# QoS control in wireless ATM

Youssef Iraqi [a], Raouf Boutaba [b] and Alberto Leon-Garcia [c]

[a] *University of Montreal, DIRO, C.P. 6128, Succ. A, Montreal, QC, Canada H3C 3J7*
[b] *University of Waterloo, DECE, Waterloo, Ont., Canada N2L 3G1*
[c] *University of Toronto, DECE, 10 Kings College Road, Toronto, Ont., Canada M5S 3G4*

This paper introduces a 3-level multiagent 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 agent 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. The handoff delay absorption is also addressed.

## 1. Introduction

Like wired ATM, Wireless ATM (WATM) [8] 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. This mechanism introduces another level of difficulty in providing QoS services.

This paper introduces a 3-level multiagent architecture for QoS control in WATM. The architecture aims at self-regulating network congestion control management through global network state awareness and agent 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 queueing approach and an adaptive cell discarding congestion control scheme. A dynamic reconfiguration of the agents with a handoff delay absorption process is performed during handoffs in order to continue meeting user QoS requirements.

This paper contains 7 sections. Section 2 presents the adopted congestion control scheme. Section 3 introduces the multiagent architecture describing the three agent-level structure as well as the agents' cooperation and communication protocols. Sections 4–6 detail the behaviour of the agents involved in the architecture. Section 7 discusses the (re-)configuration of the multiagents system during handoffs as well as the handoff delay absorption process.

## 2. A congestion control scheme

The ATM network supports 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 per-VP queuing in UBR services.

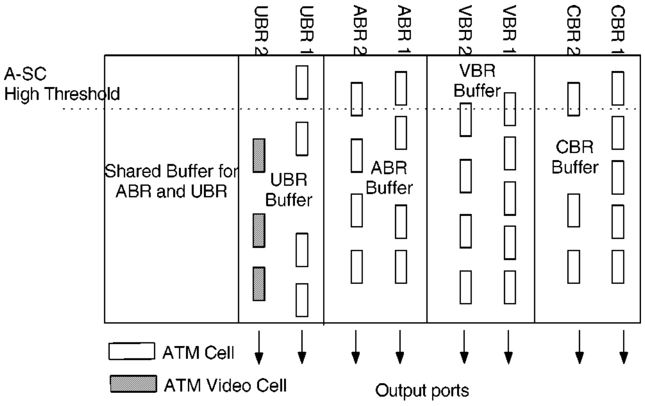

Figure 1. Switch buffer structure.

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 how to 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.

This policy automatically allocates the buffer space to high priority CBR and VBR traffic and dynamically manages 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.

The congestion control scheme used here and presented in [6] is based on a three-threshold 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

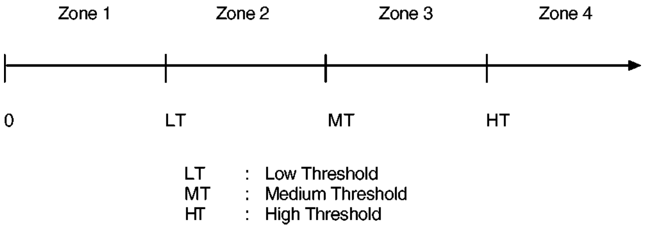

Figure 2. Buffer thresholds.

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 multiagent system in order to reduce cell loss ratio while guaranteeing the end-to-end transit delay.

## 3. Multiagent architecture for WATM congestion control parameter regulation

Most existing video QoS control frameworks lack the use of static resource management and congestion control parameters. For instance, source 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 do not take benefit of network load changes.

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

As presented in [9,11], 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 renegotiation, 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 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 cooperate with other agents to carry out tasks beyond the capability of a single agent.

Among the actions to be performed by the agents is the continuous monitoring of the network state and the use of this knowledge to make decisions based on predefined rules and policies and with respect to user goals. These are achieved by the monitoring of network resources and quality parameters associated with each service class. In this article, we emphasise the automatic adjustment of the A-SCD [6] parameters (e.g., thresholds, drop tolerance) in order to ensure a low cell loss ratio with a bounded end-to-end cell transfer delay in a WATM network.

