TECHNICAL RESEARCH REPORT

Performance Monitoring in ATM Networks

by A. Arora, J. Baras

CSHCN T.R. 98-12
(ISR T.R. 98-22)

Performance Monitoring in ATM Networks

Anubhav Arora and John S. Baras

Center for Satellite and Hybrid Communication Networks

and Institute of Systems Research

University of Maryland at College Park

April 20, 1998

Performance Monitoring in ATM Networks[1]

Anubhav Arora and John S. Baras

Center for Satellite and Hybrid Communication Networks

and Institute of Systems Research

University of Maryland at College Park

Contents

[1]Research supported by NASA under cooperative agreement NCC3-528 and by the Army Research Laboratory under cooperative agreement DAAL01-96-2-0002.

Abstract

ATM networks provide end-to-end QoS guarantees to connections for their lifetime in the form of bounds on delays, errors and losses. Performance management involves measurement of these parameters accurately and taking control measures, if required, to improve performance. This is very important for real time connections in which losses are irrecoverable and delays cause interruptions in service. In this paper, an intelligent management architecture and an information flow model for performance management proposed in the literature is reviewed. A brief description of the related ITU-T standard for performance monitoring is given and several schemes for measurement or estimation of QoS parameters are documented. Future work required in the area is also presented in the last section.

1 Introduction

Asynchronous Transfer Mode (ATM) is a connection-oriented fast packet switching protocol. Information is carried in fixed size cells along virtual channels, contained in virtual paths. ATM is designed to support various kinds of connections, voice/video/data etc. using statistical multiplexing of sources to achieve efficient resource utilization.

Every incoming call is characterized by its type - Constant/Variable/Available/Unspecified Bit Rate (CBR/VBR/ABR/UBR) - and the Quality of Service (QoS) it requires. The QoS of a connection consists of parameters like the Mean Cell Rate (MCR), Peak Cell Rate (PCR), Cell Transfer Delay (CTD), Cell Delay Variance (CDV), Cell Loss Ratio (CLR), Cell Error Ratio (CER) etc. defined for CBR and VBR services. The parameters CTD, CDV, and CLR are most important for real time connections in which delays are interruptions in service and losses maynot be recoverable using retransmissions. Real time connections are usually CBR or VBR depending on the coding of the source.

In the connection setup phase a route is determined, with the network guaranteeing (the guarantee is of a statistical nature) the QoS required by the connection along the complete path. The network is obligated to provide this QoS throughout the duration of the connection but is allowed to drop cells if the source violates the QoS contract. To this end, performance management i.e., measurement and control of the QoS being provided to a connection, is necessary for the network. It is also necessary for congestion control, traffic management, planning in the network and possibly for billing. Measurement of QoS parameters for a channel (or path) is useful for the connection also to know if the agreed QoS is being provided.

Performance management thus encompasses gathering of network performance statistics (in the form of QoS parameters), analyzing its trends and taking proactive or curitive control measures. A few papers on performance management in ATM networks are reviewed here. In the next section, the related ATM standard from ITU-T is reviewed regarding its performance monitoring aspects. In section 3, an intelligent management architecture and an information flow model for performance monitoring are reviewed. Section 4 looks at schemes for measuring or estimating the important QoS parameters. In the last section, future work required in the area and conclusions are presented.

2 The Operations And Maintenance (OAM) Standard for ATM

The OAM standard for ATM is recommendation I.610 from ITU-T [1]. There are two vertical OAM levels defined for the ATM layer, the virtual path level (F4) and the virtual channel level (F5). OAM flows are provided by cells dedicated to VPCs (or VCCs) and follow the same path as user cells. These flows can be further classified into two horizontal levels, end-to-end flows (from a VP (or VC) endpoint to another VP (or VC) endpoint) and segment flows (defined as single or multiple inter-connected VP (or VC) links). Intermediate nodes can monitor existing flows and insert new OAM cells but cannot terminate flows which are not their own. The OAM functions defined for the ATM layer are fault management, continuity check, loopback and performance monitoring.

