Journal of Network and Systems Management, Vol. 11, No. 1, March 2003 (© 2003)

A Multi-Agent Architecture for QoS Management in Multimedia Networks

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

Increasing interest in the transmission of audiovisual applications over best-effort networks (i.e., IP and ATM ABR, and UBR services) and efficient video-aware congestion control techniques have to be designed in order to improve the video quality in the presence of cell loss. This paper presents a new slice-based discard scheme to be used with Guaranteed Frame Rate service (formerly UBR+). The scheme adaptively and selectively adjusts discard levels to the switch's buffer occupancy, video cell payload types, and Forward Error Correction Drop Tolerance. To improve its performance, we introduce a distributed multiagent system that provides a self-regulating network control management by means of automatic adjustment of congestion control parameters. The control agents are distributed in the network and operate on a local scope referred to as a "domain" consisting of two adjacent ATM switches. The aim of the overall Intelligent Quality-of-Service (QoS) control framework is twofold. First, to ensure graceful picture quality degradation by minimizing the cell loss probability for critical video data while guaranteeing a bounded cell transfer delay. Second, to optimize the network effective throughput by reducing the transmission of "nonuseful" data. Performance evaluations show significant improvement over traditional approaches.

KEY WORDS: QoS; Intelligent Agent; ATM; UBR+; MPEG-2; Management Policy.

1. INTRODUCTION

ATM networks offer five classes of services: constant bit rate (CBR), real-time variable bit rate (rt-VBR), non real time variable bit rate (nrt-VBR), available bit rate (ABR), and unspecified bit rate (UBR). Of the five classes of service, ABR and UBR are primarily designed for data traffic that has a bursty unpredictable behavior. These best-effort services are primarily based on utilizing the excess bandwidth in the network at a lower usage cost, and are expected to support a significant part of the multimedia traffic.

[1] University of Waterloo, School of Computer Science, 200 University Avenue West, Waterloo, Ont. N2L 3G1, Canada. E-mail: {*rboutaba, iraqi*}*@bbcr. uwaterloo.ca*

[2] University of Versailles, CNRS-PR*i*SM Laboratory, 45 av. des Etats-Unis, Versailles, France. E-mail: *mea@prism.uvsq.fr*

1064-7570/03/0300-0083/0 © 2003 Plenum Publishing Corporation

UBR is the simplest service class in the sense that users only negotiate their peak cell rates (PCR) when setting up the connection. They can then send bursts of video frames as desired at any time at the peak rate. If too many sources send traffic at the same time, the total traffic at a switch may exceed the output capacity, thus causing delays, buffer overflows, and losses. The network tries to minimize the delay and loss, however, with no guarantees. In this paper, we propose a quality-of-service control framework for the delivery of best-effort video applications over the UBR service. The goals of this framework are: (1) Minimize loss for critical video data with bounded end-to-end delay for the arriving cells; and (2) Reduce the "bad" throughput crossing the network. To achieve these goals, the proposed framework includes three components: an intelligent video data partitioning and prioritization mechanism located at the source; an efficient switch scheduling strategy with an adaptive discarding technique; and a multi-agent system for global network resource management and dynamic tuning of switch parameters. The control of network parameters, based on policies derived from goals, has the advantage of creating independence between policy information and management components, which in turn enables dynamic change of policies.

The paper is structured as follows. In Section 2, we present the integrated and cooperative mechanisms located at the edge (i.e., source and destination) and within the network (i.e., switches) for improving the picture quality of video applications over the UBR best-effort service. We emphasize on the new extended cell priority assignment scheme, named DexPAS, as well as the introduction of an AAL5 AudioVisual Service-Specific Convergence Sublayer (AV-SSCS). This section also gives a complete description of a new multi-layer cell discarding mechanism, named Partial video Slice Discard with support of Forward Error Correction (FEC-PSD). The latter takes into account both the hierarchical MPEG-2 data structure and the forward error correction (FEC) capability in the AAL5 AV-SSCS. Sections 3–5 describe the intelligent multi-agent system including the multi-layer architecture, the exchange of management information, inter-agent communication, and the agent control policies. Section 6 concludes the paper and point out directions for future research.

2. AUDIO-VISUAL QUALITY-OF-SERVICE CONTROL FRAMEWORK

2.1. A Dynamic Extended Priority Assignment Scheme (Dex-PAS)

To avoid worsening congestion and higher transit delays, several proactive and reactive control approaches propose to drop a lower priority cell (rather than delaying it) and re-allocate its buffer space to a higher priority cell [1]. These techniques mainly rely on ATM prioritization capabilities. Data prioritization may be applied at two different levels: cell-level and connection-level [2]. Data prioritization at the cell level discriminates between cells within a single channel. The cell loss priority (CLP) bit in ATM headers is used to provide a two-levels cell priority mechanism. A similar approach is proposed by Pancha and Elzarki [3], where an MPEG video stream is partitioned using frequency domain transform and subsequently transmitted over a single ATM virtual channel. The data partitioning scheme is implemented at MPEG block and macro-block layers. A simpler approach sets the priority levels of cells belonging to consecutive frames to different values [4]. Both approaches use the cell loss priority (CLP) bit and do not efficiently capture MPEG data structure complexity due to limitations of the CLP mechanism.

In Mehaoua and Boutaba [5], a new field in ATM cell headers is defined and referred to as Extended CLP (ExCLP). This field comprises the classical CLP bit and the adjacent Payload Type Identifier (PTI) ATM-user-to-ATM-user (AUU) bit [6]. Used individually, these two single bits define only three distinctive cell types: high priority cells, low priority cells and End-Of-Message (EOM) cells. The recollection of these two bits in the new ExCLP field permits a better utilization of the cell header with the definition of up to four cell types within a single channel.

In this paper we propose to use the Extended CLP (ExCLP) to define a new video data formatting and prioritization scheme named Dynamic Extended Priority Assignment Scheme (DexPAS). DexPAS is fast, simple to implement, supports multi-layer forward error correction (FEC) and is sufficiently generic to be performed at any MPEG data layer (i.e., frame, slice, macro-block, or block). The emphasis in this paper is on the slice and frame layers. In particular, the data partitioning is performed at the video slice layer whereas the priority assignment is performed at the video frame level. DexPAS uses the ExCLP field to dynamically assign cell priorities according to the current MPEG frame type (i.e., (I)ntra (P)redictive or (B)i-directional predictive) and the reception of backward congestion signals from the network. Table I presents the mapping of MPEG data frames into the ExCLP field. In contrast, the traditional approach restricts the number of priority to two and under-utilizes ATM capabilities.

Cells belonging to Intra-coded frame have a high priority and their ExCLP flag is set to ‘00’. B-frames have the lowest priority and the associated cells have an ExCLP value of ‘01’. For P frames, they are alternatively assigned a high and a low priority depending on the network load. At the beginning of the transmission,

Table I. New ExCLP Field Mapping

Cell type	CLP	PTI-AUU	Priority
Intra/predictive	0	0	high
Predictive/bidirectional	0	1	low
End of control block/XORing FEC	1	0	very high
End of video slice	1	1	very high

P-cells are initialized with a high priority and the resulting mode is called "*IP/B partitioning mode*".

