Adaptive QoS-Based API for ATM Networking

V. Bansal, R. J. Siracusa, J. P. Hearn, G. Ramamurthy and D. Raychaudhuri

NEC, USA, C&C Research Laboratories

4 Independence Way, Princeton, NJ 08540.

This paper describes work-in-progress on a new adaptive QoS-based application programming interface (API) for ATM networking. This ATM API, referred to as the "ATM Service Manager (ASM)" is motivated by the observation that current transport interfaces do not provide QoS features necessary for multimedia applications to achieve desired performance/cost objectives on multiservice ATM networks. In general, an application connected to ATM has a choice of transport protocols (e.g. TCP, UDP, alternative multimedia stream protocols), ATM service class (e.g. ABR, VBR, CBR) and QoS parameters for each media type. It is our view that application software should be shielded from the complexity of QoS-based service management by an adaptive API which is responsible for mapping the application requirements to ATM.

1 Design Objective for ATM Service Manager

Current transport network APIs do not provide mechanisms for the application to specify the QoS[1] that they can expect from the network. The ASM addresses this deficiency providing a new API and takes the responsibility of satisfying the applications QoS by using the following mechanisms:

Dynamic renegotiation of traffic and QoS parameters

The current notion that the negotiated QoS for a connection remains the same for the lifetime of the connection is not always valid for multimedia applications. Applications should be able to renegotiate QoS during the lifetime of the connection. Based on the quality perceived by the user, the application may want to change its QoS requirements, thus directly influencing the resources needed and the associated transmission cost. For example, in a video-on-demand application the user might want to change the quality (resolution) of the picture while the session is in progress.

Customized Transport Stack

The transport system architecture in this framework is a vertical process architecture [2], in which, a separate protocol stack is associated with each media belonging to a higher level connection. The advantage of this architecture is that, it takes into account the traffic characteristics and the QoS requirement of the connection. Each media has a separate customized transport stack based on its QoS requirements. For example audio streams have stringent real time requirements and do not require error control. Medical image retrieval applications on the other hand, require error free delivery even at the expense of some additional delay. Based on the knowledge of the media and its QoS requirement, the ASM determines the appropriate transport protocol options to be used for the media.

Characterization of Application Traffic Parameters

The network cannot provide guarantees on QoS without knowing the characteristics of the traffic it has to carry. On the other hand one cannot expect all applications to specify their traffic characteristics and the QoS requirement, since the application 1) may not know them in advance, 2) the characteristics may change during the life-time of the connection, 3) the applications may not be sophisticated enough to provide this information. Hence, system utilities are necessary that will monitor the traffic streams and determine their characteristics on a dynamic basis. The traffic characteristics are determined by making appropriate measurements on the traffic stream and mapping them into suitable stochastic or deterministic parameters.

Map application parameters to network parameters and vice versa

The set of higher level traffic and quality of service parameters specified by the application have to be mapped into a set of lower level network related parameters that the underlying network can interpret[3]. This feature provides a separation between the application and the network, and allows the application to specify these parameters in a language [4] that is most convenient and natural to the application. The QoS parameters specified at the application layer are used by the transport system to:

1) Map the parameters into a set of bounds on the delay and error the network can introduce. These parameters will be specified at connection setup time.

2) Take corrective actions during the protocol execution, to maintain the connection at an acceptable level of quality.

3) Inform the application (upper layer) in case of QoS degradation so that the application can take appropriate action.

Respond to network congestion notification signals

A control path between the ASMs and the signaling module allows dynamic QoS control, connection setup and modification, and exchange of network congestion messages. When congestion occurs in the network the signaling module receives congestion notification signals from the network which, after processing by the signalling module are forwarded to the ASMs. The ASM is programmed to take a variety of actions depending on the type of media it is serving. Actions taken by the ASM include renegotiation of bandwidth, traffic shaping, and even communication with the application to modify the generated traffic (e.g. change the quantization step in a video coder).

2 Prototype Implementation:

An outline of the structure of the ATM Service Manager software is shown in Fig. 1. The basic flow in this architecture is as follows: Higher layers (Media daemons and/or applications) request connection setup to the ATM Service Manager's *Connection Manage*r using the new APIs in which they specify the proper traffic and QoS parameters. Once the connection is setup the ASM's *QOS Monitoring module* estimates the quality-of-service of the connection using the quality-of-service reports that it receives and also the connections long term statistics. This module also takes different action if it decides that the QoS for the connection is not been met. Also the *Traffic estimation module*, estimates the traffic characteristics of the connection by continuously monitoring the data stream as it flows through the data plane. The *Renegotiation module* is invoked by either the QoS Monitoring module or the Traffic estimation module if they decide to renegotiate resources for the connection with the network.

