /* t is the available time */
        add to Switch_1 Min[TimeMaxAdd_1, Max(β*t, t-
    TimeMaxAdd_2)]
        add to Switch_2 Min[TimeMaxAdd2, Max((1-β)*t,t-
    timeMaxAdd_1)]}
        /* policy P.3, 0≤β≤1 */
          }
   }
  }
```

6. Multi-Agents System Configuration

6.1 Initial configuration

After the connection set-up phase, a SA process is launched on every switch. Then, the first SA in the direction of the information flow will send a *join* message (an OAM cell with the join bit set, see fig.5) to its next neighbour requesting it to be its partner. When receiving the *join* message, a DA process is launched on the switch and a *join-next* message (an OAM cell with the join-next bit set, see fig.5) is sent to the next switch. This later will act as the first SA by sending a *join* message to its upstream neighbour.

Handoff in a Wireless ATM network require changes in virtual connections. Several handoff schemes have been proposed, such as connection extension [7], full re-establishment [8], partial re-establishment [8], multicast join/leave [8] and multicast group [9]. This will require a reconfiguration of the multi-agent system. The next section presents the agents reconfiguration process in case of a partial re-establishment handoff.

6.2 Reconfiguration during handoff

In the partial re-establishment handoff scheme, a new path is established from the new base station to a node in the original connection path. Hence, this scheme requires the discovery of the crossover switch (COS), the setting up of the new partial path and the tearing down of the old partial path.

Crossover switch discovery is the process of locating a suitable COS so that a new partial path can be established from the new base station (BS) to this COS. In [10] five COS discovery schemes have been proposed and evaluated, which are: *loose select, prior path knowledge, prior path resultant optimal, distributed hunt and backward tracking.*

When a handoff occurs the agents must be reconfigured to continue meeting user requirements in terms of transit delay and cell loss ratio (see fig. 8).

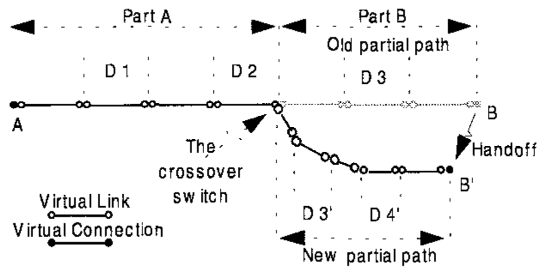

Figure 8: handoff with partial re-establishment

As SAs work by pairs of agents (see section 4.1), there are three possible configurations:

(1) If the COS agent's partner belongs to the part A of the connection (see fig.8), then no reconfiguration is needed. The COS will establish the new partial path with the new delay constraint (the maximum allowed transit delay minus the maximum delay across part A of the connection) and send a *join-next* message towards the new adjacent switch.

(2) If the COS agent's partner belongs to the part B of the connection and the DA is on the COS (see fig.9), then the COS will establish the new partial path and the DA will initialise the new COS agent's partner. The DA will send it a *join* message requesting it to be the new partner of the COS agent.

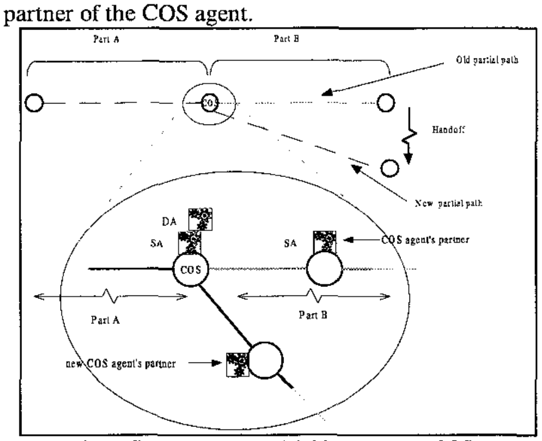

Figure 9: configuration 2, the DA initialise the new COS agent's partner

(3) If the COS agent's partner belongs to the part B of the connection and the DA is on the COS partner switch (see fig.10), the COS agent will send a *migrate* message to its partner to migrate the DA towards the COS. Then, the COS agent will proceed like in configuration 2.

