TECHNICAL RESEARCH REPORT

An End-to-End Measurement-Based Scheduling Architecture for ATM Networks

by Anubhav Arora, John S. Baras, George Mykoniatis

CSHCN T.R. 99-7
(ISR T.R. 99-46)

An End-to-End Measurement Based Scheduling Architecture for ATM Networks

Anubhav Arora, John S. Baras, George Mykoniatis
Center for Satellite and Hybrid Communication Networks
Institute for Systems Research
University of Maryland at College Park
College Park, MD 20742
anubhav@eng.umd.edu [*]

Abstract

ATM networks provide end-to-end QoS guarantees to connections for their lifetime in the form of bounds on delay, error and/or loss. The guarantees are important for real time connections in which losses are irrecoverable and delays cause interruptions in service. Performance management involves measurement of QoS parameters and applying control measures, if required, to improve performance or resource utilization. In this paper, we propose a new hierarchical scheduling algorithm based on dynamic priorities which are adaptive to end-to-end QoS measurements made on QoS sensitive (real-time) connections. This architecture can provide bounds on average delay and delay variation with varying background network traffic.

Keywords: ATM, Scheduling, QoS, Performance management, In-service monitoring.

1 Introduction

1.1 Introduction to ATM

ATM is a connection oriented fast packet switching protocol. It is designed to be the transport protocol for B-ISDN and to support various higher layer services for example voice,

[*] This work was supported by the Center for Satellite and Hybrid Communication Networks, under NASA cooperative agreement NCC3-528

video, web data, using statistical multiplexing to achieve efficient network resource utilization [1]. Every ATM connection is characterized by its flow type - CBR, VBR, ABR, UBR - and the QoS it requires at the cell level [2]. The traffic contract of the connection specifies its type and the various associated rate parameters like the Sustained Cell Rate (SCR), Peak Cell Rate (PCR), and the QoS of a connection consisting of the parameters Cell Transfer Delay (CTD), Cell Delay Variance (CDV), Cell Loss Ratio (CLR), and Cell Error Ratio (CER). The VBR service is further divided into real-time VBR and non-real-time VBR. The parameters CTD, CDV, and CLR are important for real time connections in which delays are interruptions in service and losses may not be recoverable using retransmissions. Real time connections are usually CBR or rt-VBR depending on the coding of the source [2], [3], [4].

In the connection setup phase a route to the destination is determined using signaling and routing protocols and the network guarantees the QoS required by the connection along the complete path. The network is obligated to provide this QoS throughout the duration of the connection (as long as the source does not violate the traffic contract). It provides guaranteed bandwidth to the CBR traffic, and guarantees an average rate to VBR traffic with accommodation of bursts of traffic. The VBR traffic contract defines the SCR, the PCR and a Maximum Burst Size (MBS). Conformance of the traffic with respect to the rate parameters of traffic contract is established at the network service access point using a traffic policer. For QoS sensitive connections, the ATM network is also obligated to provide statistical guarantees on delay and loss parameters, for example, the fraction of cells incurring a delay more than CTD would be bounded, the jitter in the end-to-end delay would be bounded by CDV etc. The nrt-VBR traffic is usually rate based and may require guarantees on errors and losses only. The network employs a variety of traffic management functions (resource management at nodes, connection admission control, cell tagging, traffic shaping etc.) and congestion control functions (cell discarding, forward congestion indication etc.) to support the various kinds of service requirements [4], [2].

Performance management i.e., measurement and control of the QoS being provided to a connection, is thus necessary for the network. The means to control the QoS being given to a connection are the traffic management functions. However the means to measure or estimate the QoS of a connection are scarce and exist only in literature. Measurement is needed for tuning of congestion control and traffic management algorithms, planning the network, and for billing in future. Measurement of QoS parameters of a channel (or path) is useful for the connection user also to know if the required QoS is being provided.

