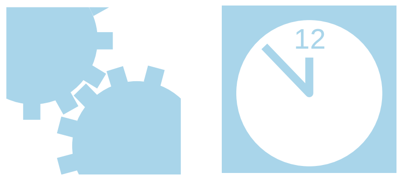


INQIRE: A Software Approach to Monitoring QoS in ATM Networks

Thomas M. Chen, Southern Methodist University
Steve S. Liu and Michael J. Procanik, GTE Laboratories Inc.
David C. Wang, GTE Telephone Operations
D. Dean Casey, GTE Laboratories Inc.

Abstract

Unlike traditional best-effort packet networks, ATM networks support a wide range of services by providing a specified QoS on each ATM connection. Currently, QoS can be measured with specialized high-speed testing equipment but their complexity and cost prevent them from wide field use. This article describes the INQIRE project approach based on software for enabling common personal computers (with an ATM network interface card) to function as QoS monitoring stations at the edges of an ATM network. The main objective is to develop a low-cost, off-the-shelf alternative to broadband testing equipment. An INQIRE station, consisting of a PC running the INQIRE application, can actively probe any selected connection and collect sample measurements of the QoS. The monitoring approach is intended to be non-intrusive and work with any ATM network without requiring any modifications to existing ATM switch equipment or interruptions to active services.

Asynchronous transfer mode (ATM) is a connection-oriented, fast packet switching protocol adopted for broadband integrated services digital networks (B-ISDN) [1, 2]. All types of voice, video, and data information are carried in 53-byte ATM cells consisting of 5-byte headers and 48-byte information fields. Unlike traditional best-effort packet networks, ATM networks can support a wide range of services by providing a specified quality of service (QoS) for each virtual connection. As listed in Table 1, QoS parameters characterize the user-perceived end-to-end network performance per connection. Some QoS parameters, such as cell loss ratio and maximum cell transfer delay, are negotiated explicitly during connection establishment, whereas other QoS parameters may be implicitly assumed or unspecified. The particular set of QoS parameters specified for a connection depends on the type of service (e.g., constant bit rate, variable bit rate, or available bit rate).

Through the call admission control procedure, a subscriber signals a request for a new connection to the network, and the network makes a decision to accept or block the new connection depending on the requested QoS and the available network resources. Generally, a new connection will be accepted if the network estimates that the available resources will be sufficient to satisfy the requested QoS and throughput. If the new connection is accepted, the network implicitly agrees to a so-called traffic contract with the responsibility of sustaining the QoS for the connection as long as the source traffic is conforming to its declared traffic characteristics [2].

The traffic contract concept underlying ATM services raises the practical question of how the actual QoS may be observed and verified. The QoS is directly observable to users who can measure the end-to-end performance of their connections by means of information exchanged in the ATM adaptation layer (AAL) or higher protocol layers. For example, users can exchange timing information to measure the end-to-end delay. However, the QoS is not directly observable to network providers for a number of reasons. First, the network provider has visibility only into the cell header but not into the cell payload. However, it is not possible to make use of the ATM cell header because it has been designed to be simple for fast cell relaying. Cell header fields that would be useful for performance monitoring, such as cell sequence numbers and timestamps, have been deliberately omitted to reduce the processing burden for ATM switches. Second, QoS monitoring is a responsibility of network management, and network management systems can poll ATM switches for their performance data, typically using Simple Network Management Protocol (SNMP) [3]. However, network management information is oriented to localized performance statistics for individual switches, and does not include end-to-end QoS statistics. Third, specialized high-speed testing equipment is commonly used to test connections, but broadband testers are typically expensive, complex to use, and intended more for laboratory testing than wide field use.

In the longer term, an operations and maintenance (OAM) method for in-service performance monitoring has been standardized [4]. The method involves the insertion of OAM performance management cells between blocks of user cells. ATM switches must recognize the OAM cells and relay them with the user cells. An ATM switch performing the role of an endpoint in the procedure will terminate and process the OAM cells (involving an error check, timestamping, and counting cells) and report the monitoring results in an OAM cell in the backward direction. Due to the additional processing and cell handling complexity involved in the OAM procedure, most current ATM switches do not support these capabilities yet.