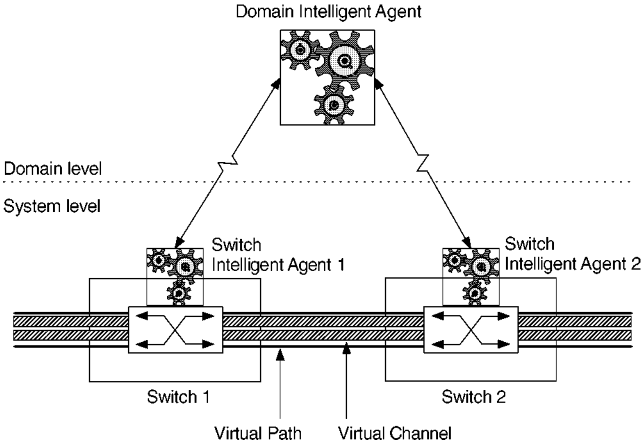

Figure 3. A managed domain.

### 3.1. Multiagent 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) [3]. In this paper, a Super Agent (SupA) layer is introduced to control all the DAs and, hence, provide an end-to-end QoS control.

As depicted in figure 3, the lower architecture layer is controlled by a set of intelligent agents, referenced as Switch Agent (SA). These autonomous agents are located at every ATM switch. The aims of SA are the monitoring of the network component behaviour (e.g., buffer queue length) and the automatic adjustment of the related thresholds depending on directives coming from the upper domain IA. Since these SAs have partial knowledge of the controlled 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.

SAs are responsible for the syntactical aspects of the management information (e.g., collection, and representation), while the DAs and the SupA focus on the semantic aspects (intelligent processing, decision-making, etc.). The task of a SupA is to aggregate the information reported by its subordinate DAs. Since it embeds an end-to-end knowledge of the state of the system under control, it is able to make management decisions leading to the invocations of management actions executed by the underlying intelligent agents (see figure 4).

At the system level, an SA may execute its tasks totally decoupled from its neighbours. By using such a multilevel 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, intera-

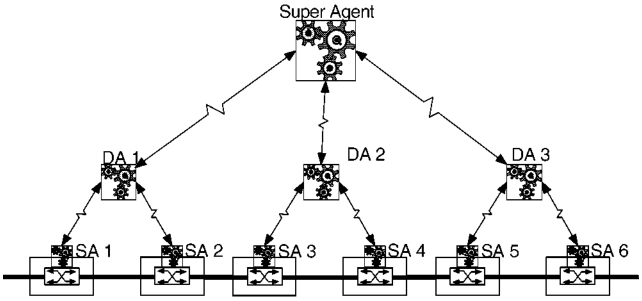

Figure 4. The overall architecture.

gent distances and control data amount should be as small as possible. For example, the DA is physically collocated 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 agent 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 as well as the handoff delay absorption problem.

### 3.2. Intelligent agents cooperation and communication operations

To support agent cooperation 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 [1]. 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) as 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 as user cells. F4 and F5 OAM cells flow between endpoints or only on a segment of a connection depending on the values of the VCI field and the PTI field, respectively.

In this paper, we propose to carry control information between SAs, DAs and the super agent through F4 OAM segment flow cells. More precisely, using the F4 performance management OAM cells with monitoring/reporting function type. We suppose that these special OAM cells are never dropped by the congestion control scheme.

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 (figure 5).

To collect the relevant management information, the SupA also inserts periodically an F4 OAM cell (with the SupA/DA bit set (see figure 7)), but this special cell is looped back at endpoints (figure 6).

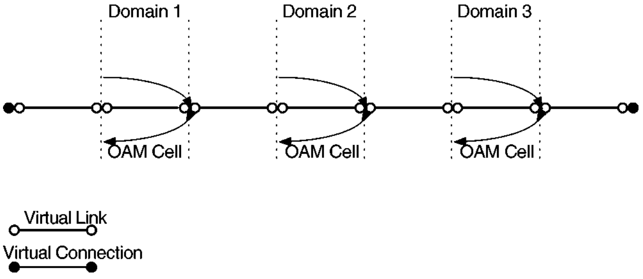

Figure 5. IA information exchange mode.