When the buffer queue length *(Ql)* exceeds an upper threshold, early congestion is detected and the ATM switch sends a feedback signal to the source, which in turn adjust P-cells priority level to low (i.e., "*I/PB partitioning mode*"). In our implementation we use forward Resource Management (RM) cells with the Congestion Indication (CI) flag marked to notify the destination and afterwards the source. These cells are called I/PB-RM cells. When *Ql* decreases below a lower threshold, a new signal (i.e., using IP/B RM cells) is transmitted to the source, which in turn modifies its data partitioning by switching back P-cells' priority to a high priority (i.e., IP/B mode). The '10' value is used to allow the design of a two-level video-aware cell discard scheme located at every switch along the connection path. The cell with an ExCLP field set to '10' is referenced as End-Of-control Block (EOB) and delimits a group of video cells under the FEC control.

Similar to PTI ATM-user-to-ATM-user bit used to indicate whether it is the last cell of an upper-layer protocol data unit (e.g., TCP packet), we define a flag to distinguish between successive video slices. The cell with an ExCLP flag set to '11' is termed End-Of-video Slice (EOS) cell. Finally, DexPAS takes the advantages of both static I/PB and static IP/B priority partition techniques [4] and extends ATM capabilities to provide up to four priority levels whereas the traditional approach restricts the number of possible cell types to three and thus under utilizes the ATM header. This dynamic priority assignment strategy minimizes loss of critical video frames and provides better performance than static CLP-based techniques. The main drawback of the scheme is that its efficiency is a function of the round trip delay, and therefore depends on the network topology and the length of the links.

While a new upgraded and nonbackward compatible ATM cell structure has been recently proposed that reinforces the end-to-end error control ability, and greatly increases the efficiency in data transmission [7], with DexPAS each CLP and PTI bit assumes its original meanings according to the standard [6]. Consequently, any ATM switch crossed by cells set by DexPAS can operate correctly and may or may not take advantage of our prioritization scheme. However, in order to ensure optimal end-to-end quality-of-service for video applications, each node along the path should at least perform the FEC-PSD mechanism presented in Section 2.3.

2.2. An Audio-Visual Service-Specific Convergence Sublayer (AV-SSCS)

The key layer that controls the end-to-end performance is the ATM adaptation layer. AAL type 5 is currently the most commonly used. However, AAL5 is inadequate for the transmission of variable bit rate video and may therefore require additional features. For instance, the AAL5 does not provide adequate protection against erasures (cell losses). It was mainly designed for loss-sensitive data transfer applications that make use of reliable transmission protocols, which handle error correction by means of a retransmission mechanism. Therefore, it only provides an error detection mechanism based on the CRC-32 parity check computed on a PDU basis together with the integrity of the length of the received packet (when compared with the Length Indicator field). In Mehaoua *et al.* [8], a new audiovisual SSCS for AAL5, called AV-SSCS, is proposed to address this issue.

The proposed SSCS implements a robust multi-level FEC mechanism for the transmission of hierarchical MPEG encoded video. This AV-SSCS is based on both Reed-Solomon [9] and burst erasure (RSE) codes [10]. In comparison to those based only on Reed-Solomon codes with byte interleaving (e.g., in the case of AAL1 [11] and the more recent AAL2 [12]), this approach permits the construction of flexible virtual matrix at the SSCS from subsequent SSCS-PDU, and correction at the byte and cell granularities. The virtual matrices are called Control Blocks (CB) (see Fig. 1) and are associated with a Cell Drop Tolerance (CDT) parameter that depends on the matrix dimensions. The CDT parameter indicates the maximum number of cells that can be discarded by a congested switch along the connection path before considering the whole CB totally corrupted and unrecoverable by the destination.

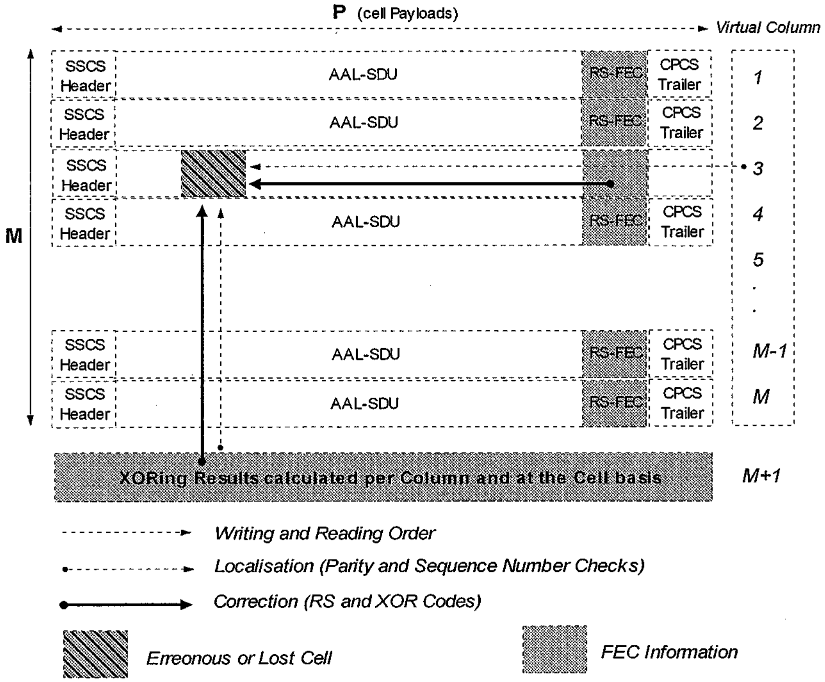
P (cell Payloads)
Virtual Column
SSCS Header
AAL-SDU
RS-FEC
CPCS Trailer
M
1
2
3
4
5
M-1
M
XORing Results calculated per Column and at the Cell basis
M+1
Writing and Reading Order
Localisation (Parity and Sequence Number Checks)
Correction (RS and XOR Codes)
Erreonous or Lost Cell
FEC Information

Fig. 1. Multi-level FEC scheme using Control Block structure.

The aim of the proposed video packet discarding scheme is to take into account the Cell Drop Tolerance per Control Block in order to intelligently start and stop cell discard. This scheme is described in the following section.

2.3. An FEC-Aware Partial Video Slice Discard Scheme (FEC-PSD)

2.3.1. The FEC-SD Algorithm

When network congestion occurs, a smart reaction is to drop incoming cells based on their importance. The Selective Cell Discard (SCD) can be significantly improved with packet-oriented communications. Indeed, the drawback of SCD is that the transmission of useless cells (in our case tail of corrupted slices), may congest upstream switches. The alternate strategy is to drop all subsequent cells from a slice as soon as one cell has been dropped. This strategy is also referred to as Partial Packet Discard (PPD) and has been widely studied for improving the effective throughput of TCP connections. In our network configuration with shared buffers, the question that arises is which incoming slices should be a candidate for partial discarding.

The simplest approach for selecting a packet to drop, called drop-tail, discards the cell that has just arrived in the queue and caused the congestion. Floyd and Jacobson showed that connections with short packets can unfairly suffer using this approach [13]. Mankin proposed to replace the drop-tail policy with random drop, where the dropped packet is chosen at random from the queue [14].