Characteristic	QoS parameter
Accuracy	Cell error ratio
	Severely errored cell block ratio
	Cell misinsertion rate
Speed	Cell transfer delay (maximum, mean)
	Cell delay variation
Dependability	Cell loss ratio

Table 1. *QoS parameters.*

Before OAM performance management capabilities become commonplace in future ATM switches, there is an immediate need for a flexible, easily fieldable tool for in-service monitoring of end-to-end QoS. In-service monitoring means that QoS can be measured without interrupting services. This article describes the INQIRE (In-Service QoS Remote Edge Instrument) project that has developed software for enabling common personal computers (with ATM network interface cards) to function as QoS monitoring stations at the edges of an ATM network. The main objective is to develop a low-cost, off-the-shelf alternative to broadband testing equipment. An INQIRE station, consisting of a PC running the INQIRE application, can probe any selected route with test cells and collect sample measurements of the QoS. The in-service monitoring approach is non-intrusive, and works with any existing network of ATM switches without requiring any special switch functions. The next section describes the monitoring approach used by INQIRE. We then outline the modular architecture of the INQIRE software and the functions of each module. Finally, performance and implementation issues are discussed.

The INQIRE Approach

As mentioned earlier, users at the endpoints of an ATM connection can naturally observe the actual QoS they are receiving by exchanging end-to-end information through the AAL or higher protocol layers. On the other hand, network providers are not able to observe QoS directly due to several constraints. First, most current ATM switches do not have many capabilities beyond cell relaying, such as OAM performance management functions. Hence, QoS monitoring cannot depend on sophisticated processing or cell handling functions within the network. Second, network management systems can monitor network performance by polling data from switches, but this paradigm is oriented toward localized performance data for individual switches, not end-to-end performance data. In addition, current network management systems are typically proprietary systems that do not readily interoperate with each other, which makes end-to-end monitoring even more difficult. Third, monitoring should be in-service and non-intrusive, meaning that user services should not be interrupted and active connections should not be invaded with test traffic.

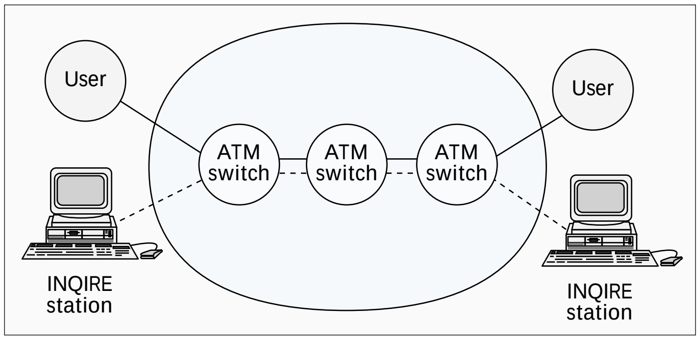


Figure 1. *INQIRE stations use a parallel virtual connection with the same route as the user connection.*

The INQIRE approach has been developed for in-service performance monitoring of ATM connections without requiring any special functions in ATM switches. It is a software-based approach involving INQIRE stations situated around the network edges. INQIRE stations are based on a PC platform to be low-cost and easily fieldable. An INQIRE station consists of a common PC equipped with an ATM network interface card and the INQIRE software to perform all necessary functions for monitoring:
- Generating, receiving, and processing test cells
- Collecting and analyzing QoS sample measurements
- Graphically displaying statistical data

INQIRE stations can actively probe the network with test cells from the network edges. From the perspective of the network, the test cells are carried like other user cells through the network without requiring any special handling by ATM switches. Processing of test cells is required only outside of the network at the INQIRE stations. In order to measure cell delay and cell loss, these test cells contain special information fields in their payloads including:
- Test cell type/function
- Test cell sequence number
- Sender timestamp
- Receiver timestamp
- Remote station processing time

Unlike specialized broadband testers, PCs are currently incapable of sufficiently high-speed performance to directly insert test cells into a selected user connection and extract them. Therefore, in order to monitor a selected connection, test cells are sent along a parallel virtual connection (VC) that has been established with the same route and QoS class as the selected VC. An example configuration involving two INQIRE stations at both ends of a selected route is shown in Fig. 1. In this example, the user cells and test cells traverse separate virtual channels within the same virtual path (VP) connection. The current version of INQIRE relies on permanent VCs and VPs that are pre-established by network managers.

