Appeared in: Proceedings of the 1st USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara, CA, April 1999

Real-Time Intrusion Detection and Suppression in ATM Networks

R. Bettati W. Zhao D. Teodor

Department of Computer Science
Texas A&M University
College Station, TX 77843-3112

Abstract

Distributed mission critical systems require support for ultra-secure communication, in which intrusions must be detected and suppressed in real time, possibly before the affected messages reach the receiver. When the distributed application has real-time requirements, the effects of intrusion are particularly severe. In addition to covered channels and potentially tampered data at the receiver, such systems may experience violations of timing requirements and timing instabilities in components not directly related to the intrusion. Systems with real-time requirements have admission and access control mechanisms in place to ensure that timing requirements can be met during normal operation. Such admission control mechanisms require load profiles of traffic (for example in form of leaky bucket descriptors) so that resources can be appropriately allocated to meet application requirements during system operation. In this paper, we report on our project aiming at real-time detection of intrusions in ATM networks. We take advantage of the specification of the traffic profile during connection setup, and use a traffic modeling technique to determine the profile of the traffic on the connection in an arbitrary point in the network, thus providing a base line for detection of load deviations. We designed and analyzed a security device that uses the profile information, detects violations. The traffic is modeled in an accurate but efficient manner. As a result, our device is able to detect an intrusion within 25 μs, yet is simple enough to be economically realized in existing VLSI technology.

1 Introduction

High-performance networks with support for Quality of Service (QoS), such as Asynchronous Transfer Mode (ATM), are increasingly being deployed to support distributed mission-critical computing, at shipboard level or at wider scale [1]. For example, ATM technology provides the backbone for various core technology subsystems of the SmartShip program for AEGIS class cruisers [11], such as integrated condition awareness system, damage control system, machinery control system, and integrated bridge system. The networks used in many of these systems must meet stringent timing and security requirements. In this paper, we report on our project aiming at providing real-time intrusion detection for these types of networks. By real-time detection, we mean that a solution should detect and suppress network intrusions within very short time periods, say, 100 μs.

In addition to its high speed, ATM's ability to provide QoS support to users makes it increasingly popular for many such systems. While QoS for a connection can be characterized by many parameters, for real-time applications it is bandwidth guarantees and delay bounds that are perhaps the most important parts of the QoS specification. Unfortunately, relying on bandwidth or delay guarantees makes this type of systems very vulnerable to denial-of-service attacks, in addition to traditional intrusions. Indeed, as with other types of networks, potential attacks in an ATM network include the modification of connection and path data in a switch in ways that are beneficial to the attacker. In this way, the attacker would be able to insert, divert, or delete traffic in an unauthorized manner. Although such attacks can be local in nature, they can have a global impact by affecting not only the attacked connections, but other connections as well. For example, localized or intermittent flooding by an intruder can cause the network to violate the QoS require-

ments of many unrelated connections. This may in turn cause applications to time out, and the effect may range from invocation of timing recovery actions to total loss of system control. Thus, the damage can be widespread and very serious for a mission critical system and, hence, must be confined in real-time.

Deleting and suppressing flooding by intruders is difficult to achieve effectively at switch or network level, as it may easily masquerade as "friendly" traffic. Detection approaches therefore often have to rely on end-to-end mechanisms with very long latencies.

During normal operation, connections in networks with support for QoS guarantees need to go through a connection *establishment* phase. The new connection specifies its QoS requirements along with a characterization of the amount of traffic that it will carry. The admission control component of the system will then determine whether enough resources are available to satisfy the requirements of the new connection without violating guarantees of previously established connections. Once the connection is established, a policing mechanism typically enforces that the sender adheres to the traffic specification defined at establishment time. If an appropriate traffic model is used, and a sufficiently detailed traffic specification is provided at connection setup time, both can be used to accurately profile traffic during the lifetime of the connection. The traffic model should be capable of describing the traffic generated at the source as well as the traffic at an arbitrary point within the network. In an ATM network, traffic belonging to different connections gets repeatedly multiplexed and demultiplexed at the entrance to the network and in the switches. Consequently, the traffic pattern of a connection undergoes several changes as it traverses the network. The traffic pattern of a connection inside the network may be substantially different from its pattern at the source. In particular, it differs substantially from the traffic specification provided during connection setup time.

An important contribution of our work is a traffic model that very accurately characterizes traffic flows in a network, and so allows for the definition of accurate *traffic descriptors*. We will describe in Section 2.2 how we use *maximum* and *minimum traffic functions* to define an envelope on the amount of traffic generated by a sender or a set of senders in a distributed application. As we will demonstrate, these functions are powerful enough to describe all types of traffic encountered in time-critical applications, both at the sources and inside of the network. At the same time, these mathematical functions are concise and easy to manipulate.