The architecture of most commercial switches today is output buffered (Fig. 1(a)). The fabric is non-blocking and can transfer all the incoming cells to their output port in one cell arrival time (referred to as a slot). Contention for the output port is the source of delays and queueing in the switches. For every output port, there is a buffer segmented on a per-type basis (CBR, VBR etc.). In most switches, the buffer is further segmented into per-VC queues (Fig. 1(b)). For every time slot on the output port the scheduling algorithm identifies the VC from the set of connections with non-empty queues whose head-of-the-line cell would be transmitted in that slot. The algorithm schedules the CBR and VBR streams to provide

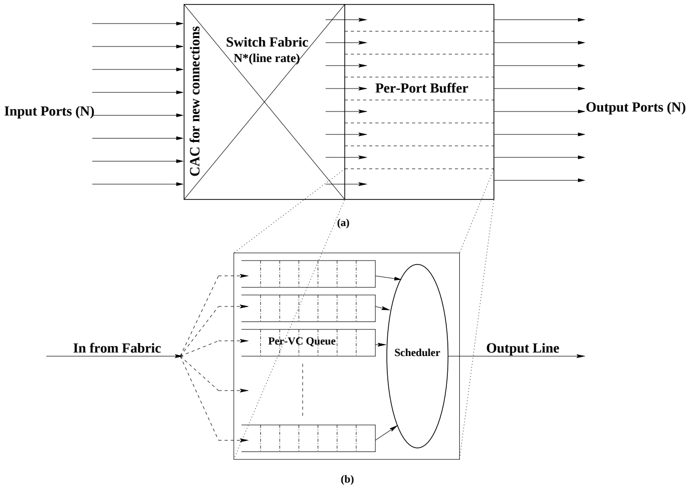

Figure 1: **(a) Output buffered ATM Switch architecture (b) Per-VC Queued buffer.**

them the bandwidth negotiated at connection setup, and schedules the best-effort ABR and UBR streams in the remaining slots. Currently, the scheduling decisions in most algorithms are made on the basis of the rate requirement of a connection and not the QoS parameters [5], [6].

1.2 QoS: Objectives and Provisioning

The QoS parameters delay, jitter and loss are important for CBR and rt-VBR classes. These are end-to-end performance parameters. With the development of real time applications and the use of the internet for voice and video connections, the need to guarantee delay, jitter and loss is increased. The delay and jitter requirements of real time applications would also be independent of the bandwidth of the connection. Thus the network should be capable of providing delay and jitter bounds on a connection independent of its bandwidth. Most of the scheduling algorithms in practice and in literature are designed to provide bandwidth guarantees to connections, optimizing on the average time spent in the system or the fairness of bandwidth distribution amongst connections. They calculate bounds on delay and loss in a node using a particular scheduling discipline and then provide the end-to-end bounds that can be supported using the number of hops in a connection. Delay and bandwidth are thus a function of each other and this implies that a connection requiring a tight delay bound has to reserve a large bandwidth. Another approach is to use traffic shapers (which buffer the incoming traffic and transmit it in a predictable pattern) at every node in conjunction with the scheduler to provide deterministic bounds, which are again a function of the shaping and scheduling discipline. To separate QoS from bandwidth the translation of end-to-end QoS requirements to local node level bounds is required, which has been ad-hoc and based on the knowledge of number of hops on the path which may not really be known at the connection establishment phase. Thus at present the resource allocation (in the form of scheduling parameters) at nodes at connection setup does not incorporate the desired end-to-end QoS bounds in any of these approaches (Fig. 2). These methods are also static and resources once allocated are not changed during the lifetime of a connection. This can either result in degraded QoS to a connection or inefficient network resource utilization.

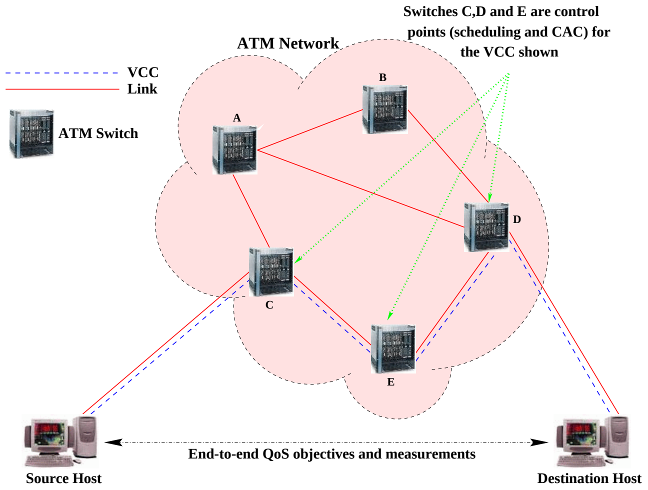
Switches C,D and E are control points (scheduling and CAC) for the VCC shown
ATM Network
VCC
Link
ATM Switch
A
B
C
D
E
End-to-end QoS objectives and measurements
Source Host
Destination Host

Figure 2: **Network view showing the measurement and control points.**

The motivation of this work thus comes from the anticipated need of monitoring of QoS parameters associated with delay and loss of QoS sensitive connections in a network, and applying control measures based on these measurements with the objective of providing the guaranteed QoS to connections, and achieving efficient resource utilization on the links and nodes. We develop a scheduling discipline that can dynamically adapt the priorities of connections in response to QoS measurements made in the network, to either increase the schedulability region of the scheduler or to improve the QoS being provided to a connection.