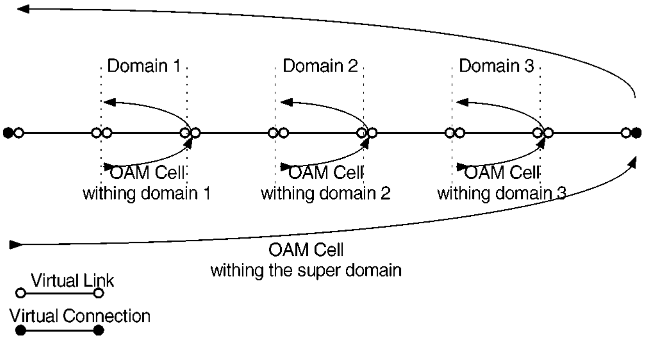

Figure 6. Super agent information exchange mode.

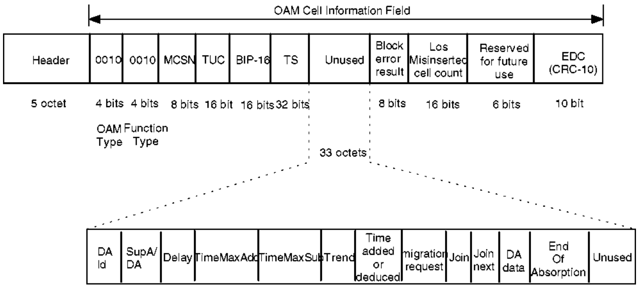

Figure 7. OAM F4 cell structure.

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 7 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 7.1.

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

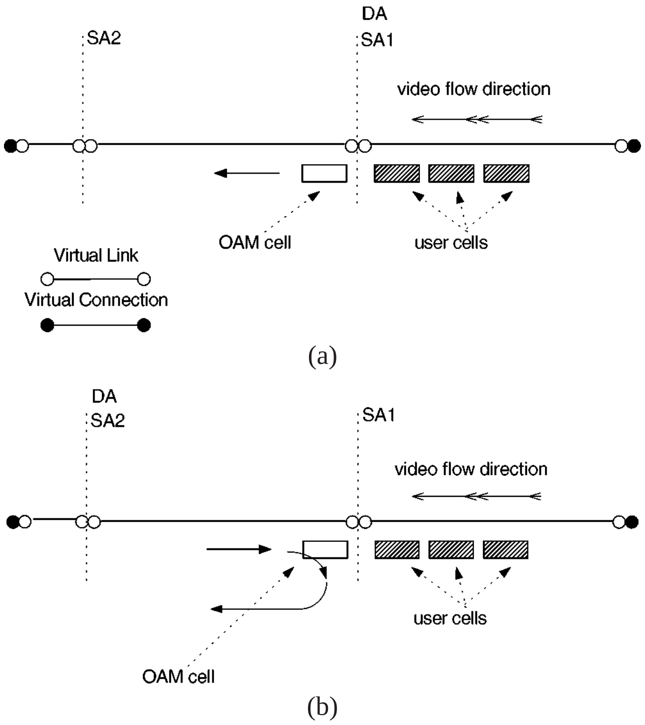

Figure 8. SA and DA communication according to information flows.

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 8. In figure 8(a), the SA1 updates its variables and the DA sends an OAM cell to the SA2. In figure 8(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.

## 4. Switch agent behaviour

### *4.1. Switch agent parameters*

The SA maintains the following resource and control parameters: the available space in the shared buffer (F)(see figure 9); the output port rate; and the high threshold (HT). Each SA sends to its DA the following information: The maximum 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 this 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 maximum queue length parameter is not sufficient to allow the DA to make accurate decisions.

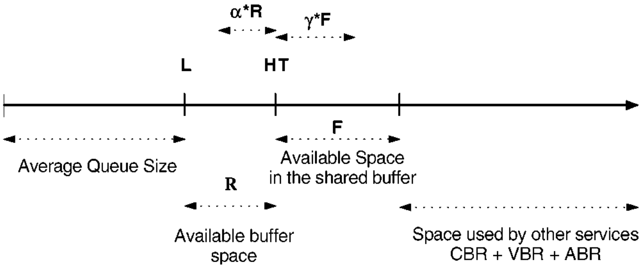

Figure 9. Switch buffer parameters.