In Mehaoua and Boutaba [15], a variant of PPD called Adaptive Partial video Slice Discard (A-PSD) has been proposed to cope with this problem in video networking environments. The proposed approach consists of selecting the packet (i.e., video slice) to be dropped with respect to MPEG data hierarchy and the congestion level (i.e., switch queue length). In this paper, we propose an enhancement to this mechanism to support Forward Error Correction. The new scheme, named FEC-aware Partial video Slice Discard (FEC-PSD), is performed at both SSCS control group (CB) and video slice levels. It reduces the number of corrupted slices by assuming that a maximum number of cells per control block can be recovered by the destination SSCS using FEC techniques. This parameter (the "Cell Drop Tolerance" or CDT) depends on the size of the SSCS Control Block structure chosen by the end-terminals.

Compared to a simple A-PSD, FEC-PSD stops discarding as soon as the congestion decreases and only if the number of previously dropped cells in every control Block is below the cell drop tolerance 'CDT'. Using this approach, the proposed scheme acts at a finer data granularity (i.e., Control Block) and better preserves entire slices from discarding. Obviously this can not be achieved without the use of DexPAS that allows the detection of both slice and control block boundaries at the cell level.

The integration of the three mechanisms (i.e., DexPAS, AV-SSCS, and FEC-PSD) provides us with an efficient and intelligent video delivery service with

Quality of Picture (QoP) control optimization. The ultimate aim of our framework is to ensure graceful picture degradation during overload periods, as well as increasing network performance (i.e., effective throughput). The framework allows a more accurate video cell discrimination and progressive drop by dynamically adjusting the SA-PSD mode with respect to cell payload types, switch buffer occupancy and drop tolerance.

We define a low (respectively high) priority slice as a slice belonging to a low (respectively high) priority frame. We drop a lower priority slice first rather than delay it and reallocate its buffer space to a higher priority slice. This proactive strategy is performed gradually by including high priority cells if necessary. As evaluated by Mehaoua and Boutaba [5], this approach can significantly improve the network performance by minimizing the transmission of "nonuseful" video data before buffer overflow. The selective and adaptive Partial video Slice Discard algorithm is detailed in the following subsection.

2.3.2. FEC-PSD Parameters

Since video connections are multiplexed at the Virtual Path (VP) level and different Cell Drop Tolerance may be determined at the AAL, FEC-PSD must operate on a per-VC basis. This allows detecting accurately video slice boundaries for each connection during cell discard and, in turn, to use more efficiently the associated Cell Drop Tolerance. Consequently, FEC-PSD employs four state variables and one counter variable to control each video connection. Two state variables are associated with the video slice level and two state variables with the SSCS control block level.

(1) S_PRIORITY indicates the priority level of the current slice. The indicator is modified at the reception of the first cell of this slice with respect to its priority field (i.e., CLP), i.e., to indicate whether the switch is handling a high (**S_PRIORITY** = 0), or a low (**SPRIORITY** = 1) priority slice.

(2) **S_DISCARDING** indicates whether the switch is discarding (**S_DISCARDING** = 1) the slice (i.e., the tail) or not (**S_DISCARDING** = 0). **S_DISCARDING** is set to "0" by default. Two conditions can change it to "1".

- If "Q1" exceeds Low_Threshold (see Section 2.3.3) AND **S_PRIORITY** = "1" then **S_DISCARDING** is set to "1".
- If "Q1" exceeds High_Threshold and **S_PRIORITY** = "0" then **S_DISCARDING:** = "1".

When congestion decreases (i.e., "Q1" decreases below High_Threshold or Low_Threshold) and the last cell of a slice (EOS) arrives, **S_DISCARDING** is set back to the "*not_discarding*" state. Consequently, other cell arrivals may only change the flag from *not_discarding* to *discarding*.

(3) **CB_DROPPED** is a counter that indicates the number of cells discarded by the switch for the current control block. It is intialized to zero at the reception of a new control block.

(4) **CB_DISCARDING** indicates whether the switch is discarding (**CB_DISCARDING** = 1) this control block or not (**CB_DISCARDING** = 0). In contrast with the slice level control, the indicator changes from *discarding* to *not_discarding* in two situations: the **CB_DROPPED** counter reaches the Drop Tolerance ‘T’; otherwise a new block is received. Other events such as cell arrivals only change the flag from *not_discarding* to *discarding.*

(5) **CB_EFCI_MARKING** indicates whether the switch is tagging (**CB_EFCI_MARKING** = 1) or not tagging (**CB_EFCI_MARKING** = 0) the EFCI bit of the cells for the current control block. **CB_EFCI_MARKING** is set to “0” by default.

- If **CB_DISCARDING** = “1” (i.e., discarding) and **CB_DROPPED** = “T” (i.e., the drop tolerance) then **CB_EFCI_MARKING** is activated and the EFCI bit for every incoming cell is set to “1” until the reception of the End-of-Block cell (EOB).
- At the reception of an end-of-block cell (EOB), **CB_EFCI_MARKING** is deactivated (i.e., not_marking) and the EFCI bit of the current EOB cell is not modified.

The use of both **CB_DISCARDING** and **CB_EFCI_MARKING** indicators allow us to more efficiently manage losses occurring at subsequent switches and belonging to a control block. Indeed, when a block is partially discarded by a switch node, the subsequent switches in the path cannot take into account these cell losses to update the associated drop tolerance. The consequence is that the switches handle erroneous cell drop tolerance with adverse effect on the algorithm performance. At the control block level, the Drop Tolerance can be seen as a loss credit shared by the crossed switches.

We propose to entirely consume the loss credit as soon as a cell loss occurs. **CB_DISCARDING** is used to ensure that, for every control block, losses are concentrated in a single switch. If cells from a block tail arrive in a congested node, the use of EFCI marking allows the detection of nonrecoverable blocks since the whole drop credit has been used by a previous switch. In such a situation, we commit to the slice level control by entirely dropping the remaining slice.

2.3.3. FEC-PSD Buffer Architecture

As mentioned in the introductory section, ATM supports multiple traffic classes associated with different quality-of-service requirements. In the current shared-memory ATM switch architecture, output links share a single large memory, in which logical First-In First-Out (FIFO) queues are assigned to each link. Although memory sharing can provide a better queuing performance than separated buffers, it requires carefully designed buffer management schemes for a fair, robust and differentiated operation [16]. Therefore, advanced bandwidth management techniques such as Selective and Adaptive Partial Slice Discard require that the switches organize the output buffering on a per-VP basis, but perform tail drop on a per-VC basis [17]. Consequently in our QoS control framework, there are two levels of operation: (1) the SA-PSD runs at the VC level; and (2) the switch parameter tuning is performed by the multi-agent system on a per VP/QoS type level.

In contrast with Per-VP queuing, FIFO queuing requires incoming cells to wait for earlier cells in the same priority queue to drain before they can be transmitted. FIFO Queuing results in unpredictable delay and nonguaranteed Cell Delay Variation (CDV) across multiple switches.