In the next section, we will look at papers on scheduling algorithms and distributed scheduling. In section 3, we will introduce a novel performance management architecture comprising of a dynamically adaptable priority scheduling discipline and end-to-end QoS monitoring mechanisms. Conclusions of the paper would be presented in section 4.

2 Scheduling Algorithms

There are a plethora of ATM switching and queueing architectures. In this thesis we will confine ourselves to the most popular architecture in industry, output buffered per-VC queued switch. Short reviews of scheduling disciplines are found in [5] and [6].

Priority based scheduling: A priority based scheduler assigns priorities to each queue and serves them in the order of the priority. A lower priority queue would be served only when there are no cells waiting in the higher priority queues. There is usually FIFO queueing within one priority queue. The QoS of the lower priority queues is thus completely determined by the behavior of the higher priority queues. Applied to ATM, priorities are assigned equal in one traffic class for example CBR, nrt-VBR. This kind of scheduling is QoS insensitive and does not take into account the requirements of the real-time streams.

Work-conserving fair share scheduling (Generalized Processor Sharing (GPS)): A fair share scheduler guarantees a fair share of the link to a queue according to its weight (reserved bandwidth). It introduces isolation between the queues based on their bandwidth demand. A work conserving scheduler is one which is never idle when cells are present in any of the queues. An ideal fair share scheduler employs Generalized Processor Sharing, in which the traffic is given service in ratio of their weights in infinitesimal amounts. In ATM, the basic unit of service is a cell so there exist various approximations of GPS: Fair Queueing (FQ) [7], Packetized Generalized Processor Sharing (PGPS) [8], [9], Self Clocked Fair Queueing (SFQ) [10], and Worst-Case Fair-Weighted Fair Queueing (WF^2Q) [11].

Work-conserving fair share scheduling (Round Robin (RR)): All the schemes related to GPS are based on virtual finish times. They require keeping track of a virtual time function in the scheduler and also timestamping each cell with a deadline (or finish time). The round robin schemes can however be built using frames and reserving slots for connections based on their proportion of bandwidth. Variations exist depending on the distribution of slots for a connection and what the scheduler does if the queue is empty: Weighted Round Robin (WRR), Deficit Round Robin (DRR) [12], and Uniform Round Robin (URR) [13].

All the fair share scheduling schemes mentioned above distribute bandwidth to connections in the amounts reserved to them. A proportionate distribution of the available bandwidth is done to ensure fairness. In ATM networks, the network is not obligated to provide more bandwidth to connections than specified in the traffic contract and it is upto the network to decide what to do with the available extra bandwidth. It can be used for providing better QoS to real time connections for example. Also, the delay and jitter requirements of connections are not satisfied, rather bounds are provided depending on the bandwidth provided to a connection. Thus these schemes are not consistent with the requirements ATM networks to carry traffic with different QoS requirements.

Traffic shaping and other scheduling schemes: Delay Earliest Due Date (D-EDD) [14] provides deterministically or statistically guaranteed delay bounds to services. Rotating Priority Queue+ (RPQ+) [15] emulates D-EDD using a set of priority queues and periodically updating their priority. Schemes like Jitter Earliest Due Date (J-EDD) [16] and Deterministic Delay Guarantees [17] use traffic shapers at every node to make the traffic profile predictable at every switch. In [18], the authors propose a coordinated scheduling scheme to guarantee end-to-end delay bounds. The deadline of a cell passing through multiple switches running D-EDF scheduling is iteratively computed at each switch (using the previous deadline stamped on the cell). However all these schemes address the problem of providing a certain kind of QoS, either delay bounds or jitter bounds. None of them can satisfy a range of different QoS requirements.