### *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 (with the DA id field set correctly (see figure 7)):

- DELAY: The maximum service delay experienced by the cells is calculated using the maximum queue length (HT) and the output port rate.
- TIME_MAX_ADD: The DA may ask an 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:

  $$\gamma \times (\textit{Number of Available Cell Slots}/\textit{Output Rate}).$$

  Here $\gamma$ 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 an 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 to also express this information in terms of temporal units, calculated as follows:

  $$\alpha \times \left[\frac{(\textit{High Threshold} - \textit{Average Queue Size})}{\textit{Output Rate}}\right].$$

  Here $\alpha$ represents the percentage of $R$ (figure 9) that the switch can free (policy P.2).
- TREND: This parameter represents the trend of the $R$ variable (figure 9). 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:

  $$\textit{TREND} = -\frac{\partial R}{\partial t}.$$

The SA reconfiguration operations during handoff will be introduced in section 7.2. The management policies of the SA are:

P.1. SA can only use a percentage $\gamma$ 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 $\alpha$ of the available buffer space only.

### 4.3. Switch agent pseudo code

```
Send()
  Repeat {
    $Delay = Maximum\_Queue\_Size / Output\_Rate$
      /* The maximum service delay in the switch*/
    $Time\_Max\_Add = \gamma \times (Nbr\_Free\_Cell / Output\_Rate)$
      /* The maximum time that the domain-agent can
        add to this switch */
    if $Average\_Queue\_Size < High\_Threshold$
      then $Time\_Max\_Sub = \alpha \times [(High\_Threshold -$
          $Average\_Queue\_Size) / Output\_Rate]$
      /* The maximum time that the domain-agent can
        subtract from this switch */
      else $Time\_Max\_Sub = 0$
    $trend = -\partial R / \partial t$}
Receive()
  Repeat {
    At reception of $t$ do
      /* $t$ is the time sent by the domain agent */
      $High\_Threshold = High\_Threshold$
        $+ t \times Output\_Rate$}
```

## 5. Domain agent behaviour

### 5.1. Domain agent parameters

Each DA sends to the SupA the following information: The maximum service delay within the domain (DELAY); the maximum delay to add at this domain (TIME_MAX_ADD) which depends on the available space in the shared buffer queue of the two switches controlled by the DA; and the maximum delay to subtract at the domain (TIME_MAX_SUB).

Periodically, each DA calculates and inserts the following information into the OAM F4 cell at the destination of the SupA (with the DA id field set correctly (see figure 7)):

- DELAY: The maximum service delay experienced by the cells within the domain is calculated using the maximum service delay for the two switches.
- TIME_MAX_ADD: The SupA may ask a DA to increase its allowed maximum delay. The DA calculates this parameter as follows:

$$TIME\_MAX\_ADD\_1 + TIME\_MAX\_ADD\_2.$$

- TIME_MAX_SUB: The SupA can ask a DA to decrease its allowed maximum delay. The DA calculates these value as follows:

$$TIME\_MAX\_SUB\_1 + TIME\_MAX\_SUB\_2.$$

- TREND: This parameter represents the domain load trend and depends on the controlled switches trends.

### 5.2. Domain agent operations and policies

Each DA maintains the following information: Current maximum transit delay of the switches; the maximum delay to add to each switch; the maximum delay to subtract from each switch; and the switches' 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 to be always bounded.

The DA receives four parameters, from each SA, and uses these parameters 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 to the most loaded switch (S2). To avoid discarding the cells already in the 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).

When the two switches are not loaded and the global delay is below the maximum allowed delay (for the domain), the remaining time credits can be used by the super agent. The super agent can give those time credits to another loaded domain.

The domain agent will process only the OAM cells with its own DA Id. If the SupA/DA bit is set, this means that the OAM cell was sent by the super agent, if not then the OAM cell contains information concerning its SAs.