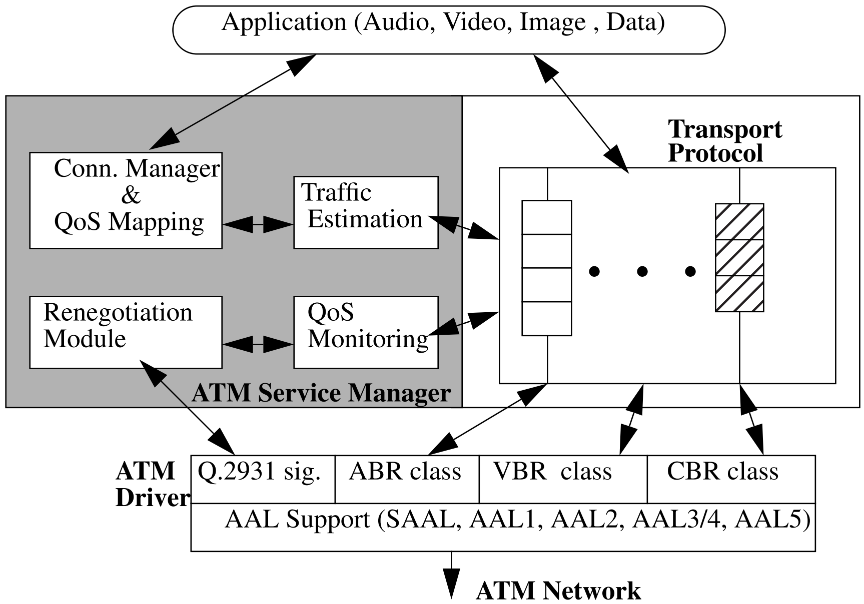

Fig. 1: Implementation model of ATM Service Manager

2.1 Connection Setup procedure

The connection setup procedure separates the control and data channel between the peers. The connection request from an application is intercepted by the connection management module of the ASM. It determines the traffic parameters and/or QoS parameters for the connection, and sets up an *ASM control* VC to the destination. Once the ASM Control VC is established, the source and the destination ASMs can communicate with each other and the application, and determine the actual traffic and QoS parameters (which either side of the connection might know). The ASMs can now determine the ATM service class and the type of AAL it must choose at the ATM layer, and the transport options that must be assembled at the transport layer. After the connection related QoS parameters and transport options are determined and mutually agreed upon by the two ASMs, the ASMs sets up the *data* VCs between the source and destination with the help of signalling module, assembles the required transport stack, on either end and then inform the application that it can commence transmission of data on that data VC. The ASM control VC between the corresponding ASMs is maintained throughout the life of the connection, thus enabling the applications and their respective ASMs to communicate with each other as and when the need arises.

2.2 QoS Monitoring

Monitoring the QoS of the connection is an important feature of this framework. The ASM provides QoS Monitoring by defining a QoS Report (QSR) packet that is exchanged between the two ASMs on either end of a connection. The VC Quality Monitoring mechanism measures packet data loss, packet delay and delay jitter. For example in a video server application where the ASM at the receiver end monitors the QoS on its inbound link, if the ASM detects any degradation in the QoS, it communicates this information through a QSR packet to the ASM at the server end. The ASM at the server end can now take appropriate action to restore the QoS on its outbound link. This could include, renegotiation for additional bandwidth, a change of class, or reduction in the resolution by switching to a lower bit rate. The VC Quality Monitoring process associated with the ASM sends QoS Reports periodically, as well as when there is a deviation of the QoS from a set threshold.

2.3 Transport options selection

Based on the QoS requirements of the application the ASM decides the transport options that must be configured for a connection. ASM decides which transport options are needed from the multimedia transport protocol (MTP) that we have implemented. Presently the transport options include:
Error Detection, Loss Recovery Mechanisms, Flow Control Mechanisms, Encryption, Segmentation and Reassembly and AALs.

There are also logical dependencies between different modules which would make pairing of different modules mandatory. Once the application's requirements are interpreted, the right set of transport modules for the connection can be configured. Below we list some transport options based on the condition of the link:

1. *Choose Error Detection*:
a) If the route to the destination is not reliable and bit errors can occur frequently.
b) If the route to the destination go over non-ATM links
2. *Choice of Loss Recovery Mechanisms*:
a) Choose Selective repeat is used if the cell loss/error rate of the VC is low.
b) Choose go-back-n is used if the error loss/error rate is high.
c) Choose loss detection only, if application wishes to do its own error recovery.
3. *Flow Control Mechanisms*: Use pacing or rate control at the packet level to prevent receiver over run.
4. *Encryption*: If application requires security features.
5. *Segmentation and Reassembly*: Use fixed length TPDUs if application requires the frame boundaries to be preserved. Normally fixed TPDUs are required when the application performs high level error recovery.
6. *ATM Adaptation Layer selection*:
a) If application requires CBR service and has delay stingent requirements use AAL1
b) If application requires VBR service and has delay stringent requirements use AAL2
c) else use AAL5 (AAL3/4 in cases where use of MID field is required)

References

[1] Andrew Campbell, Geoff Coulson and David Hutchison, " A Quality of Service Architecture", Computer Communication Review, April 1994.

[2] Douglas C. Schmidt and Tatsuya Suda. "Transport System Architecture Services for High-Performance Communication Systems", IEEE JSAC, May 1993.

[3] Nahrstedt, K. and J. Smith, "Revision of QoS Guarantees at the Application/ Network Interface", Technical Report, University of Pennsylvania, 1993.

[4] Patricia G.S Florissi and Yechiam Yemini , "QuAL:Quality Assurance Language", ITS'94.