Thus, each scheme is suitable only to specific classes of service and optimizes limited parameters. None of them has the capability to satisfy varying QoS requirements, which is the objective of ATM as the carrier technology for the B-ISDN. Most of the schemes are local to one node and are static, i.e., the parameters do not change in the lifetime of a connection. Thus allocation of resources remains same throughout the connection regardless of their efficiency. Thus a new scheduling architecture is needed that can provide end-to-end QoS in the form of bounds on rate, delay, jitter and loss to connections.

3 Performance Management Architecture

The architecture presented in this section aims to provide end-to-end delay and jitter guarantees to QoS sensitive connections independent of their bandwidth, and bandwidth guarantees to other connections. The heart of the architecture is a priority based scheduling algorithm. The scheduling discipline is developed for the output buffered per-VC queued ATM switch architecture. For QoS sensitive connections, assigning switch level bounds on the delay and loss parameters from the required QoS bounds is a problem because the number of hops in the connection is not known at connection setup. We propose to use a hierarchical priority scheduling scheme, where the priority is dynamically adjusted using QoS measurements on a specific connection. This scheme isolates connections requiring bandwidth guarantees and connections requiring QoS performance objectives. Using measurements on QoS-sensitive connections, updates (or alarms) of these measurements are then communicated to the switches in the path using OAM cells. The switches can adapt their priority levels to maintain better or worse QoS. A switch can also degrade the priority of a connection if it is overloaded, and the QoS on the connection is much better than required (for example if the maximum delay of a real time connection is an order of magnitude less than the required QoS). If a violation in QoS guarantees is detected, an alarm in the form of an OAM cell will be sent to one or more switches to upgrade the priority of the connection. Thus there are two functional modules:

1. The scheduling algorithm - runs on every port in switches in the network to control the delay and bandwidth being provided to a connection at a node. The priorities of the connections are updated in response to measurements.
2. The monitoring and message passing protocol - used for periodic end-to-end measurements of delay and loss parameters on QoS sensitive connections. Measurements are done using OAM cells. OAM cells will also be used to send messages to specific nodes to update priorities of connections in response to measurements.

3.1 Scheduling algorithm

The first issue in developing a scheduling discipline is to know the buffering scheme in the switch and the traffic types. The premise here is that the switches are output buffered and per-VC queued. There are two classes of services that may require delay and loss guarantees, CBR and rt-VBR. The nrt-VBR connections may require a bound on cell loss. CBR always requires a bandwidth guarantee, and rt-VBR and nrt-VBR require bandwidth guarantees in the form of an average bandwidth (SCR) and high probability of switching bursts of cells (of maximum size MBS) at rates upto a maximum rate (PCR). ABR requires a minimum rate (MCR) and the rest is best-effort. UBR is also a best effort service. We will also assume that all the traffic in the network has been policed at the access points and conforms with its UPC parameters.

For nrt-VBR streams (and CBR streams requiring no QoS guarantees), the objective is to schedule them based only on their rate requirements. Thus, in a period of time, they should receive a share of the output link but how the cells will be distributed in that time is not of importance. For real time streams requiring delay guarantees, the distribution of cells in a period of time is also important, even when the average share of the connection for the output link is fair. Thus these cells need to be scheduled keeping in mind tight delay requirements. If the stream is sensitive to maximum delay only and not jitter, the cells have to be scheduled according to the time spent in the system and its performance objective, rather than the rate of the connection. If the stream is jitter sensitive, the cells have to be scheduled in a manner that the pattern of arrival of cells into the switch is approximately maintained at the output link, i.e., the variance of delay in the switch has to be within bounds. The rate of the connection does not come into picture in this class, given that the stream is policed for rate at the network access node. If the connection is sensitive to maximum delay and jitter, the average delay has also to be kept in bounds in addition to above requirements. A loss sensitive connection essentially demands a certain amount of buffering in the switch, which can be provided by either reducing the delay of the connection in the switch or giving the connection a large buffer.

Thus we define different classes of service in a switch:

1. Rate based and QoS insensitive (example nrt-VBR): requiring an average bandwidth.
2. Maximum delay sensitive (CBR or rt-VBR): requiring bounded time delay in a switch.
3. Jitter sensitive (CBR or rt-VBR): requiring approximately same pattern on the output as on the input. Also requires a bound on the maximum delay in a switch.
4. Loss sensitive: requiring a certain amount of buffer in the switch in addition to any of the above requirements.

Note that for all CBR and VBR connections, policing and traffic shaping is done at the network access points to make the traffic conform to its UPC specifications once it is inside the network.