### 5.3. Domain agent pseudo code

```
Repeat()
  {if $Delay \geqslant Delay\_1 + Delay\_2$ 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 */
    $\mathit{trend\_1} > 0$ and $\mathit{trend\_2} < 0$ do
      add to Switch_1 $\mathit{Min}(\mathit{TimeMaxAdd\_1}, \mathit{TimeMaxSub\_2})$
      sub to Switch_2 $\mathit{Min}(\mathit{TimeMaxAdd\_1}, \mathit{TimeMaxSub\_2})$
      /* policy P.4 */
    $\mathit{trend\_1} > 0$ and $\mathit{trend\_2} > 0$ do
      $t = (\mathit{Delay} - \mathit{Delay\_1} - \mathit{Delay\_2})$
      /* $t$ is the available time */
      add to Switch_1 $\mathit{Min}[\mathit{TimeMaxAdd\_1}, \mathit{Max}(\beta \times t, t - \mathit{TimeMaxAdd\_2})]$
      add to Switch_2 $\mathit{Min}[\mathit{TimeMaxAdd\_2}, \mathit{Max}((1 - \beta) \times t, t - \mathit{TimeMaxAdd\_1})]\}$
      /* policy P.3, $0 \leqslant \beta \leqslant 1$ */
  $\mathit{TimeMaxAdd} = \mathit{TimeMaxAdd\_1} + \mathit{TimeMaxAdd\_2}$
  $\mathit{TimeMaxSub} = \mathit{TimeMaxSub\_1} + \mathit{TimeMaxSub\_2}$
  $\mathit{Delay} = \mathit{Delay\_1} + \mathit{Delay\_2} + d$
  $\mathit{Trend} = \mathit{Trend\_1} + \mathit{Trend\_2}$
  /* $d$ is the transmission delay between the two switches */
  Send these values to the SupA when requested}
```

## 6. Super agent behaviour

### 6.1. Super agent policies

The SupA distributes the delay between the domains to realise the minimum loss rate while guaranteeing the same end-to-end delay. This is achieved by decreasing time credits of the less loaded domains and increasing the time credits of the higher loaded domain.

The SupA applies the following management policies:

P.6. The SupA distributes the time credit between the domains in a pondered manner.

P.7. The most loaded domain will receive more credits than the others.

P.8. The end-to-end delay distributed between the domains has always to be bounded.

The SupA receives four parameters, from each DA, and uses them to make a decision about time distribution between the domains. It reduces the time allocated to the less loaded domain and adds it to the most loaded domains.

The super agent can implement its management policies in many ways ranging from simple rules to inference intelligent system. This choice of implementation is out of the scope of this paper.

The super agent distributes also the handoff delay once this later is absorbed as will be explained in section 7.3.

### 6.2. Super agent pseudo code

```
Repeat()
  for each domain $D$
    send an OAM cell with the DA id set correctly
    receive information from DA
  intelligent processing and decision according to the management policies
  send response to each DA
```

## 7. Multiagent system configuration

### 7.1. Initial configuration

After the connection set-up phase, an 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 figure 7) 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 figure 7) is sent to the next switch. This later will act as the first SA by sending a join message to its upstream neighbour.

The task of the super agent can be taken by one of the switches, for example the first or the last. The DAs can also elect one special agent for this task. This process is out of the scope of this study. Nevertheless, a switch which is most likely to remain in the connection (i.e., not affected by the handoff process) may be a suitable choice and will be investigated in further work.

Handoff in a wireless ATM network require changes in virtual connections. Several handoff schemes have been proposed, such as *connection extension* [7], *full reestablishment* [4], *partial reestablishment* [4], *multicast join/leave* [4] and *multicast group* [2]. This will require a reconfiguration of the multiagent system. The next section presents the agent reconfiguration process in case of a partial reestablishment handoff.

### 7.2. Reconfiguration during handoff

In the partial reestablishment 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 figure 10).