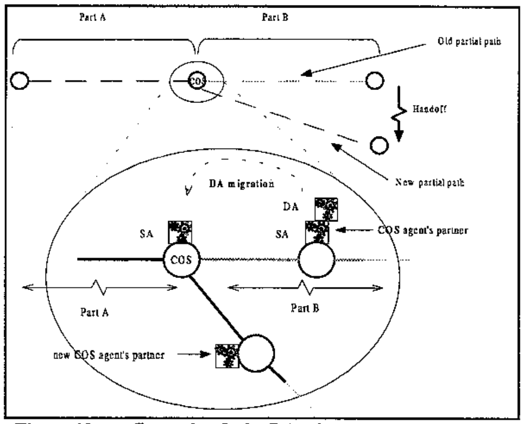

Figure 10: configuration 3, the DA migrate towards the COS

To migrate DA, the COS will send an OAM cell with the *migration request* bit set. The COS agent's old partner will respond with an OAM cell containing the DA data. The COS agent will then launch a new DA process with the data received. This assumes that the code of the two types of agents, namely SA and DA, are located on every switch. However, only one DA process is executing at a given time for each pair of partner switches. After the establishment of the new partial path, the DA will send a *join* message to the new partner.

6.3 Handoff delay absorption

The handoff process introduces a delay that must be absorbed in order to continue meeting user requirements negotiated at the first setup. The COS must take into consideration this delay while setting up the new partial path. In this perspective, two cases are possible according to the flow direction.

In the case where the transmission flow is from A to B two situations are possible:

(1) All the handoff delay is already absorbed by part A of the connection. In this case, the COS will establish the new partial path with no further delay constraints (the overall connection must respect the initial delay constraint).

(2) Only part of the handoff delay was absorbed by part A of the connection. In this case, the new partial path must absorb the remaining delay, so the COS will try to add this constraint during the setup phase of the new partial path.

In the case where the transmission flow is from B to A, then the new partial path must absorb all the handoff delay.

Once the handoff delay is absorbed, initial delay constraints are observed by the multi-agent system.

7. Conclusion

This paper has presented a multi-agent architecture for congestion control in WATM. The agents operation relies on an adaptive cell-discarding algorithm presented in [2]. However, the proposed architecture is generic in that it can support different congestion control schemes. The agents dynamically manage the buffer space at the level of a switch and interact to reduce the cell loss ratio while guaranteeing a bounded transit delay. This approach is more efficient than classical static video QoS control frameworks as it provides a self-regulating network control management by means of global network state awareness and agents co-operation. Moreover, as mobile WATM networks are characterised by the occurrence of handoffs, the proposed multi-agents system integrates a dynamic reconfiguration capability. The latter allows the multi-agents system to continue performing its task over the newly set up connection and thus continue guaranteeing end-user QoS requirements.

8. References

[1] K. Rauhala, Ed., 'Baseline Text for Wireless ATM Specification, ATM-Forum, BTD-WATM-01.03, July 1997.

[2] A. Mehaoua, R. Boutaba and G. Pujolle, 'An Extended Priority Data Partition Scheme for MPEG Video Connections over ATM', 2nd IEEE ISCC'97, Alexandria, Egypt, June 1997, pp. 62-67.

[3] A. Schuhknet, G. Dreo, 'Preventing Rather Repairing: A new Approach in ATM Network Management', INET'95 HyperMedia, Honolulu, Hawaï, June 1995.

[4] T. Zhang, S. Covaci, R. Popescu-Zeletin, 'Intelligent Agents in Network and Service Management', IEEE GLOBECOM'97, London, 1997, pp.1855-1861.

[5] T. Magedanz, K. Rothermel, S. Krause, 'Intelligent Agents : An Emerging Technology for Next Generation Telecommunications', IEEE INFOCOM'96, SF (CA), USA, March 1996.

[6] ITU-T I.610, 'B-ISDN Operation and Maintenance Principles Functions', Geneva, March 1993.

[7] P. P. Mishra and M. B. Srivastava. 'Call Establishment and Re-routing in Mobile Computing Networks', Technical report, AT&T Bell Labs. TM 11384-940906-13, Sept. 1994.

[8] K. Keeton, B. Mah, S. Seshan, R. Katz, and D. Ferrari, 'Providing Connection-Oriented Network Services to Mobile Hosts', In USENIX Symposium on Mobile and Location Independent Computing, Aug. 93, pp. 83-102.

[9] R. Ghai and S. Singh, 'A Protocol for Seamless Communication in a Picocellular Network', In Proc. IEEE Supercomm/ICC, May 1994, pp. 192-196.

[10] C. K. Toh, 'Crossover Switch Discovery for Wireless ATM LANs', ACM Journal on Mobile Networks and Applications–Special Issue on Routing in Mobile Communication Networks, 1(2), Oct. 96, pp. 141-165.