Performance monitoring is done by inserting end-to-end or segment monitoring cells at F4 or F5 level. These cells monitor a block of user cells (of size 128, 256, 512 or 1024). The main objective of this monitoring is to detect errored blocks and loss/mis-insertion of cells. There are forward monitoring cells to measure and backward reporting cells to report the measured values to the source. The various fields defined for these cells include

- a sequence number to identify the cells
- the total number of user cells sent (a modulo 64k counter) and total number of user cells with high Cell Loss Priority (CLP = 1)
- a block error detection code (even parity Bit Interleaved Parity - 16)
- an optional time stamp for delay measurements
- for backward monitoring cells - total number of user cells received, total number of CLP=1 user cells and the block error result on the received cells

Using these fields, block errors and difference in the number of transmitted and received cells can be known. However, this does not give precise information about lost and mis-inserted cells. Only round trip delays are accurately measurable using the optional timestamp field (which is not implemented by most vendors currently) and that rarely provides information about one-way delay or bottle-neck links on the path.

It is observable that this provides limited information regarding the delays and losses which are critical to real time services. The standard does not specify how these values are used for proactive or reactive control actions required if the QoS guarantees are not being met. It also does not specify how frequently these measurements should be made; this will depend on the bandwidth of the path, the network state, and user requirements.

3 Management Architecture

3.1 Intelligent Distributed Performance Management

In [2], a generic intelligent mechanism for network management based on a distributed multi-agent system is defined. The agents are semi-autonomous, making instantaneous local decisions when they have enough knowledge and communicating with other agents when the network is not loaded. The agent is composed of a knowledge module, a control and decision making module, and a communication module. It is attached to a node and may work in isolation when necessary. The agents communicate using a general blackboard that contains at the beginning, the facts of a given problem. The agents read in the current state of the problem and write into it to co-operatively converge to a solution. The blackboard also contains a control system to supervise the actions of the agents.

A network performance monitoring architecture is proposed using the above model. The network is divided into domains and different levels of problem solving are defined - an agent working alone, agents in a domain working together and agents in multiple domains working together. OAM cells [1] are the carriers of performance information, in the form of various QoS parameters. A congestion control scheme is then proposed. It uses four thresholds to discard CLP=1 (indicating congestion) cells on specific or all ports and to send out Explicit Forward/Backward Congestion Notification (EFCI/EBCI). Upon receiving EBCN, a previous node drops CLP=1 cells on the congested path as it would be dropped on the congested node anyway.

Simulation results given for a very simple network with uniform routing policy indicate that tuning of thresholds using the multi-agent paradigm improves the cell loss by an order of magnitude as compared to fixed thresholds.

In the paper, the complexity of a real time agent is not investigated. The structure of the agent, its use of the QoS parameters and its threshold tuning policy (using predictions from a learning algorithm) are not clearly specified. The simulation assumptions seem too simple to be able to effectively model a real network. Also, the information flow in the simulation is not through OAM cells in the simulation. OAM information may be stale and inaccurate and is also quite limited in scope, which can adversely affect the knowledge of an agent.

3.2 Information Flow Model

The paper [3] explores the use of special cells as an information carrier for a wide range of monitoring and control functions. The management cells defined in the paper can conceivably carry measurement data, network state information, control information and administrative information. This can serve as an effective model for the communication of parameters required in the previous intelligent architecture.

Use of performance management OAM cells as defined in [1], requires no processing capabilities at intermediate nodes. They can be used to measure the round trip delay, the difference in the number of cells transmitted and received block errors. However, the exact number of cells lost or misinserted, the nodes experiencing congestion, and the one way delay cannot be accurately measured using the performance management OAM cell. The authors propose the use of management cells for improved monitoring as described in the next section. However, these methods require specific processing capabilities at end nodes and intermediate switches, like timestamping a cell on ingress and egress etc. which are not present in state-of-the-art switches today. Very few commercial switches implement the OAM standard completely. Thus there is a need for monitoring methods which are algorithmically simple, require minimal processing at switches and gather relevant data.

The paper also proposes the use of management cells for traffic and congestion management using explicit feedback rate control (for ABR and UBR traffic), advocating the use of ATM Block Transfers with bandwidth renogotiation for each block using RM cells (all cells in a block are carried or dropped at each node, ensuring high good-put), and dynamic VP bandwidth management (to accomodate new connections in a VP). Routing information can be communicated using management cells.