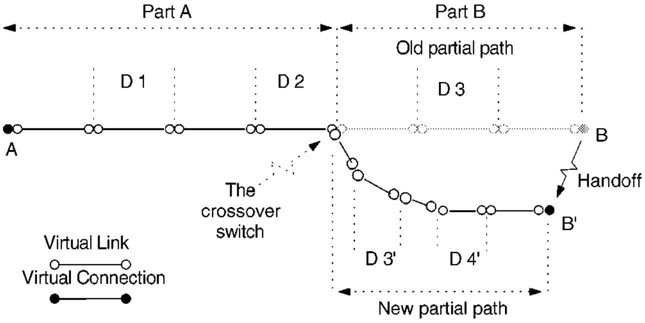

Figure 10. Handoff with partial reestablishment.

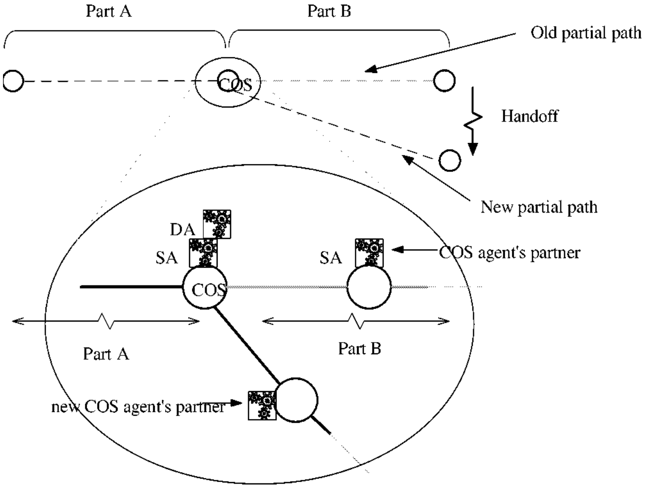

Figure 11. Configuration 2, the DA initialise the new COS agent's partner.

In this work we suppose that the super agent is not affected by the handoff. This issue will be addressed in further work.

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 figure 10), 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 figure 11), 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.

(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 figure 12), 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.

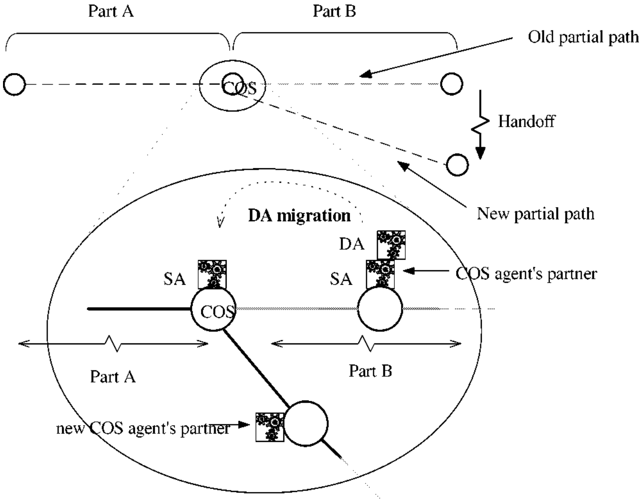

Figure 12. Configuration 3, the DA migrate towards the COS.

To migrate the 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 codes 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.

### 7.3. 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.

Let

- $D$ be the maximum end-to-end delay,
- $dA$ the maximum delay experienced by the ATM cells in part A,
- $dAmax$ the maximum delay allowed in part A (note that $dA$ can be lower than $dAmax$),
- $dBmax$ the maximum delay allowed in part B,
- $dB'max$ the maximum delay allowed in the new partial path B′,
- $dH$ the handoff delay.

**Case 1.** In the case where the transmission flow is from A to B two situations are possible (see figure 13):

(1) All the handoff delay is already absorbed by part A of the connection. This means that $dA + dH \leqslant dAmax$.

In this case, the COS will establish the new partial path with no further delay constraints (the over-

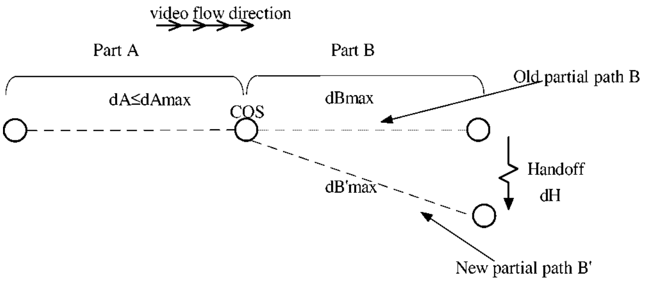

Figure 13. Video flow from A to B.