Based on the traffic modeling techniques developed, we design and analyze a security device that uses traffic information to detect intrusions. The device meets the ATM forum UNI data specification. An evaluation shows that it is able to perform covert network traffic detection, suppression, and alert in a timely fashion (within 25 μs) even under peak traffic conditions. Its implementation is both cost effective and stable.

The rest of this paper is organized as follows: In Section 2, we introduce our traffic modeling techniques. The design and analysis of the security device is presented in Section 3 while Section 4 concludes the paper with final remarks.

2 Traffic Modeling

In order to provide a valid characterization of the traffic of a connection anywhere along the path of the connection, the traffic model must be flexible enough to capture perturbances as the traffic travels along its path. Also, the network must be modeled as to capture the elements that add perturbance to the traffic as it flows through the network. To model the traffic we use pairs of deterministic traffic bounding functions. To capture the active elements within the network we model it as a network of servers and distinguish between constant and variable servers.

2.1 The Network

For our purposes, an ATM network consists of ATM switches connected by communication links. An ATM switch itself consists of input ports, the switching fabric, and output ports. A cell that arrives at an input port of a switch is transported by the switching fabric to an output port, where it is transmitted along the physical link associated with the output port. Messages are segmented into fixed-size cells. This simplifies the traffic analysis because the cell transmission time is constant, and time can be normalized appropriately.

For the purpose of traffic analysis, the network is traditionally decomposed into a collection of *servers* [4]. Each server provides an abstraction for a network component in the system. For example, the input ports, the switching fabric, the output ports, and the physical links can each be modeled as a server.

We distinguish two types of servers: constant servers and variable servers [4, 9]. *Constant servers*, such as physical links, input ports, and most common switching fabrics, impose a constant delay to each cell and do not modify the traffic flow characteristics of a connection. *Variable servers*, on the other hand, add a non-constant delay to each cell, and so modify the traffic characteristics of connections. An output port, for example, acts as a multiplexor and may simultaneously receive cells belonging to different connections competing for transmission on the link associated with the port. Thus, cell blocking may occur, and cells may be forwarded in an order that is determined by the scheduling policy adopted by the switch. An output port, which is a multiplexor, must therefore be considered as a variable server.

Constant servers do not affect the traffic flows. Therefore, they need not be further considered to derive the characterization of traffic flows inside a network. Variable servers, however, modify the traffic that flows through them, and their effect on traffic has to be understood, so that the accuracy of a traffic descriptor does not excessively suffer as the traffic traverses one or more variable servers. In the following section, the network will therefore be modeled as a network of variable servers only.

2.2 Maximum and Minimum Traffic Functions

We define the *output traffic function* $R_{i,X}(t)$ of connection M_i at the (variable) server X to be the amount of data of Connection M_i departing from Server X during the time interval $[0, t)$. Obviously, $R_{i,X}(t)$ precisely describes the traffic of Connection M_i at the output of Server X. The fact that this function is time-dependent makes it an unlikely candidate for a traffic descriptor. We consider two more concise functions, which deterministically bound the expected traffic of a connection, and therefore can be used as envelope to characterize the traffic.

We call Function $F_{i,X}(I)$ the *maximum traffic function* for Connection M_i at Server X if for any $I > 0$,

$$F_{i,X}(I) = max_{s \geq 0} \left(R_{i,X}(s + I) - R_{i,X}(s)\right)$$

That is, the maximum amount of traffic output from Server X for Connection M_i during any time interval of length I is at most $F_{i,X}(I)$.

Similarly, we define $f_{i,X}(I)$ to be the *minimum traffic function*. That is, for any $I > 0$,

$$f_{i,X}(I) = min_{s \geq 0} \left(R_{i,X}(s + I) - R_{i,X}(s)\right)$$

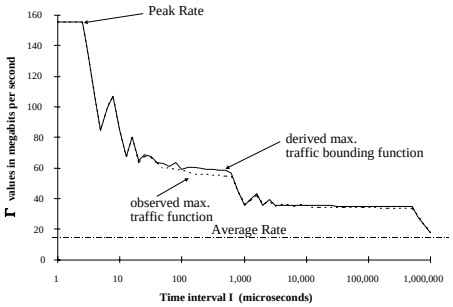

Figure 1: Maximum Rate Function

Again, during any interval of length I, the amount of traffic output from server X for Connection M_i is at least $f_{i,X}(I)$. Figure 1 shows a related measure, the *maximum rate function* $\Gamma(I) = \frac{F_{i,X}(I)}{I}$ of a traffic flow. In this example, a traffic stream is measured by a network analyzer as it enters an ATM network, and the maximum average rate $\Gamma()$is plotted as a function of the averaging interval I.