The approach takes advantage of the fact that the ATM switches will statistically multiplex the test cell stream and user cell stream together. Hence, the test cells will experience a QoS similar to that seen along the selected user connection. It is important to have the same route and service class because the test cells should see the same switch buffers as the user cells of the monitored connection. Because both connections share the same route and service class, the test cells essentially sample the QoS seen by the other connection. The INQIRE approach relies on statistical sampling to avoid the need for high-speed cell insertion/extraction in the user cell

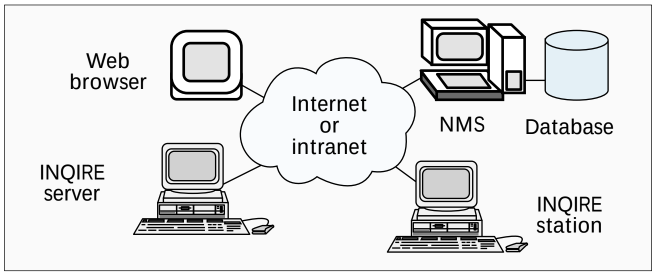


Figure 2. *A network management system and an INQIRE server accessed via a Web client.*

stream. The test cells are transmitted at a relatively low rate so that the network performance is not significantly perturbed, but even a relatively low 5 kb/s rate would allow 10 test cells/s. It is not the objective of INQIRE to detect every congestion period, which may be too brief to detect with low-frequency sampling; the objective is to measure QoS which is defined over the long term. At the opposite end of the route, the second INQIRE station will loop back the test cells to the first station. Naturally, a second station would not be necessary if a cell loopback can be configured in the last ATM switch in the route, but only round-trip delays could be measured in this case.

Ideally, at least one station is situated at each access switch so that any route in the network can be probed with two stations at the endpoints. These stations may be controlled from the graphical user interface at each console, or they may be remotely controlled from an INQIRE server. In the case of remote control, the INQIRE server provides a Web interface to receive user input and display data to users. It also sends special control messages to INQIRE stations around the network to initiate monitoring on user-selected connections. During monitoring, the stations send their collected QoS measurement data back to the server for display via the Web interface.

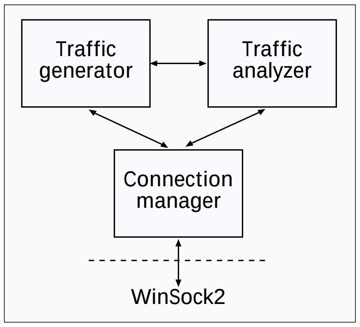


Figure 3. *INQIRE software modules.*

INQIRE is intended to work in combination with a network management system (NMS) to augment the NMS with end-to-end monitoring data that it would not collect otherwise. INQIRE stations and servers can work with an NMS as shown in Fig. 2. As usual, the NMS polls individual switches for their local data, and in this example it is assumed to maintain a database of management information. The management information is typically related to localized switch performance and integrity, not end-to-end QoS. With Web server capabilities, the NMS can make the management data viewable to Web browser clients through the Internet or an intranet. At the same time, INQIRE stations involved in monitoring sessions will export their QoS measurement data into the NMS database. The NMS makes the QoS data available to Web clients in addition to the regular network management data. In order to minimize complications to existing NMSs, the NMS receives INQIRE data into its database but may not allow users to initiate and control monitoring tests. In that case, the NMS can be a passive recipient of unidirectional information from INQIRE stations.

Users can initiate and control monitoring tests through a Web interface to the INQIRE server. The INQIRE server is able to receive form input data from users to initiate monitoring. During monitoring, the server receives real-time QoS measurement data from stations and exports this data to Web clients. The transfer of information is bidirectional between the INQIRE server and stations. The QoS data may be logged long-term at the INQIRE server or at each station, depending on which location is better considering processing loads and available storage space. The current version of INQIRE chooses to log data at each station in order to relieve the server of the burden of archiving data from potentially many stations.