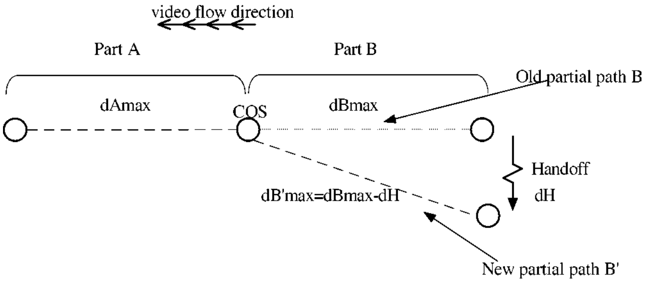

Figure 14. Video flow from B to A.

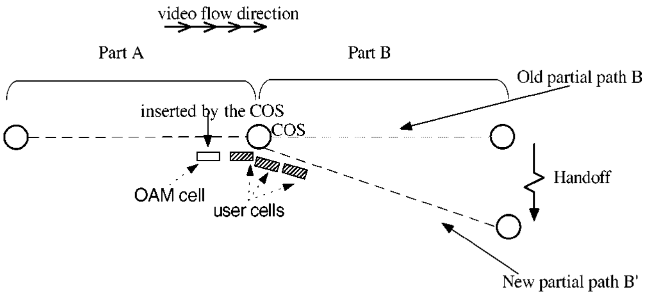

Figure 15. Delay absorption in case 1.

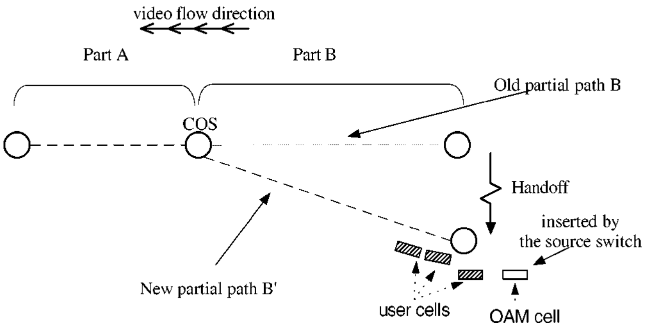

Figure 16. Delay absorption in case 2.

all connection must respect the initial delay constraint). In this case the new delay constraint is $dB'max = dBmax$.

(2) Only part of the handoff delay was absorbed by part A of the connection. This occurs when $dAmax - dA < dH$.
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. The remaining delay is $\delta H = dH - (dAmax - dA)$, and the new delay constraint is $dB'max = dBmax - \delta H$.

**Case 2.** In the case where the transmission flow is from B to A (see figure 14), then the new partial path must absorb all the handoff delay $\delta H$. The new delay constraint in this case is $dB'max = dBmax - dH$.

In the case where the transmission flow is from A to B and a part of the handoff delay is absorbed by part A of the connection (case 1 situation 2), after the establishment of the new partial path with the new constraint, the COS inserts an OAM cell at the end of its queue (with the EndOfAbsorption bit set (see figures 7 and 15)). When this special OAM cell arrives at the destination switch, the new delay constraint is no longer necessary. At the reception of this special OAM cell the destination switch will inform the super agent and this latter will distribute the handoff delay among its subordinate domain agents.

In the case where the transmission flow is from B to A (case 2), the source switch will insert a special OAM cell at the end of its queue (with the EndOfAbsorption bit set (see figures 7 and 16)). When this special OAM cell arrives at the COS, this later informs the super agent that the handoff delay has been absorbed. The super agent will distribute time credit to its subordinate DAs so that initial delay constraints will be observed by the multiagent system.

## 8. Conclusion

This paper has presented a 3-level multiagent architecture for congestion control in WATM. The agents operation relies on an adaptive cell-discarding algorithm presented in [6]. 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 cooperation. Moreover, as mobile WATM networks are characterised by the occurrence of handoffs, the proposed multiagent system integrates a dynamic reconfiguration capability. The latter allows the multiagent system to continue performing its task over the newly set up connection and thus continue guaranteeing end-user QoS requirements.