The proposed scheduler is hierarchical, with a top level scheduler assigning slots to lower level schedulers for the different traffic classes (Fig. 3). Each of the per-VC queues i has an associated urgency parameter $U_i(n)$ (on a scale of 0-100), defined as the urgency of the head-of-the-line cell of that queue to exit the queue at that slot n. The urgency of the lower-level scheduler at a particular time is the maximum of the urgencies of the queues under it. For each slot on the output, the lower level schedulers contend and the slot is assigned to the highest urgency queue (ties are resolved randomly). Three lower level schedulers are defined:

1. Rate based: The rate based QoS insensitive traffic will be switched by a round robin scheduler like WFQ. The urgency of a cell at the head of a queue (of this scheduler) to grab a slot at the output rises as the connection gets no bandwidth on the output and vice versa. A moving average of the bandwidth provided to the connection is maintained and the urgency reflects its difference with the traffic contract.

2. Delay based: The deadline of each cell j of queue i, is the local delay bound in the switch δ_i plus the arrival time of the cell A_i^j, is calculated and stamped on the cell. The urgency of a head-of-the-line cell k, $U_i^k(n)$, to grab a particular slot on the output rises linearly with time as the deadline approaches. Thus we have

$$\begin{aligned} U_i^k &= 0 \qquad n\epsilon(0, A_i^k), \\ &= \frac{(100-\alpha_i).(n-A_i^k)}{\delta_i} + \alpha_i \\ & \qquad\quad n\epsilon[A_i^k, A_i^k+\delta_i], \\ &= 100 \qquad n \geq A_i^k + \delta_i \end{aligned} \tag{1}$$

The parameters α_i and D_i^k determine the delay of a cell inside a switch. The choice of these parameters depends on the desired level of performance. A range of delay bounds can be provisioned using these parameters.

3. Jitter based: The notion of deadline here is that the cell should be transmitted at a time closest to the deadline. The scheduler should pick the head-of-the-line cell whose deadline is the closest to current time Thus the urgency of a cell at the head of the queue rises to the maximum very sharply at the time of the deadline and stays there. Thus we now have

$$\begin{aligned} U_i^k &= 0 \qquad n\epsilon(0, A_i^k+\beta_i), \\ &= \frac{(100-\alpha_i).(n-A_i^k-\beta_i)}{\delta_i-\beta_i} + \alpha_i \end{aligned}$$

$$\begin{aligned} & & n\epsilon[A_i^k+\beta_i, A_i^k+\delta_i^k], \\ = & \quad 100 & n \geq A_i^k+\delta_i \end{aligned} \tag{2}$$

The parameter β_i provides some space to smooth local fluctuations in the delay and make sure that most of the cells reach the head-of-the-line before it. The parameter α_i assures that if there is no large contender, the cells will exit the queue at times very nearly $A_i^k + \beta_i$. The parameter (δ_i-β_i) provides a rough bound on the delay variation expected in one switch. It is observable that delay bound traffic is a special case of this type of traffic.

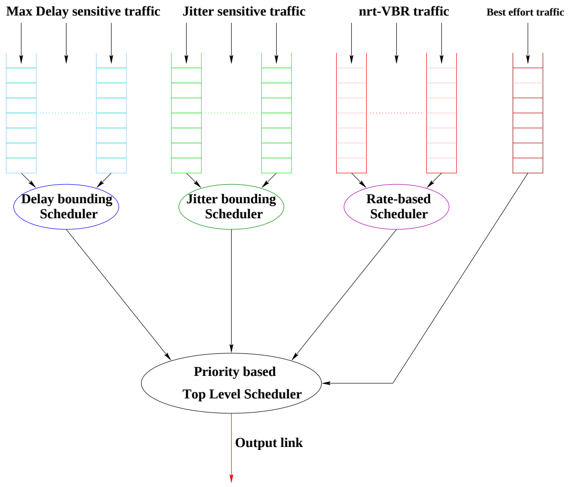

Figure 3: **Proposed hierarchical scheduler.**

If the slot is not demanded by any of the lower level schedulers, it is assigned to the best effort traffic queue (ABR and UBR). The parameters α_i, β_i, and δ_i have to be chosen so as to provide a fair distribution of resources amongst the various types of connections. Note that the concept of fairness now is very different from that of providing extra bandwidth proportionately. Fairness now extends to comparing the resources used by connections demanding different QoS requirements. These parameters are also adapted according to network measurements. For example, if a connection is incurring a high average delay, the parameter α_i can either be increased at the bottleneck node or all the nodes in the path. Thus, the assignment of switch parameters at connection startup need not be exact as they can be tuned with time. If a violation with respect to loss parameters is detected, the switches can either increase the buffer allocated to the connection or increase its priority in order to reduce the bufferring requirement inside the network.