Software Architecture

INQIRE Station Software

The INQIRE software enables a PC to perform these basic functions required for monitoring:
• Generating and receiving test cells
• Calculating and analyzing network QoS measurements
• Graphically displaying real-time measurement data and accumulated QoS statistics
• Displaying alarms for baseline QoS violations and connection faults
• Archiving the QoS measurement data

The INQIRE application offers a custom graphical user interface, a simple command line interface, or a Web interface. The software architecture has a modular design consisting of three main modules, as shown in Fig. 3: the connection manager, traffic generator, and traffic analyzer. Communication between the modules is conveyed through internal control messages.

Essentially, the *connection manager* module is responsible for:
• Coordinating the timing with the other modules
Initiating and terminating monitoring tests
• Transferring QoS data to the NMS database
• Handling the main graphical user interface when tests are initiated at the console

An example of the user interface for setting up tests is shown in Fig. 4. A monitoring test can be initiated from this interface after the appropriate configuration parameters have been selected (e.g., route, starting time, duration of test, test cell generation rate). When a test is initiated, the connection manager will verify that sufficient hard disk space is available on the INQIRE station to archive the entire set of data expected from the test. If the disk space is available, the test will proceed automatically; otherwise, the user is provided with an opportunity to delete old files to free up disk space before the test proceeds.

Additionally, the connection manager handles parameter values for a “baseline QoS” which is the QoS level expected under normal conditions. The baseline QoS parameter values (e.g., cell transit delay, cell loss ratio) can be specified by the user or set to predefined defaults for a service class. During a monitoring test, the baseline QoS parameters will be used by the traffic analyzer module to detect cases of QoS violations.

During tests, the connection manager in an INQIRE station will periodically transfer the QoS measurement data from the station to an NMS database every 15 min. The current version of INQIRE can interface with any database management system using the Microsoft open database connectivity (ODBC) standard.

When multiple INQIRE stations are involved in a monitoring test, the connection manager modules in the respective stations exchange status information before and during the test. During monitoring, the connection manager checks all incoming cells including test cells, keep-alive cells (sent periodically to verify connectivity), and other status information cells from a peer station. All incoming test cells are time-stamped. The connection manager will react to incoming status information cells to issue internal control messages to the other modules as needed to carry out the appropriate processing.

If an INQIRE server is used for the remote control of a monitoring test, the connection manager module has additional responsibilities to communicate with the server. It will perform a security check on all server commands, interpret and process any server commands, and report all status information, including QoS measurement data and QoS violations, to the server.

The *traffic generator* module is responsible for composing the contents of outgoing test cells and passing them to the connection manager for transmission into the network. The test cell generation rate, traffic profile, and total number of test cells are some of the parameters specified by the user during test setup. The test generator module maintains outgoing cell counts and inserts a timestamp into the payload of each outgoing test cell.

During monitoring, the traffic generator module performs various functions in the background:

- Keeps track of the test progress and reports the end of the test to the other modules
- Periodically sends "keep-alive" cells to verify connectivity
- Carries out requests from the other modules (e.g., PAUSE/RESUME commands)
- Reports outgoing cell counts to the traffic analysis module for display purposes

If the INQIRE station is looping back test cells during monitoring, the traffic generator module provides the loopback conduit for incoming test cells.

Finally, the *traffic analyzer* module has three main functions: calculating QoS statistics from collected sample measurements, creating database files, and providing a graphical user interface for controls and statistics/alarms display. This module calculates the transit delays of all incoming test cells, detects lost test cells, and calculates statistics such as maximum and average cell delay, cell delay variation, and cell loss ratio. It compares the QoS sample measurements with the baseline QoS and displays alarms for QoS violations and faults (detected by continual loss of cells).

All test results and statistics are archived in database files in an INQIRE station. A database entry records all information pertinent to every test cell such as the test cell number, transmission time, cell transit time, return time, and other information. If needed, these ASCII database files allow for later off-line statistical analysis.