We assume that cell arrivals belonging to best-effort video connections are stored in the UBR output buffers (see Fig. 2). Similarly, other incoming cells are stored in corresponding output buffers according to their type of service. The portion of unused buffer is then considered as a shared buffer. The question that arises is 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 and which allows dynamic allocation of shared buffers between the different classes. This simple policy automatically allocates the buffer space for high priority CBR

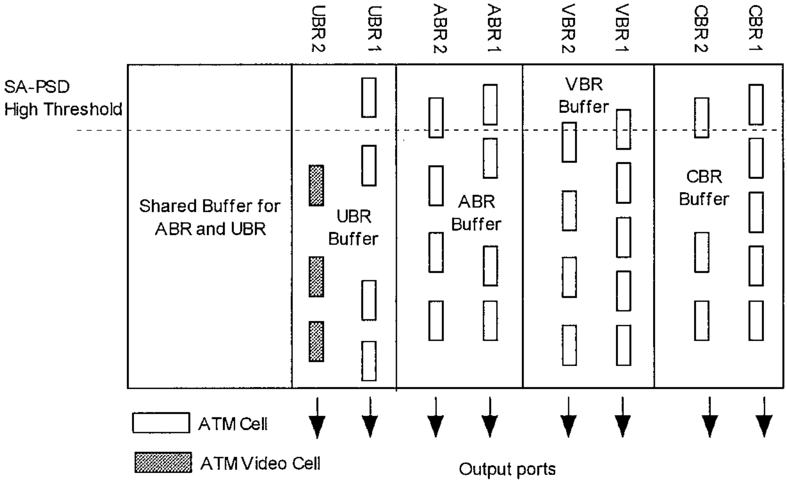

Fig. 2. Switch Buffer Architecture.

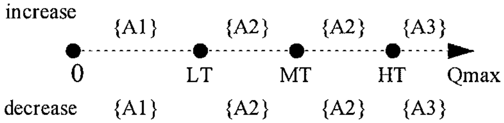

Fig. 3. Buffer thresholds.

and VBR traffic and dynamically manage the remaining buffer space for best-effort ABR and UBR traffic. Dynamically managing buffer space means that all shared buffer space is flexibly allocated to VPs on an as-needed basis. As depicted in Fig. 3, FEC-PSD uses three buffer thresholds: Low_Threshold (LT), Medium_Threshold (MT) and High_Threshold (HT). The utilization of three thresholds instead of two reduces the speed of oscillation for the transmission of DexPAS Resource Management cells and has shown better performance.

2.3.4. FEC-PSD Operation Modes

The thresholds shown in Fig. 3 define three operation modes:

(1) Mode {A1}: If the buffer queue length (Q1) is lower than Low_Threshold, for every connection having their CB_discarding indicator equal to “0”, no cells are discarded. For all other connections, the cells are accepted and may have EFCI marked if **CB_EFCI_MARKING** is activated (i.e., equal to “1” - see Section 2.3.2).

(2) Mode {A2}: If the total number of cells in the buffer exceeds Low_Threshold but is still below High_Threshold, for every video connection emitting a low priority video slice, FEC-PSD starts to discard their incoming cells with respect to the drop tolerance associated with each connection. We fairly distribute the discarding among the targeted connections using a round robin policy. If the light congestion is subsisting, the algorithm commutes to the slice level and starts to discard the incoming low priority cells until the reception of an EOS cell. The last cells are always preserved from elimination since they provide indication of the next slice. The cells with a higher priority are accepted in the buffer. This mode stops when ‘Q1’ falls down the Low_Threshold.

(3) Mode {A3}: This mode is activated when Q1 exceeds High_Threshold (HT). Incoming slices are eligible for discarding regardless of their priority level. This mode behaves like Mode {A2} for intelligently spreading the losses over connections with respect to their drop tolerance. It stops when queue length falls below HT.

The I/PB RM cells are transmitted to all video sources when Medium Threshold (MT) is exceeded, while IP/B RM cells are sent only when Q1 drops below Low Threshold (LT). At the reception of feedback signals, the sources immediately change their operation mode. Consequently, some P-frames may transmit cells with different priority. Using this adaptive strategy B-slices are firstly dropped to quickly reduce buffer occupancy during light congestion, while P and I-slices are preserved from elimination. If the congestion worsens, both B and P-slices become subject to elimination.

This proactive scheme ensures a graceful and manageable QoS degradation between a set of video connections sharing the same buffer pool. Using the selective FEC, recoverable slices are preserved from loss while highly corrupted ones are discarded to improve the network performance (i.e., effective throughput). Using the distributed discarding, picture degradation is fairly shared among the connections. Since video connections may use different grouping modes with nonequivalent drop tolerance factors, switches can virtually aggregate the drop tolerances to intelligently manage cell discarding during overload periods.

3. A MULTI-AGENT ARCHITECTURE FOR SELF-REGULATING NETWORK CONTROL PARAMETERS

The drawback of most existing video QoS control frameworks is 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 do not dynamically adjust to QoS variations. Similarly, switch parameters (e.g., buffer thresholds) are set for the virtual path lifetime and do not benefit from network load changes.

The static approach is not optimal and can be improved using an intelligent multi-agent system that ensures self-regulating network management through global network state awareness and agents' interaction. As presented by Schuhknet and Dreo [18] and Zhang *et al.* [19], the primary task of these intelligent agents is to relieve the network operator from the adjustment of resource allocation and congestion control parameters (i.e., 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 [20]. 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 and the system resources 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 way, intelligent agents have been used (See Vayias *et al.* [21]) for ATM control and resource management in multi-provider environments and software agents have been used [22] for the dynamic configuration of ATM permanent paths.

This section emphasizes the automatic adjustment of the FEC-PSD parameters (i.e., Thresholds, Cell Drop Tolerance) to ensure a low cell loss ratio with a bounded end-to-end cell transfer delay in an ATM network.

3.1. Multi-Agent 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 Fig. 4, the lower architecture layer is controlled by a set of intelligent agents, referred to as Switch Agents (SA). SAs are located at every ATM switch and monitor the switch behavior (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 minimal control information exchange within a specific Managed Domain.

SAs are responsible for the syntactical aspects of the management information (i.e., collection and representation), while the DA focuses on the semantic aspects (i.e., intelligent processing, decision-making, etc). A DA aggregates information reported by its subordinate SAs. Since it embeds a greater knowledge of the system's state, it is able to make management decisions leading to the invocation

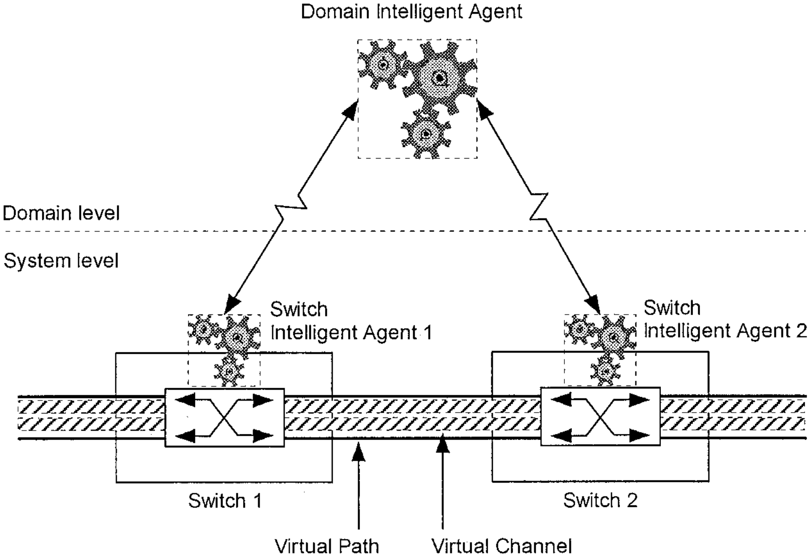

Fig. 4. A Managed Domain.

of management actions to be executed by the underlying SAs. At the system level, an SA may execute its tasks totally de-coupled from its neighbors. By using such 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 decisions, reactions have to be in the order of magnitude of cell switching. To meet this temporal requirement, inter-agent distances and the amount of control data 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 agent activities, we will focus in this paper on the management of the switch output buffer. After the connection set-up phase, an SA process is launched on every switch. Subsequently, the first SA in the direction of the information flow sends a *join* message (an OAM cell with the join bit set, see Fig. 6) to its next neighbor requesting it to be its partner. Upon 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. 6) is sent to the subsequent switch. This later acts as the first SA by sending a *join* message to its upstream neighbor.