Assigning the urgency parameters at connection startup and updating according to measurements is related to the definition of fairness of resource allocation amongst services with different requirements. These problems are a part of the work in progress in this project. A working definition of fairness amongst various connection is based on the definition of the average incurred urgency of a connection Z_i. For a cell k of connection i, define V_i^k as the value of U_i^k at the time the cell was sent on the output link. Z_i is now defined as the average of V_i^k over all cells of connection i that have been transmitted. If the parameters have been chosen correctly, Z_i would be the nearly the same for all connections. Current work focuses on relating fairness with the initial choices and updates of parameters. The updates of these parameters are based on end-to-end measurements made on QoS sensitive connections as discussed next.

The algorithm also requires that some cells be stamped with a deadline. From the implementation perspective, it is not possible to timestamp cells in real time inside a switch. Thus we need approximations to this scheme which can accomplish the same objective without timestamping of cells. This would be the focus of future work in the project.

3.2 Monitoring

In-service monitoring refers to the measurement of QoS parameters of connections by inserting special cells in the data path. The Operations, Administration and Maintenance (OAM) standard for ATM from ITU-T [19] specifies fault management and in-service performance monitoring mechanisms. The main objective of performance monitoring (as in [19]) is to detect errored blocks and loss/mis-insertion of cells in a block. There are forward monitoring cells inserted by the source to measure parameters along the connection and backward reporting cells inserted by the destination in the back-channel to report the measured values to the source. The use of the optional timestamp field enables delay measurements. However, one way delay measurement requires that the clocks at the source and destination be synchronized, thus only round trip delays are accurately measurable using the optional timestamp field and loopback of cells at the destination. The standard at no point mentions the objective of using the minimum overhead of OAM cells. The precision with which the measurements are to be made, both the precision of each measurement and the interval between measurements, is not addressed in the standard.

The aim of monitoring in the new performance management architecture is to periodically measure the end-to-end QoS parameters for connections (cell delay, delay variation and/or cell loss) at the minimum possible bandwidth overhead. Monitoring of a connection is done using an OAM cell injection (and capture) device placed just before the network access equipment by the network service provider [20]. The function of this device is to insert and extract OAM performance monitoring cells and calculate the statistics of various QoS parameters. These measurements are then used to verify the conformance of QoS with the traffic contract. If QoS violations or deteriorating trends are observed, messages are sent to all (or only bottleneck nodes) to improve the QoS of a connection by increasing its priority. Periodic updates of QoS are also sent to all nodes so that nodes may decrease the priority (in order to increase resource utilization) if the QoS is much better than the contract. These messages would be sent using a new type of performance management OAM cell. The control measures can be coordinated in two scenarios:

- In a centralized setup, the device making measurements and calculating the statistics would also perform the functions of evaluating whether the QoS is in conformance with the guarantees and detecting any patterns of deterioration. This device then tries to identify the switch responsible for the violation or deterioration in QoS using identification schemes and sends a message addressed to that switch to take corrective measures and boost the priority of the connection.

- In a distributed setup, the device will send out measurements or functions of the measurements (for example alarm states) in a special cell on the connection to all the switches. The switches in the path read the information and respond to it by changing the priority of the connection, if required, according to their resource availability.

The OAM cells are capable of measurement of cell losses but the mechanism to measure delay statistics is not sufficient. Thus, in [21] and [22] the authors propose the use of management cells. Each switch on the path stamps these cells with the delay incurred at that node (a review of monitoring mechanisms can be found in [23] and a comprehensive discussion on performance monitoring in ATM is in [24]). Therefore, accurate delay and jitter measurements are possible. This scheme does not take into account the bandwidth of management cells and the minimization of the overhead. Also, accurate jitter measurements are not possible as that would require a very high sampling rate.

A new protocol for measurement of delay statistics in a connection is proposed in this paper. It is based on the definition of a new type of OAM cells called *pattern cells.* For a particular VCC, consider time slots according to the Peak Cell Rate. Pattern cells reflect the pattern of user cell transmissions in the form of a bit sequence. Each bit represents one or more time slots, depending on the OAM bandwidth overhead that can be incurred on the connection.