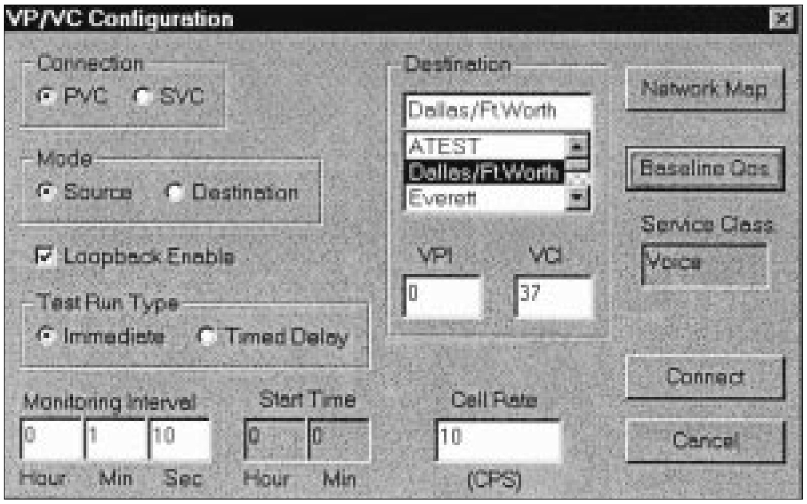
The traffic analyzer module provides a GUI for the user to exercise certain controls during a monitoring test. The controls allow the user to pause, resume, and terminate a test at any time. Furthermore, the QoS measurement data is displayed during the progress of a test. Displayed information includes a real-time plot of cell delay measurements and up-to-date QoS statistics, and a cumulative histogram analysis of cell transit delay.

The INQIRE Server

As mentioned earlier, INQIRE stations can be remotely controlled by a server. The server provides a Web interface for user input and display, and sends special control messages to INQIRE stations around the network to initiate monitoring on user-selected connections. During monitoring, the stations send their collected QoS measurement data back to the server for display via the Internet or intranet. An INQIRE server can be run on a standalone server or an INQIRE station; in the latter case, the same PC is both an INQIRE station and server.

The server software consists of two modules: a controller module and a common gateway interface (CGI) module. The controller module handles the remote control of dispersed INQIRE stations, coordinates the initialization and completion of monitoring tests, and displays the real-time QoS measurement data and QoS violation alarms reported from the stations. The server communicates with stations using UDP/IP. A proprietary session protocol is implemented above the UDP protocol to ensure that every received UDP packet comes from an authorized station involved in an active monitoring test.

The CGI module allows authorized users to interact with the INQIRE server through Web browsers. The CGI module is the interface between the Web server and the INQIRE application. It takes user data from Web forms as inputs to the INQIRE application (e.g., to start a new monitoring test) and translates the QoS data collected from the stations into browser-viewable form.



VP/VC Configuration

Connection: [✓ PVC] SVC

Mode: [✓ Source] Destination

[✓ Loopback Enable]

Test Run Type: [✓ Immediate] Timed Delay

Destination: Dallas/Ft.Worth
ATEST
Dallas/Ft.Worth
Everett

VPI: 0 VCI: 37

Service Class: Voice

Monitoring Interval: 0 Hour 1 Min 10 Sec

Start Time: 0 Hour 0 Min

Cell Rate: 10 (CPS)

Figure 4. *The user interface for setting up a monitoring test.*

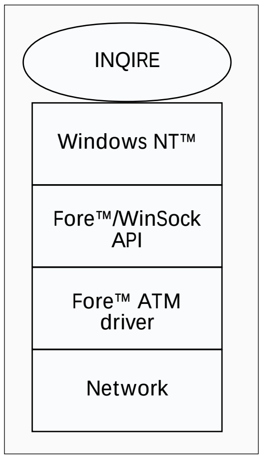


Figure 5. *Relationship of protocol layers.*

Performance and Implementation Issues

The Operating System

Currently, the prototype INQIRE software is implemented in C++ incorporating Microsoft Foundation Classes (MFC). Figure 5 shows the relationship between the INQIRE application and the operating system and network. Because Windows NT™ is a multitasking environment and not a real-time operating system, any time-critical measurements are restricted by the nature of the environment. We have found that other tasks such as user applications, screen savers, disk caching or even mouse movements are allocated CPU time. This multitasking environment causes contention with the INQIRE application for CPU cycles, which becomes problematic because INQIRE timestamps outgoing cells immediately prior to their transmission and incoming cells after they are received. The resource sharing of the operating system may result in inaccuracies when timestamps are calculated and inserted into each cell payload. Occasionally, an incoming cell will find that its time-