3.2. Intelligent Agents Co-Operation and Communication Operations

To support agents' co-operation and communication, 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 [23]. 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 (i.e., VPI) as user cells, but a separate virtual channel (i.e., 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 value of the VCI field and the PTI field respectively. In this paper, control information between SAs and DA is carried by 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 an F4 OAM cell periodically (i.e., every fixed time interval T), which is looped back at the switch agent within a single domain (see Fig. 5).

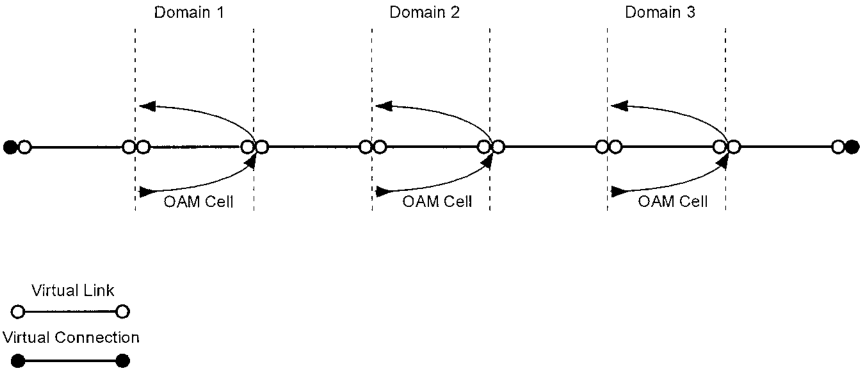

Fig. 5. IA Information Exchange Mode.

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. In contrast, shorter intervals allow faster response but introduce greater variance in the response. The determination of the T value is out of the scope of this paper. Nevertheless, a T parameter in the order of magnitude of a smoothed Round Trip Time (RTT) may be suitable and will be investigated in future work.

Figure 6 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 for use in the initial system configuration. 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.

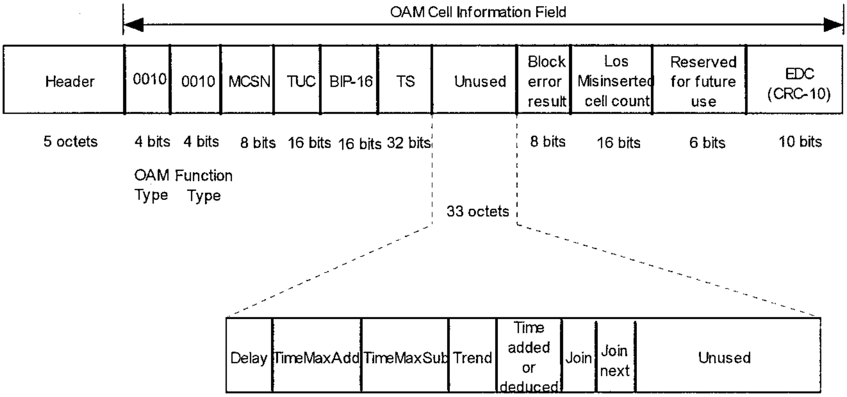

Fig. 6. OAM F4 cell structure.

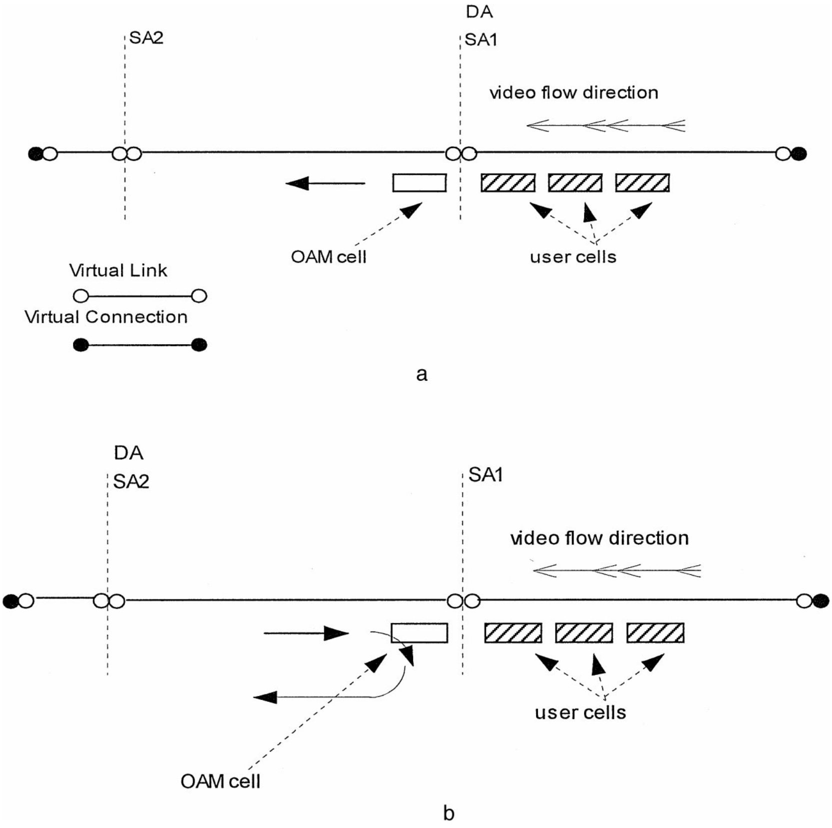

Fig. 7. (a) SA; (b) communication according to information flows.

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 direction of the information flow, two cases are possible and are shown in Fig. 7. In Fig. 7a, the SA1 updates its variables and the DA sends an OAM cell to SA2. In Fig. 7b, the DA sends an OAM cell to SA1. Upon reception of the OAM cell, SA1 applies the necessary modifications and notifies its partner with another OAM cell in order for this one to change its parameters.

4. SWITCH AGENT BEHAVIOR

4.1. Switch Agent Parameters

The SA maintains the following resource and control parameters: The available space in the shared buffer (F); 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 the switch (**TIME_MAX_SUB**). It depends on HT and the average queue size (L) and the trend in the switch load (**TREND**). The first three variables are transmitted to the DA expressed in terms of temporal units (i.e., 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 may not be sufficient to allow the DA to make accurate decisions.

4.2. Switch Agent Operations/Policies

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

- **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, computed as follows: γ^* (number of available cell slots/output rate), where γ represents the percentage of the shared buffer (F) that the switch is allowed to use (policy P.1).
- **TIME_MAX_SUB:** The DA may 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 also has to express this information in terms of temporal units, computed as follows: α^*[(**HT** - Average queue size)/ output rate], where α represents the percentage of R (see Fig. 8) that the switch can free (policy P.2).
- **TREND:** This parameter represents the trend of the R variable (Fig. 8). If R increases, the load of the switch decreases. Conversely if R decreases, the load of the switch increases. This value is computed as follows: **TREND** $= -\partial R/\partial t$.