4 Measuring key QoS parameters

This section presents several measurement (or estimation) schemes for the QoS parameters, cell delay, cell loss ratio and cell error parameters.

4.1 Delay measurement using management cells

To accurately measure the one-way delay, OAM performance management cells require the source and destination clocks to be precisely synchronized to the same time, which is hardly the case. However, this method gives an accurate estimate of the round trip delay as it is the difference of two times measured from the same clock. For the method proposed in [3], the switches should be capable of collecting information about their own internal performance. Also, ATM switches should be able to modify the management cells in flight.

The one-way cell delay can be accurately measured by breaking it into delays experienced at each switch. A switch can time-stamp a management cell when it is received at the input. At the departure of the cell, the cell delay can be computed and added to the delay value already written in the cell (initialised to zero by the source). The processing delay required for this process and the propagation delays are fixed and can be precomputed. Thus, a management cell accumulates the delay along the path. The delay field at the destination gives a sample of the cell transfer delay which does not suffer from the clock synchronization problem. An alternative is to have multiple time-stamp fields in the management cell so that the delay at each switch can be recorded separately. This may be useful for diagnostic purposes, for example to determine the bottleneck link.

Management cells are required to occupy a small portion of the bandwidth, specially when the network is congested. This criterion dictates the inter-sample time. The authors have briefly alluded to this issue and stated some heuristic arguments. This scheme requires new processing capabilities at the switches to modify cells on ingress and egress. No mention is made about the accuracy of such a scheme either (which can possibly justify the additional complexity in the switches). The next section describes a procedure based on statistical analysis of the remote clock which can be used with OAM performance management cells.

4.2 Delay measurement using clock parameter estimation

This technique proposed by Roppel in [4] relies on estimation of the one-way cell transfer delay by analyzing the properties of the remote clock. Essentially, the remote clock can be modeled, its parameters estimated, and the time-stamp of the destination can be corrected for the offset. In this method, the switches do not need any new processing capabilities.

The remote clock can be modeled using a time offset parameter (ΔT_o) and a clock frequency drift parameter (α).

$$C(t) = t + \Delta T_o + \alpha(t - t_o) + \epsilon(t)$$

These two parameters can be estimated using a regression model on the delay samples from the backward reporting performance management cells. This method however requires that the minimum delay along both directions to be the same so is not suitable for asymmetric and hybrid networks. The number of samples required to converge to correct clock parameters may be large, thus the time for convergence for low bit rate links can be very high. These disadvantages can prohibit the use of this scheme for a large class of networks and connections.

The above scheme can however be modified so that each node in the network estimates the parameters of its neighbors' clocks. This can be done with better accuracy as this involves only one queueing delay in the path, and may be done more simply and accurately using a CBR connection during light loading. Once a node knows the equations of all neighbor clocks, remote clock offset from source can be known by propagating it hop-by-hop along the connection path at the connection setup time. Thus, the source can know the model for the remote clock at connection setup time and use it to correct timestamp values in OAM cells from the destination. This allows for measurement of one-way delay, although bottle-neck links can not be known. This scheme, which uses the performance management OAM cell, neither involves a large time delay to achieve accuracy nor new processing capabilities at switches and can be more accurate than Roppel's scheme.

4.3 Cell loss estimation using a fuzzy system

Estimation of CLR using a fuzzy system and a related CAC algorithm are described in [5]. The fuzzy system used has a singleton fuzzifier, a product inference engine, and a center average defuzzifier with gaussian membership functions. Such a fuzzy system has universal approximation capabilities, i.e., any real valued continuous function can be approximated using such a system.

The CLR ($P(m)$) is taken as a function of S_m (it can be the buffer size or service capacity or the number of users) which evolves as $S_m = m\delta + S_0$. The CLR when S_m is small is measured and supplied as the initial input to the system. The input to the system are previous M input-output pairs, $\{[P(k-(i+2)), P(k-(i+1))]; P(k-i) - P(k-(i+1))\}, i = 0, ..., M-2$. The output of the system is the difference between the next and current CLR values.