In the 45 octet payload of the forward monitoring cell, it is proposed to use forty bytes representing the pattern of user cell transmissions. Let the number of time slots represented by a bit be $n = 1, 2, 3, ...$ and the desired OAM overhead be h. Then each cell (320 bits) represents $320n$ PCR time slots in which the average number of cells transmitted is $\frac{320*n*SCR}{PCR}$. Thus the bandwidth overhead of using pattern OAM cells is $\frac{PCR}{320*n*SCR}$. So if h is the desired overhead,

$$n = upperint\left\{\frac{PCR}{h*320*SCR}\right\}. \tag{3}$$

Now find the least integer i such that

$$\frac{2^i - 1}{i} \geq n. \tag{4}$$

Thus, a block of i bits represents $n*i$ time slots and the number of cell transmissions in these slots can be encoded using the i bits. The relative time of transmission of a cell from the time of transmission of the first cell can now be known to within i PCR time slots always. In the standard I.610, the minimum time of sending OAM cells is every 128 cells thus giving the best case accuracy of determining the relative position of a cell is within 128 SCR time slots on the average. Consider an example: let $h = 0.5\%$, $\frac{PCR}{SCR} = 10$. Thus, $n = 7$ and $i = 6$. So the accuracy of our protocol is to relatively place a cell within 6 PCR time slots equivalent to 0.6 SCR time slot and that of OAM protocol (inserting OAM cells after every 128 cells) would be 128 time slots. In this case, our protocol has a bandwidth overhead of 0.5% and OAM protocol has an overhead of 0.77%. The price to pay here is in real time encoding of timing information in pattern cells. Also note that the pattern cells indicate the exact number of cells transmitted and thus CLR can be approximately calculated. Since the relative delay of each cell can be known, the absolute delay of each cell can also be calculated if the absolute delay of one cell is known (for example by using management cells). The accuracy of measurements i is directly related to the overhead incurred h and it is possible to trade one off for other. It is also possible to change one parameter, for example decrease the overhead when the connection is experiencing good QoS. This can be based on the idea of adaptive sampling proposed in [25].

Thus using these monitoring mechanisms, it is possible to measure the QoS parameters for QoS sensitive connections. Using the signaling cells (a type of OAM cell), these measurements are communicated to all the nodes in the path periodically. The control framework can be distributed or centralized and the pros and cons of both are also subjects for future work. These measurements are then used for updates of the priorities of connections in the schedulers as discussed in the last section.

4 Conclusions

ATM networks were designed as the carrier technology for the Broadband-ISDN. An important requirement for an integrated services network is the ability to carry traffic with varying demands of QoS. The QoS can range from only a bandwidth requirement, to requirements on end-to-end maximum cell delay, delay jitter and/or cell loss ratio. Delay and loss parameters are important for real time connections. The mechanism of controling the delay and bandwidth provided to a connection in a switch is the queueing and scheduling disciplines of a switch. In this paper we assume per-VC queues in a switch, which is the most popular commercial switch architecture. All scheduling schemes in commercial switches and in literature are based on optimization of one parameter like the bandwidth of a connection or the delay of a connection. These schemes provide static resource allocation and control the QoS at one node whereas the requirement is end-to-end.

In this paper we propose a new performance management architecture in which connections can be guaranteed end-to-end QoS. The scheduling at one switch is capable of providing bandwidth to connections, and maximum delay or jitter guarantees to real time connections independent of their bandwidth requirement. It is based upon the definition of urgency of a cell which increases with time in different ways for different types of connections. For QoS sensitive connections the parameters controlling the curve can be changed based on end-to-end QoS measurements. Using in-service monitoring mechanisms, the delay and jitter statistics are measured. These measurements are done using performance monitoring cells injected with the user cell stream. Messages are sent to all nodes (or only bottleneck nodes) if QoS violations or deteriorating patterns are detected. Periodic updates of QoS are also sent in order to facilitate switches to change parameters for better resource utilization.

The new architecture is thus capable of providing end-to-end QoS guarantees using an adaptable scheduling discipline and in-service monitoring methods. Efficient resource allocation in the network is also achieved using these methods. Since the architecture is built in a modular fashion, adding a new QoS requirement to the existing infrastructure would not involve a complete redesign. Future work in this directions includes the study of the definition of fairness between flows with varying QoS requirements and of the effect of control measures of a connection on the network performance and on the performance of other connections. The choice of the scheduling parameters of connections at startup and their updates based on measurements is also under study.