This paper has also addressed the handoff delay absorption problem. Simulation measures are envisaged to evaluate the performance of our approach.

## Acknowledgements

This article is based on our previously published material from wmATM'99 workshop, organized by the Wireless Mobile ATM Task Force of Delson Group.

## References

[1] B-ISDN operation and maintenance principles functions, ITU-T I.610, Geneva (March 1993).

[2] R. Ghai and S. Singh, A protocol for seamless communication in a picocellular network, in: *Proc. IEEE Supercomm/ICC* (May 1994) pp. 192–196.

[3] Y. Iraqi, R. Boutaba and A. Mehaoua, Configurable multiagent system for QoS control in WATM, in: *GLOBECOM '98*, Sydney, Australia (November 1998).

[4] 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* (August 1993) pp. 83–102.

[5] T. Magedanz, K. Rothermel and S. Krause, Intelligent agents: An emerging technology for next generation telecommunications, in: *IEEE INFOCOM '96*, San Francisco, CA, USA (March 1996).

[6] A. Mehaoua, R. Boutaba and G. Pujolle, An extended priority data partition scheme for MPEG video connections over ATM, in: *2nd IEEE ISCC '97*, Alexandria, Egypt (June 1997) pp. 62–67.

[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 (September 1994).

[8] K. Rauhala, ed., Baseline text for wireless ATM specification, ATM-Forum, BTD-WATM-01.03 (July 1997).

[9] A. Schuhknet and G. Dreo, Preventing rather repairing: A new approach in ATM network management, in: *INET '95 HyperMedia*, Honolulu, Hawaï (June 1995).

[10] C.K. Toh, Crossover switch discovery for wireless ATM LANs, Mobile Networks and Applications, Special Issue on Routing in Mobile Communication Networks 1(2) (1996) 141–165.

[11] T. Zhang, S. Covaci and R. Popescu-Zeletin, Intelligent agents in network and service management, in: *IEEE GLOBECOM '97*, London (1997) pp. 1855–1861.

**Youssef Iraqi** received a computer science engineering degree from ENSIAS, Morocco, in 1995, and an M.Sc. degree in computer science from the University of Montreal in 1996. He is now a Ph.D. candidate at the University of Montreal. From 1996 to 1998 he was a research assistant at the Computer Science Research Institute of Montreal (CRIM). He has been involved with research in the area of network management using intelligent agents. His current research interests include intelligent agent systems, QoS control, and resource management in wireless networks.
E-mail: iraqi@acm.org

**Raouf Boutaba** recently joined the University of Waterloo as a Professor in the Department of Computer Science. Before that he was a Professor in the Electrical and Computer Engineering Department of the University of Toronto. From 1995 until late 1997 he built and was the Director of the Telecommunications and Distributed Systems Division in the Computer Science Research Institute of Montreal. He is an adjunct Professor at the University of Montreal since 1995. Between 1990 and 1995 he was involved in EC ESPRIT and ACTS projects. Dr. Boutaba conducts research in integrated network and systems management, wired and wireless multimedia networks, and quality of service control in Internet networks. He started and chaired the IFIP/IEEE International Conference on the Management of Multimedia Networks and Services in 1997.
E-mail: rboutaba@bbcr.uwaterloo.ca

**Alberto Leon-Garcia** is a Professor in the Department of Electrical and Computer Engineering of the University of Toronto and he currently holds the Nortel Institute Chair in Network Architecture and Services. He is a fellow of the IEEE "For contributions to multiplexing and switching of integrated services traffic". He teaches undergraduate and graduate courses in communication networks, and conducts research in resources management of broadband networks and service end systems, switch and router design, Internet performance, and wireless packet access networks. He is Director of the Master of Engineering in Telecommunications program. Professor Leon-Garcia is author of the textbooks Probability and Random Processes for Electrical Engineering (Addison-Wesley, Reading, MA), and Communication Networks: Fundamental Concepts and Key Architectures, co-authored with Dr. Indra Widjaja, to be published by McGraw-Hill in 2000.
E-mail: alg@comm.utoronto.ca