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.

γ and α are parameters of the management policies that the network administrator can tune to achieve the highest possible performance. The selection of the parameters γ and α is crucial for the stable operation of the network, however it is

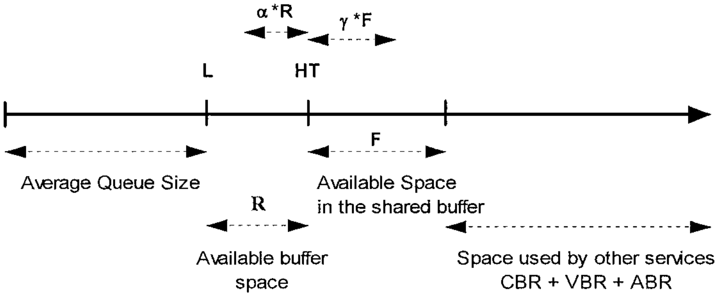

Fig. 8. Switch Buffer Parameters.

not a trivial task. The selection of optimal values for these parameters is the subject of further research. As these parameters have a local scope, it is also important to define a network-wide selection and tuning processes.

4.3. Switch Agent Pseudo Code

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

5. DOMAIN AGENT BEHAVIOR

5.1. Domain Agent Policies

Each DA maintains the following information: Current maximum transit delay at the switches; the maximum delay to add to each switch; the maximum delay to subtract from each switch; and the switch load trend. The DA distributes the delay between the two switches in order to realize 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 receives 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 always has to be 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 cells already in the buffer at S1, the value used cannot exceed **TIME_MAX_SUB** of S1. The added value cannot exceed **TIME_MAX_ADD** of S2 according to the buffer management 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) or (trend_1 = 0 and trend_2 > 0 and R_1 > R_2) or (trend_1 < 0 and trend_2 = 0
      and R_1 > R_2) 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) or (trend_1 > 0 and trend_2 = 0 and R_1 < R_2) or (trend_1 = 0 and trend_2 < 0
      and R_1 < R_2) 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[TimeMaxAdd_2, Max((1-β)*t,t-timeMaxAdd_1)]} /* policy P.3, 0 ≤ β ≤ 1 */}} }}
```

6. PERFORMANCE EVALUATION

The simulations have been implemented using OPNet modeler version 7.0.B

6.1. Simulation Setting

The simulation network topology is a simple one that consists of two traffic sources that sends ATM cells to a single destination node through two ATM switches as depicted in Fig. 9. This choice is explained by the fact that in the proposed architecture, domains manage their resources independently. So, if the cell loss ratio is improved in part of the connection, it will be on the overall connection. Source 1 sends ATM cells using UBR+ service class and source 2 sends ATM cells using VBR service class. This later represents the background traffic that a switch may experience. The goal of the simulation is to congest the link between switch 1 and switch 2 to realize cell loss and to monitor the behavior of the multi-agent system. The link between each source and switch 1 is 10Mbps. The link from switch 1 to switch 2 is 6.5Mbps. The link from switch 2 to the destination node is 8.5Mbps. The minimum service rate for UBR+ service class

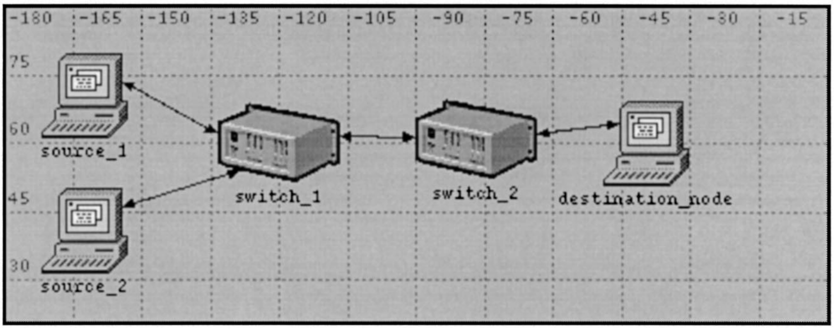

Fig. 9. Network topology.

cells is set to 1.5Mbps and 2.5Mbps at switch 1 and switch 2 respectively. The minimum service rate for VBR service class cells is set to 4Mbps and 4.5Mbps at switch 1 and switch 2 respectively. The buffer size for UBR+ cell at each switch is 200 cells, and the buffer size for VBR cells at each switch is 1200 cells. The shared buffer size for each switch is 1400 cells. The parameters α and γ are set to 0.5 and 0.4 respectively. Only the multi-agent part of the proposed QoS framework is evaluated (i.e., the queues in the switches execute a drop tail approach in case of congestion).

For each simulation run, two scenarios are considered. The first scenario is one in which the ATM switches have the multi-agent architecture installed, and the second scenario is one in which the ATM switches have no agent architecture installed. Both scenarios have exactly the same simulation setting as described earlier. Each simulation scenario is executed for about 1 minute of simulated time.

6.2. Traffic Source

The traffic source used for this simulation is based on an MPEG trace file of a music video clip downloaded from [24]. Source 1 starts transmission at frame 700 of the trace file and source 2 starts transmission at frame 0. This is to avoid any unnecessary synchronization between the two sources. Each source transmits frames at a rate of 24 frames per second, and the actual transmission of cells starts at approximately 0.02 seconds of simulation time. The 0.02 seconds is needed at the start of the simulation for ATM connection setup. The MPEG trace file has a bursty traffic pattern as shown in Fig. 10. It is the bursty nature of the transmitted traffic that will cause cell loss at the switches.

6.3. Threshold

Figure 11 shows the threshold change in the first 12 simulated seconds, where Threshold_1 and Threshold_2 refers to the change in threshold in Switch 1 and

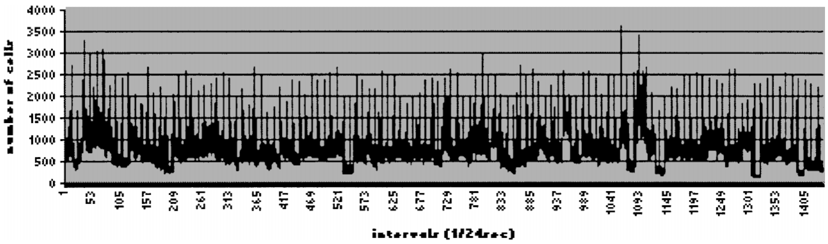

Fig. 10. Traffic pattern for the first 1440 frames of a MPEG trace.

Switch 2 respectively. Figures 11 and 12 reveal the basic policy of the architecture: when the threshold of a switch increases, the threshold of the other switch in the same domain will decrease in order to guarantee end-to-end transit delay. Note that only the first 12 seconds of simulation time is shown in the figures for readability purposes.

6.4. Cell Loss Ratio