References

[1] David E. McDysan, *ATM: Theory and Practice*, McGraw-Hill, 1995.

[2] ATM Forum, *Traffic Management Specification Version 4.0*, Feb 1996.

[3] ITU-T Recommendation I.356, *B-ISDN ATM layer cell transfer performance*, Oct 1996.

[4] ITU-T Recommendation I.371, *Traffic Control and Congestion Control in B-ISDN*, Nov 1995.

[5] Natalie Giroux and Sudhakar Ganti, *Quality of Service in ATM Networks: State-of-the-art traffic managment*, Prentice Hall PTR, 1998.

[6] S Keshav, *An Engineering Approach to Computer Networking*, Addison Wesley Longman, Inc., 1997.

[7] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of fair queueing algorithm," *ACM SIGCOMM Computer Communication Review*, vol. 19, no. 4, pp. 1–12, September 1989.

[8] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: The single node case," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, pp. 344–357, June 1993.

[9] A. K. Parekh and R. G. Gallager, “A generalized processor sharing approach to flow control in integrated services networks: The multiple node case,” *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, pp. 137–150, April 1994.

[10] S. J. Golestani, “A self-clocked fair queueing scheme for broadband applications,” in *Proceedings of IEEE Infocom ’94 (Toronto)*, June 1994, pp. 636–646.

[11] J. C. R. Bennet and H. Zhang, “Hierarchical packet fair queueing algorithms,” *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, pp. 675–689, October 1997.

[12] M. Shreedhar and G. Varghese, “Efficient fair queueing using deficit round-robin,” *IEEE/ACM Transactions on Networking*, vol. 4, no. 3, pp. 375–385, June 1996.

[13] Norio Matsufuru and Reiji Aibara, “Efficient fair queueing for ATM networks using uniform round robin,” *Proceedings of Infocom ’99 (New York)*, March 1999.

[14] D. Ferrari and D. C. Verma, “A scheme for real-time channel establishment in wide-area networks,” *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 3, pp. 368–379, 1990.

[15] D. E. Wrege and J. Liebeherr, “A near-optimal packet scheduler for QoS networks,” in *Proceedings of IEEE ’97 (Kobe, Japan)*, 1997, pp. 576–583.

[16] D. C. Verma, H. Zhang, and D. Ferrari, “Delay jitter control for real-time communication in a packet switching network,” in *Proceedings of Tricomm 91 (Chapel Hill, North Carolina)*, April 1991, pp. 35–46.

[17] Seok-Kyu Kweon and Kang G. Shin, “Providing deterministic delay guarantees in ATM networks,” *IEEE/ACM Transactions on Networking*, vol. 6, no. 6, pp. 838–850, December 1998.

[18] Matthew Andrews and Lisa Zhang, “Minimizing end-to-end delay in high-speed networks with a simple coordinated schedule,” in *Proceedings of Infocom ’99 (New York)*, March 1999.

[19] ITU-T Recommendation I.610, *B-ISDN Operations and Maintenance Principles and Functions*, May 1995, ATM Forum/95-0013R10.

[20] RAD Data Communications - White Paper, *Customer Premises ATM NTUs: Cost Effective End-to-End Management of ATM Services*, http://www.rad.com/networks/whitepap.htm.

[21] T. M. Chen et al., “Monitoring and control of ATM networks using special cells,” *IEEE Network*, pp. 28–38, Sept/Oct 1996.

[22] Thomas M. Chen and Stephen S. Liu, “Monitoring and control of strategic and tactical networks using control packets,” in *Proceedings ATIRP Consortium*, Feb 1999, pp. 183–187.

[23] Anubhav Arora and John S. Baras, "Performance monitoring in ATM networks," Tech. Rep., CSHCN and ISR, University of Maryland, April 1998.

[24] Thomas Lindh, "Performance management in switched ATM networks," in *Proceedings, Intelligence in Services and Networks: Technology for Ubiquitous Telecom Services*, March 1998, pp. 439–450.

[25] Anubhav Arora, John S. Baras, and George Mykoniatis, "Delay monitoring in ATM networks," in *Proceedings ATIRP Consortium*, Feb 1999, pp. 259–263.