When the number of steps required to predict CLR at target S_m value is very large, the accuracy of the system suffers. The authors propose using asymptotic information about CLR $\hat{P}(m)$ (assumed to be known) when the $S_m \longrightarrow S_\infty$:

$$P_{improved}(m) = [1 - \lambda(m)]P(m) + \lambda(m)\hat{P}(m)$$

where the function $\lambda(m)$ gives more weight to the asymptotic behavior as $m \longrightarrow \infty$.

The authors study the behavior of this scheme with S_m being the buffer size, the service capacity and the number of users. The results give very good approximations to the actual CLR as obtained from exact source models, with the computation time at least an order of magnitude less than any other scheme. All the three parameters can be used one after the other to obtain CLR at the target values. The accuracy of the algorithm however depends on the variance of the gaussian membership function and the asymptotic behavior assumed. The paper presents some heuristic values for the two parameters. If the algorithm can be made less sensitive to the choice of these parameters while keeping the same low computational complexity, it can be effectively used in a real system for CLR measurement.

A CAC algorithm that accurately estimates the aggregate required bandwidth by using the measured CLR and the bandwidth required by the incoming call is also described in the paper.

4.4 Cell loss estimation using in-service monitoring

In [6], a technique to measure CLR using two different traffic models is proposed. With a Markovian arrival process, the log of CLR depends linearly on the buffer size (B)

$$log(CLR) \approx -\alpha - \delta B$$

and with a long range dependent traffic model, log[-log(CLR)] depends linearly on the logarithm of the buffer size

$$log[-log(CLR)] \approx log(\delta) + \beta log(B).$$

The technique uses small pseudo-buffers to estimate the CLR from a small number of incoming cells, thus decreasing the monitoring time.

The algorithm applies linear regression to incoming samples of CLR over all pseudo buffers on both models. It keeps two running counters indicating how many times a model was better than the other based on a goodness-of-fit R^2 test. After a fixed number of samples, it chooses one of the models as the correct one and then extrapolates the CLR to the physical buffer size. The mean and the variance of CLR measurement over all pseudo buffers can be calculated now and the distribution is assumed to be normal. The paper then proposes a QoS violation algorithm based on the Neyman-Pearson detection rule.

Simulations are based on Bellcore video source traces & Bellcore LAN trace data, and the G/G/1/K & aggregated homogenous on/off sources with single server queue queueing models. Three different loads, under, critical and overload are simulated. The results are very close to the actual simulation values. Another notable point is that the algorithm chose long-range dependent model for video trace and markovian model for ethernet trace. The tradeoff between the observation period and the detection probability is discussed and the decrease in observation interval through the use of pseudo-buffers is highlighted (it is about 4-7 orders of magnitude).

4.5 Parameters for error performance

In [7], the authors advocate the use of error performance parameters like Errored Second (a second including one or more bit errors), Severely Errored Second (a second in which the BER exceeds 10^{-3}) and Unavailable Time (time that begins with 10 consecutive SESs and ends with 10 consecutive seconds not consisting of SESs) rather than just the Bit Error Rate which is felt as inadequate. The paper recommends estimation of channel error parameters using ISM (In-Service Monitoring) information about anomaly and defect events. Two channel performance estimation methods are suggested: 1) using statistical estimation of expectations of ES and SES using LOF (Loss of Frame Alignment) duration and violations of error checking codes measurements and 2) using linear or multivariate regression models denoting the relationship between ISM information gathered at path and the actual error performance of the actual channel. Field experiment results indicate that ISM information should be collected at short intervals and the monitoring path rate should be close to the channel for better estimation of the channel.

The authors have mentioned several uses of ISM but have provided no specific example or application where the BER is inadequate and the parameters mentioned above are required. In other words, the authors have given no justification or need for the parameters they propose to estimate. The mechanism proposed in the standard [1] to detect errors in small sized blocks using an even Bit Interleaved Parity (BIP-16) seems adequate to measure the error performance of the channel unless a clear application for the new parameters is specified.

5 Conclusions and Future work

Performance management is an important function of ATM networks because the network guarantees to provide a certain end-to-end quality of service to the connections and is obliged to do so as long as the source does not violate its agreed upon peak and mean cell rates. Performance monitoring is useful for traffic management, congestion control, network design, planning and possibly billing.