Controlled end-to-end cell delay	Cell delay parameter	INQIRE measurements (ms)	H-P tester measurements (ms)	Absolute difference (ms)	Percent difference
1.5 ms	Maximum	1.597	1.521	0.076	5.0
	Minimum	1.572	1.517	0.055	3.6
	Mean	1.584	1.519	0.065	4.3
15 ms	Maximum	15.120	15.032	0.088	0.6
	Minimum	15.083	15.028	0.055	0.4
	Mean	15.101	15.030	0.071	0.5
150 ms	Maximum	150.101	150.024	0.077	0.05
	Minimum	150.069	150.020	0.049	0.03
	Mean	150.085	150.022	0.063	0.04

Table 2. *INQIRE measurements of cell delay in a CBR connection benchmarked against H-P broadband tester measurements.*

stamp is not calculated immediately, or an outgoing cell may be timestamped and then find its transmission delayed due to CPU cycles spent on another task. Another factor contributing to timing inaccuracy is the extra API layers introduced by WinSock2 and its service provider driver.

Fortunately, the timing issues can be effectively resolved by a couple of measures. First, we can assign "above normal" priority to INQIRE's threads and additionally restrict the number of extraneous tasks running during a monitoring test. Second, we can move the timestamping function to the lowest-level adapter driver (bypassing the operating system). Thus, timestamping is performed immediately before an outgoing cell is handed to the adapter firmware for transmission or immediately after an incoming cell is released by the firmware. The current version of INQIRE is being reworked with both of these measures to improve timestamping accuracy.

Some measurements are listed in Table 2 to compare the accuracy of measurements made by the current INQIRE version and an H-P broadband tester. Both were used to measure the cell delays in a constant bit rate ATM connection with controlled end-to-end delay. The Hewlett-Packard (H-P) tester was chosen for comparisons due to its exemplary accuracy. From the benchmark measurements, it may be seen that INQIRE measurements were within an accuracy of 50–90 µs or 5 percent. This accuracy might be further improved if benchmark measurements can be made before any monitoring, and the average skew subtracted from the INQIRE measurements for calibration (e.g., if a premeasured skew of 70 µs was subtracted a priori from the INQIRE measurements in Table 2, the accuracy might have been improved to ± 20 µs).

Clock Synchronization

INQIRE stations make use of timestamps carried within the test cells to make cell delay measurements. When a single station uses a cell loopback at the far end to measure round-trip cell delays, there is no need for clock synchronization. However, one-way cell transfer delays are more interesting and may not be estimated accurately from round-trip cell delays (e.g., half of the round-trip delay because traffic in the two directions may be asymmetric). In order to measure one-way cell delays, two INQIRE stations are used; the second station at the far end of a connection adds another timestamp to the test cells prior to returning them in the backward direction. The one-way cell delay measurements will have meaning only if the real-time clocks of the INQIRE stations are synchronized to the same time of day.

Various methods are available for clock synchronization. For example, Global Positioning System (GPS) receivers can synchronize ground systems to a constellation of orbiting satellites to within 10 ns or better [5]. Another possibility is the network time protocol (NTP) to synchronize INQIRE stations to Internet time servers [6]. The accuracy depends on the stratum level of the time servers and the method of connection to them. A third possibility is to synchronize INQIRE stations to a common INQIRE server through circuit-switched connections. Each INQIRE station could set up a circuit-switched link to the INQIRE server, and send a message to the server. The server timestamps the message and returns it to the station. The station can deduce the propagation delay as half the round-trip delay, and add the propagation delay to the timestamp to determine its proper clock time. This is a simple method to implement using the telephone network.