Figure 13 shows the cell loss ratio for UBR+ cells for a simulation run. In the first simulation scenario the simulated network has the multi-agent architecture installed, and its cell loss ratio is represented by the curve "CLR_agent". In the second simulation scenario, the simulated network is a normal ATM network with no multi-agent architecture installed, and its cell loss ratio is represented by the curve "CLR_norm". As shown in Fig. 13, the simulated network with the multi-agent architecture has a lower cell loss ratio than the one without the multi-agent architecture. The improvement is about 83% with the multi-agent architecture installed. The cumulative cell loss ratio for VBR cells is null for the same two simulation scenarios.

Figure 14 depicts the result of a similar simulation run as in Fig. 13, except that the value of the parameters α and γ are different. In this simulation, α is set to 0.5 and γ is set to 1.0, and in the next simulation, α is set to 0.4 and γ is set to

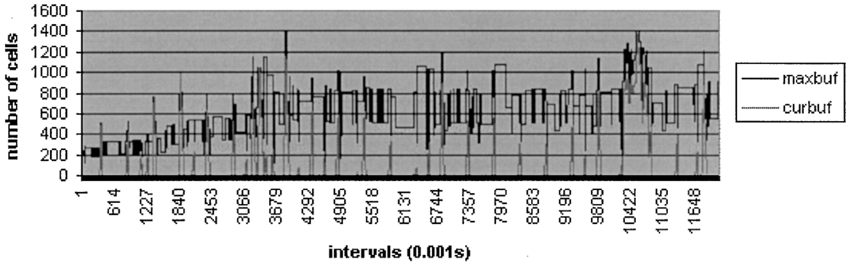

Fig. 11. Threshold change in Switch 1.

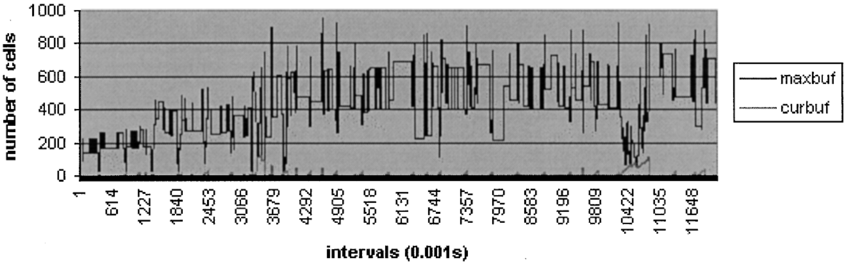

Fig. 12. Threshold change in Switch 2.

0.5. The α and γ values represent the aggressiveness in increasing and reducing the threshold of a switch. The higher the values, the more aggressive is the change. Having a more aggressive set of α and γ values, Figure 14 shows lower cell loss ratio than in Fig. 13. However if the values are set too aggressively, then other service classes that use the shared buffer may suffer. This can be seen in Fig. 15, which shows that in this second simulation run, the cell loss ratio for VBR service class cells is above zero. This is an extreme case where γ is set to 1, which means that a switch can take all the shared buffer space. γ should be set carefully to not degrade the QoS of other services.

6.5. End-to-End Delay

The multi-agent architecture reduces cell loss while guaranteeing end-to-end transit delay. The maximum end-to-end delay with the stated simulation settings above is calculated as 0.09017 seconds. Figure 16 shows the end-to-end delay for UBR+ service class cells with the multi-agent architecture installed, and clearly the curve is well below the 0.09 mark.

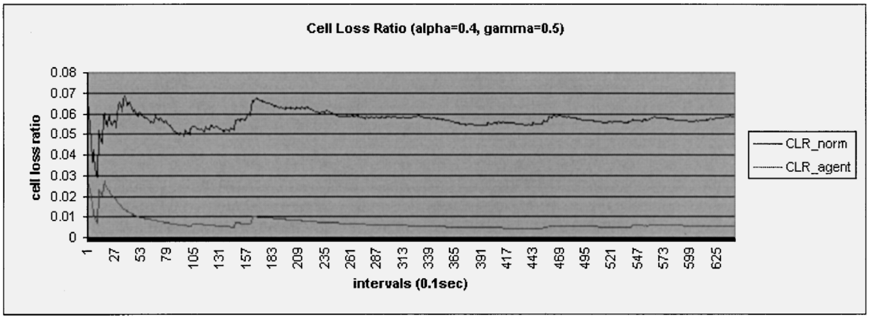

Fig. 13. Cell Loss Ratio ($\alpha = 0.4$, $\gamma = 0.5$).

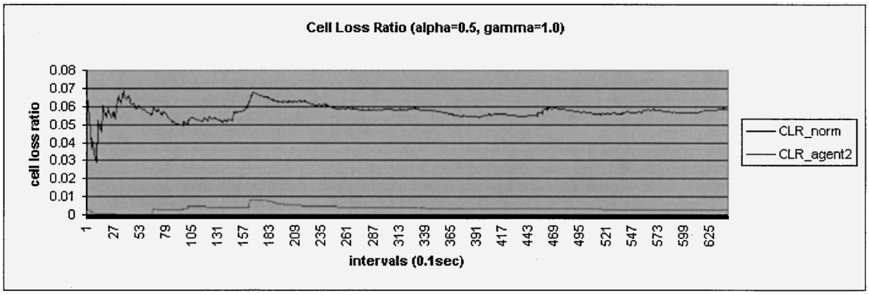

Fig. 14. Cell Loss Ratio ($\alpha = 0.5$, $\gamma = 1.0$).

7. CONCLUSIONS

Several performance evaluation results, using a variety of video sequences, show the effectiveness of improving the video quality by using a structured and cooperative set of control mechanisms to overcome the loss of ATM cells carrying VBR MPEG-2 video streams. We argue that to be able to create video systems capable of coping with cell losses encountered in computer communications, a structured set of error-resilient protocol mechanisms is needed [25,26]. Therefore, we have proposed a new quality-of-service framework for the transmission of encoded video over best-effort services. A new priority data partition technique that extends ATM prioritization capability has been proposed. Also with a better use of the ATM cell header, the Dynamic Extended Priority Assignation scheme (DexPAS) provides three different service classes per connection, in addition to the detection of video slice boundaries at the cell level. In conjunction with an intelligent video-aware cell discard scheme, called FEC-aware Partial Video Slice Discard (FEC-PSD), the system overcomes the problem of picture quality degradation caused by random drops.

To consolidate this framework we have introduced a slice-based audiovisual service specific convergence sublayer (i.e., AV-SSCS) that enhances the classical

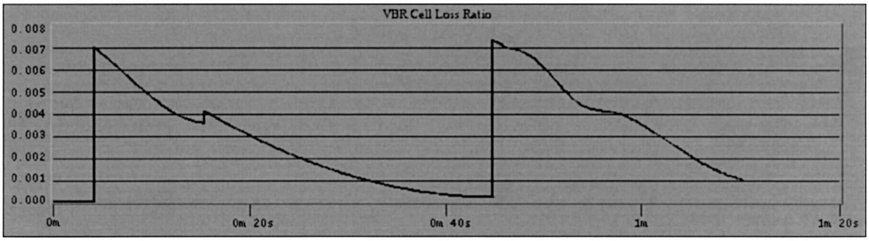

Fig. 15. VBR Cell Loss Ratio.

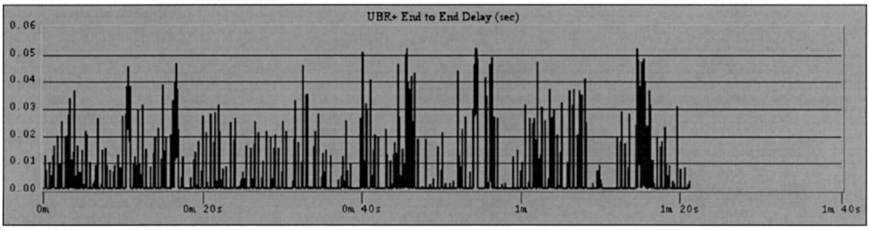

Fig. 16. UBR+ service class cell end-to-end delay.

AAL type 5. The additional features supported by the new extended AAL5 include the ability to distinguish video frame types, as well as the detection and forward correction of losses and errors at both byte and cell levels.

As such the proposed mechanisms use static control parameters like switch thresholds and the cell drop tolerance, which significantly affect loss probability at a switch node. In order to further improve the performance, we have defined an intelligent multi-agent system, which dynamically changes control parameters according to the network state, mainly maximum thresholds and consequently maximum buffer size.

It is worth noting that the proposed schemes can be implemented and used separately. In particular, the multi-agent system can operate without the intelligent video-aware cell discard scheme and viceversa. Also, it is not required that all switches in the network implement the proposed schemes. The proposed schemes and the multi-agent system can be implemented in part of the network. This allows for an incremental deployment. However, the use of the multi-agent system that reduces the number of dropped cells and the intelligent video-aware cell discard scheme which drops low priority cells first seems a winning combination in supporting bursty MPEG video streams over best effort networks such as the ATM UBR+ service.

The performance evaluation shows that the multi-agent system allows a significant improvement in terms of cell loss ratio while guaranteeing the end-to-end cell transfer delay. We expect the improvement to be higher at the level of the connection if more than one domain is involved. The agents' activity relies on design parameters, namely α, β and γ, the selection of which has not been addressed in this paper. The tuning of these parameters both locally and network-wide remains a subject for further research. It is important to note that the implementation of the proposed mechanisms and agents is relatively simple, however their deployment is more complicated due to the inflexibility of existing switches. The expected emergence of programmable switches will hopefully facilitate the introduction of intelligence into the network and thereby a more flexible management of multimedia networks and services [27].