In this paper, some literature pertaining to performance monitoring in ATM was reviewed. Operations and Maintenance (OAM) standards for ATM provide basic performance monitoring mechanism to measure errors, losses and possibly delays using OAM cells inserted after a specified block of cells. These measurements are however, quite limited in the information they gather. The standards do not clearly specify the inter-sample time and the possible uses of such measurements. Also, performance management cells are not implemented in most switches at present. Thus, a need is felt for a well defined monitoring architecture including algorithms to measure (or estimate) QoS parameters and the use of the information gathered to improve network performance. Another point to note is that the control algorithms for CAC, routing, UPC, soft-fault management etc. should dictate the performance measurement required.

A generic intelligent performance management architecture based on a distributed semi-autonomous multi-agent system interacting via a blackboard is reviewed. This architecture is used to implement a congestion management scheme and is shown to improve performance by an order of magnitude. The architecture requires information flow amongst the agents proposed to be done through OAM cells. An information flow model using special cells (called management cells) is reviewed next. However these mechanisms are complicated, require new processing capabilities at switches and their accuracy may not justify the additional cost.

Next, a few mechanisms to measure the QoS parameters - delay, loss and errors are reviewed. Two fundamentally different schemes of correctly measuring delay using management cells and estimation of remote clock parameters are presented. CLR can be estimated using a fuzzy system (given extremes information about the behavior) and using regression on measurements on several pseudo-buffers. It has been proposed to measure error performance using parameters more sophisticated than BER but a proper justification about their applications has not been given.

A number of issues remain to be addressed in this topic. The first important requirement is the ability to accurately measure the parameters using a simple, fast, topology independent scheme that requires minimal changes to existing equipment. The ability to measure the bottle-necks, and other relevant diagnostic information should also be incorporated. A simple algorithm that dictates the next optimal time to sample also needs to be found. A constant sampling policy cannot be used due to the heterogeneity in sources, bandwidth constraints and congestion in the network.

The second question is about the value of the information gathered. These statistics will be potentially used by the control algorithms - CAC, routing, congestion management, UPC, VP control etc. How they are used, will define the value of these statistics, which in turn will dictate which parameters to measure, what accuracy is required and how often to measure. In other words, a strong coupling between traffic management functions and performance monitoring aspects is required.

Another important issue is the use of performance monitoring for proactive measurements and probing. For example, measurements on alternate routes can be made when there is no congestion to aid in routing and CAC. Such measurements will require very different schemes which again will be dictated by the potential use of this information.

References

[1] "B-ISDN Operation and Maintenance Principles and Functions", *ITU-T Recommendation I.610*, May 1995.

[2] D. Gaïti and G. Pujolle, "Performance Management Issues in ATM Networks: Traffic and Congestion Control", *IEEE/ACM Trans. on Networking*, vol. 4., no. 2, Apr 1996, pp.249-257.

[3] T. M. Chen et al., "Monitoring and Control of ATM Networks Using Special Cells", *IEEE Network*, Sept/Oct 1996, pp. 28-38.

[4] C. Roppel, "Estimating Cell Transfer Delay in ATM Networks Using In-Service Monitoring Methods", *Proc. Globecom '95*, pp. 904-908.

[5] B. Bensaou, T. C. Lam, H. W. Chu and H. K. Tsang, "Estimation of the Cell Loss Ratio in ATM Networks with a Fuzzy System and Application to Measurement-Based Call Admission Control", *IEEE/ACM Trans. on Networking*, vol. 5, no. 4, Aug 1997, pp. 572-584.

[6] H. Zhu and V. S. Frost, "In-Service Monitoring for Cell Loss Quality of Service Violations in ATM Networks", it IEEE/ACM Trans. on Networking, vol. 4., no. 2, Apr 1996, pp.240-248.

[7] N. Sato, K. Asatani, H. Murakami, R. Mallon, S. Hughes and T. L. Graff, "In-Service Monitoring Methods-Better Ways to Assure Quality of Digital Transmission", *IEEE JSAC*, vol. 12, no. 2, Feb 1994, pp. 355-360.