Statistical Sampling

As mentioned earlier, off-the-shelf PCs are incapable of high-speed test cell insertion/extraction directly in a user cell stream. Therefore, the INQIRE approach relies on relatively low-rate random sampling of the QoS. Each test cell carries a timestamp to make a measurement of the cell transit delay. Then an empirical cell delay probability distribution function may be estimated from the collected cell delay samples. Each test cell also carries a sequence number to detect cell loss. The cell loss ratio can be estimated from the ratio of lost test cells. As with any statistical technique, the accuracy of the data results is an important question.

For cell delay measurements, we may be interested in confidence bounds for the estimated probability that cell delays will not exceed a given maximum delay D. Traditional estimation theory can be applied in a straightforward manner if the time between test cells is relatively long and each test cell is assumed to take an independent random sample from a stationary cell delay probability distribution. If p is the actual probability that cell delays will not exceed D, then the usual estimator $\hat{p}$ is the fraction of cell delay samples found to be less than or equal to D. If the assumptions are valid, it is known that $\hat{p}$ will be an unbiased and consistent estimator. When the number of samples N is large, $\hat{p}$ asymptotically converges to a normal probability distribution with mean p and variance $p(1 - p)/N$. Confidence intervals for a normally distributed estimator can be calculated easily; for example, the 95-percent confidence interval is

$$p \pm 1.96\sqrt{p(1-p)/N}.$$

This gives an indication of the accuracy of the estimator $\hat{p}$ for a given sample size N. Or turning this around, one can ask for the required number of samples to achieve an error accuracy of $|p - \hat{p}| \leq e$ with probability 0.95. Noting that $p(1 - p)$ can be 0.25 at the most, we find that conservatively $N \geq 0.25e^{-2}$ samples would be needed for given e.

The test cell rate will depend on the monitoring interval and desired error accuracy. The worst case is obviously a short monitoring test with stringent error accuracy, because it will be difficult to collect enough samples within a short time interval. As an example, if a monitoring test is constrained to be 10 min with a desired error accuracy of $e \leq 10^{-2}$, we would need 2500 samples or 4 test cells/s. If the desired accuracy is changed to $e \leq 10^{-3}$, we would need (conservatively) 250,000 samples or 400 test cells/s. If great accuracy is needed, it would be better to lengthen the monitoring interval if possible. For example, if the monitoring test could be relaxed to a longer 60 min, we would need a modest 70 test cells/s for an accuracy of $e \leq 10^{-3}$.

For cell loss ratio, the application of estimation theory is also straightforward under the assumptions that test cells experience the same cell loss conditions as user cells, and each test cell is lost independently and randomly. If q is the actual cell loss probability, and $\hat{q}$ is the estimator based on the proportion of test cells that are lost, then $\hat{q}$ is unbiased, consistent, and asymptotically normal with mean q and variance $q(1 - q)/N$. The 95-percent confidence interval is

$$q \pm 1.96\sqrt{q(1-q)/N},$$

or alternately, we would need conservatively $N \geq 0.25e^{-2}$ samples to achieve an accuracy of $|q - \hat{q}| \leq e$. In practice, q and e could be very small (on the order of 10^{-4} or less), implying that virtually no cells should be lost. In that case, it may be impractical to try to estimate the cell loss probability with any accuracy within a reasonable length of time. Instead, the loss of a test cell can simply be used as an indication of possible trouble in the network.

Conclusions

INQIRE is a flexible software-based alternative to specialized broadband testing equipment for measuring and evaluating QoS in high-speed ATM networks. The INQIRE approach avoids the need for high-speed cell insertion/extraction capabilities by taking advantage of statistical sampling and the statistical multiplexing inherent in ATM switches. As software running on common PC platforms, there are certain performance issues related to the operating system, but it is believed that these issues can be minimized by restricting extraneous tasks and possibly bypassing the operating system with customized driver software.

It should be recognized that the statistical nature of the approach raises a more serious difficulty. As a statistical technique, the accuracy of the collected QoS data depends inherently on the number of measurement samples or length of monitoring. While a small sample of measurements may provide a quick and rough indication of the network QoS, more confidence can be invested in the data with more samples or longer monitoring intervals. This may be an unavoidable limitation of the approach, balanced on the other hand by the cost and convenience advantages of using common PC platforms.

References

[1] ITU-T Rec. I.121, "Broadband Aspects of ISDN," Melbourne, Australia, Nov. 1988.