REFERENCES

1. G. Hebuterne and A. Gravey, A space priority queuing mechanism for multiplexing ATM channels, *TTC Specialist Seminar*, 1989.
2. Miguel A. Labrador and Sujata Banerjee, Packet dropping policies for ATM and IP networks, *IEEE Communications Surveys*, Vol. 2, No. 3, 1999.
3. P. Pancha and M. El Zarki, MPEG coding for variable bit rate video transmission, *IEEE Communication Magazine*, pp. 54–66, 1994.
4. T. Han and L. Orozco-Barbosa, Performance requirements for the transport of MPEG video streams over ATM networks, *ICC'95*, pp. 221–225, 1995.
5. A. Mehaoua and R. Boutaba, Performance analysis of cell discarding techniques for best-effort video communications over ATM networks, *Computer Networks And ISDN Systems Journal*, Vol. 29, No. 17–18, pp. 2021–2037, 1998.
6. ITU-T I.361 specification of ATM layer, March 1993.
7. Xianging Chen, Shixin Cheng, and Wei Lu, The cell structure of the next generation ATM, *ICC 2001—IEEE International Conference on Communications*, pp. 79–83, 2001.
8. A. Mehaoua, R. Boutaba, J-P Claude, and G. Pujolle, Proposal of an audio visual SSCS with forward error correction, *IEEE Symposium on Computers and Communications* (*ISCC '99*), July 1999.
9. S. B. Wicker and V. K. Bhargava, *Reed-Solomom Codes and their Applications*, IEEE Press, 1994.
10. A. J. McAuley, Reliable broadband communications using a burst erasure correcting code, *ACM SIGCOMM'90*, pp. 297–306, 1990.
11. Kotikalapudi Sriram and Yung-Terng Wang, Voice over ATM using AAL2 and bit dropping: Performance and call admission control, *IEEE Journal on Selected Areas in Communications*, Vol. 17, No. 1, pp. 18–28, 1999.
12. Yuji Nakayama and Satoru Aikawa, Cell discard and TDMA synchronization using FEC in wireless ATM systems, *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 1, pp. 29–34, 1997.
13. A. Romanov and S. Floyd, Dynamics of TCP traffic over ATM networks, *ACM SIGCOMM'94*, pp. 79–88, 1994.
14. A. Mankin, Random drop congestion control, *SIGCOMM90*, Philadelphia, 1990.
15. A. Mehaoua and R. Boutaba, Performance Analysis of a slice-based discard scheme for MPEG video over UBR+ service, *IFIP ICCC'97*, November 1997.
16. Mutlu Arpaci and John A. Copeland, Buffer management for shared-memory ATM switches, *IEEE Communications Surveys*, Vol. 3, No. 1, 2000.
17. D. M. Drury, ATM traffic management and the impact of ATM switch design, *Journal of Computer Networks and ISDN Systems*, Vol. 28, pp. 471–479, 1996.
18. A. Schuhknet and G. Dreo, Preventing rather repairing: A new approach in ATM network management, *INET'95 HyperMedia*, June 1995.
19. T. Zhang, S. Covaci, and R. Popescu-Zeletin, Intelligent agents in network and service management, *IEEE GLOBECOM'96*, pp. 1855–1861, 1996.
20. Intelligent agents for telecommunications management, Special issue of *Journal of Network and Systems Management*, Vol. 8, No. 3, 2000.
21. E. Vayias, J. Soldatos, J. Bigham, L. Cuthbert, and Z. Luo, Intelligent agents for ATM network control and resource management: Experiences and results from an implementation on a network testbed, *Journal of Network and Systems Management* (special issue on Intelligent Agents for Telecommunications Management), Vol. 8, No. 3, pp. 373–396, 2000.
22. M. Cheikhrouhou, P. Conti, K. Marcus, and J. Labetoulle, A software agent architecture for network management: Case studies and experience gained, *Journal of Network and Systems Management*,

(special issue on Intelligent Agents for Telecommunications Management). Vol. 8, No. 3, pp. 349–372, 2000.
23. ITU-T I.610, B-ISDN Operation and Maintenance Principles Functions, March 2000.
24. *http://www-info3.informatik.uni-wuerzburg.de/MPEG*, University of Wuerzburg,
25. Pedro Cuenca, Luis Orozco-Barbosa, Francisco J. Quiles, and Antonio Garrido, Loss-resilient ATM protocol architecture for MPEG-2 video communications, *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 6, pp. 1075–1086, 2000.
26. A. Mehaoua, R. Boutaba, Y. Rasheed, et A-L. Garcia, An integrated framework for efficient transport of real time MPEG video over ATM UBR best effort service, *Real-time Imaging Journal*, Vol. 7, No. 3, pp. 287–300, 2001.
27. R. Boutaba and A. Leon-Garcia (eds.), Active management of multimedia network and services, Guest-editorial, (special issue on Managing Multimedia Networks), *Journal of Network and Systems Management* Vol. 8, No. 1, 2000.

Raouf Boutaba is an Associate Professor in the School of Computer Science of the University of Waterloo. He conducts research in integrated network and systems management, wired and wireless multimedia networks, and quality-of-service control in the Internet. He is the chairman of the IFIP Working Group on network and distributed systems management. He is the recipient of the Premier's Research Excellence Award in 2000.

Youssef Iraqi is a PhD candidate at the University of Montreal, Canada. From 1996 to 1998, he was a Research Associate at the Computer Science Research Institute of Montreal. His current research interests include QoS control and resource management in wireless networks.

Ahmed Mehaoua is an associate professor at the University of Versailles, France. He received his PhD degree in computer science from University of Paris, France, in 1997. His research interests include digital video communication over ATM and IP.