[2] ATM Forum, "Traffic Management Specification Version 4.0," ATM Forum/95-0013R10, Feb. 1996.

[3] J. Case, *et al.*, "A Simple Network Management Protocol (SNMP)," Internet RFC 1157, May 1990.

[4] ITU-T Rec. I.610, "B-ISDN Operation and Maintenance Principles and Functions," Geneva, Switzerland, July 1995.

[5] W. Lewandowski and C. Thomas, "GPS time transfer," *Proc. IEEE*, vol. 79, July 1991, pp. 991-1000.

[6] D. Mills, "Precision synchronization of computer network clocks," *Comp. Commun. Rev.*, vol. 24, Apr. 1994, pp. 28-43.

Biographies

THOMAS M. CHEN [SM] (tchen@seas.smu.edu) is an associate professor in the Department of Electrical Engineering at Southern Methodist University, Dallas, Texas. He received B.S. and M.S. degrees in electrical engineering from MIT, and a Ph.D. degree in electrical engineering from the University of California, Berkeley. From 1989 to 1997 he worked on ATM research at GTE Laboratories, Inc., Waltham, Massachusetts. He is a technical editor for *IEEE Network* and *IEEE Communications Magazine*. He was the co-recipient of the IEEE Communication Society's 1996 Fred W. Ellersick best paper award. He is co-author of the monograph *ATM Switching Systems* (Artech House, 1995) and co-contributor of a chapter in the upcoming *ATM Handbook* (McGraw-Hill).

STEPHEN S. LIU [SM] (sliu@gte.com) received B.S. and Ph.D. degrees in electrical engineering from National Cheng-Kung University, Taiwan, and Georgia Institute of Technology, Atlanta, respectively. He co-developed the ISO and ANSI standard 32-degree error detection code polynomial used on Ethernet and various high-speed data networks. He joined GTE Laboratories, Inc. in 1981 and has since been working on packet-switching technology. He is a principal member of technical staff in the Backbone Technologies Department. His current interests are in PSTN evolutions and ATM performance monitoring and control.

MICHAEL J. PROCANIK (mprocanik@gte.com) graduated valedictorian from Devry Technical Institute, Woodbridge, New Jersey. He received his B.S.E.E. from Northeastern University, Boston, Massachusetts. In 1985 he joined AT&T Bell Laboratories, Murray Hill, New Jersey, where his work focused on characterizing submicron high-speed digital NMOS integrated circuits. These ICs included a mux/demux chipset operating between 2 and 3 GHz. In 1986 he received AT&T's Exceptional Contribution Award for this work. In 1988 he joined GTE Laboratories, Inc. where he has been involved in various opto-electronic SONET/ATM data links and systems. He participated in the implementation and testing of the VISTAnet network for which he received the Leslie H. Warner Technical Achievement Award, GTE's major technical award, in 1994. His current research interests are the translation of service QoS requirements into ATM network performance characteristics and methods for providing effective QoS guarantees on an ATM framework.

DAVID C. WANG [SM '91] (david.wang@telops.gte.com) received his B.S. from National Cheng-Kung University, Taiwan, and M.S. and Ph.D. from Carnegie-Mellon University, Pittsburgh, Pennsylvania, all in electrical engineering. Currently he is architecture scientist at GTE Telephone Operations in Irving, Texas. He is responsible for the specification and design of broadband network architectures. He also serves as an adjunct professor at the University of Texas at Arlington, where he teaches a course on advanced data networks. Prior to joining GTE, He was a member of technical staff at AT&T Bell Laboratories for 10 years. He was involved in the performance specification, testing, and modeling of several networking technologies.

D. DEAN CASEY [M] (dcasey@gte.com) is manager of the Backbone Network Technologies Department at GTE Laboratories. His department is currently engaged in research on broadband network switching, interworking of IP and ATM services in wide area networks, transport of compressed video on ATM networks, and network performance monitoring for active network control. He received his Ph.D. in physics in 1973 from Washington State University and has been at GTE Laboratories since 1983. During his career, he has conducted and/or managed research in a wide variety of fields, including surface physics and semiconductor and display device fabrication, as well as in broadband network technologies.