We use the functions $F()$ and $f()$ as traffic descriptors for the traffic of connections in the networks. $F()$ and $f()$ form the tightest deterministic time-invariant characterization of the traffic at the output of a server. As $F()$ and $f()$ may be defined by a large number of points, they are cumbersome to manipulate and *bounds* on the maximum and minimum traffic functions are used to characterize the traffic.

We define the *maximum traffic bounding function* $B(I)$ to be an upper bound on $F(I)$, that is, $B(I) \geq F(I)$ for all I. Similarly, we define the *minimum traffic bounding function* $b(I)$ to be a lower bound on $f(I)$, that is, $b(I) \leq f(I)$ for all I. Since we base our detection mechanism on $B()$ and $b()$, the more tightly they bound the actual traffic, the more accurate is the resulting classification into compliant and non-compliant traffic.

In the context of real-time communication protocols, maximum traffic bounding functions are used to allocate resources, and tight bounding functions are sought to prevent excessive over-allocation of network resources. In practice, a trade-off must be made between tightness of the bounding function on one side, and the overhead incurred to manipulate it on the other, together with the inherent *a-priori* uncertainty about the traffic characteristics at the sources.

Traffic functions can be easily approximated with piece-

wise linear bounding functions at any level of resolution. Consider a maximum traffic function $F()$. Assume that we know one point of the function $F()$, that is, we know $B' = F(I')$ for some value of I'. We then have a first-order approximation of $F(I)$, which is given by

$$B'(I) >= [I/I'] \cdot B + min(I', I - [I/I'] \cdot I')$$

This can be recursively used to bound the function if more points are known. In this form, coarse bounds (three to five linear segments) on maximum traffic functions can be used for resource allocation purposes, where a broad categorization of traffic streams into classes – for example teleconference, or advertising, or sports – is sufficient. More accurate bounds (say, ten linear segments) can then be used to closely characterize individual traffic streams.

Once the traffic bounding functions are known at the entrance to the network, they can be derived for any point along the path of a connection. This derivation requires to obtain the traffic at the output of a server from the traffic at its input.

Let X and Y be two adjunct servers, and let Connection M_i first traverse Server X and then Server Y. Then,

$$F_{i,Y}(t) = F_{i,X}(t + d)$$

and

$$f_{i,Y}(t) = f_{i,X}(t - d)$$

where d is the worst-case delay experienced by Connection M_i at Server Y. The value for d depends on the scheduling methodology used in the server and on the traffic functions of other connections using that server. For a FCFS service discipline, for example, the worst-case delay on Server X can be bounded as follows, assuming that Server X serves N connections, and the traffic of a connection M_j is bounded at the output of the previous server by the maximum traffic bounding function $B_{j,PREV}()$:

$$d = max_{I \geq 0} \left(\sum_{i=0}^{N} B_{j,PREV}(I) - I \right)$$

Various analytical techniques to derive worst-case delays based on traffic bounding functions for other scheduling policies, such as Static Priority, Generalized Processor Sharing, and various forms of Earliest-Due-Date have been proposed ([4, 5, 7, 8, 12] among many others). The above formula imply that, for Server Y, the upper and lower traffic at its output, modeled by $B_{i,Y}(t)$ and $b_{i,Y}(t)$, can be derived from the traffic at its input (i.e., the output of the previous Server X).

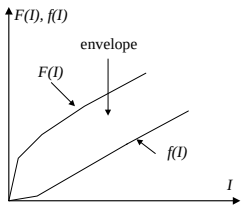

Figure 2: Maximum and Minimum Traffic Function Define an Envelope

Figure 2 illustrates how the maximum and minimum traffic functions define an envelope for the amount of traffic on the connection at the output of a server. The network will, during run time, dynamically examine the traffic and verify if the traffic lies within this envelope. In the case of a violation is found, then a (potential) intrusion is detected. Actions can be taken to immediately suppresses the violation. These functionalities are implemented in a security device, which we discuss next.

3 Design and Analysis of a Security Device

The device must perform the functions of detection, suppression and alert when non-compliant traffic is found to be passing through the network, in a timely manner. *Detection* refers to determining if a cell being transmitted out of a particular port on a switch is in accordance with the maximum and minimum traffic functions defined for the connection, that is, its VPI/VCI pair. *Suppression* involves the discarding of the offending cell and *Alert* refers to a method by which the security device reports the VPI/VCI pair of the offending cell and the switch output which produced it. Optionally, *Alert* also refers to the reporting of the reason for which the cell is found to be in violation, whether it be due to an illegal VPI/VCI pair or due to a violation of the traffic envelope.

The determining factor in the design was the need to implement the device with components that are widely available, inexpensive, and of proven stability. Because of the high data rates involved in the transmission of cells in ATM networks, it was necessary to use as much parallelism of functions as possible in hardware in order to implement the design with standard components and realizable clock speeds.

As illustrated in Figure 3, the device relies on three units functioning in tandem to handle the traffic produced by each ATM network switch output. These three units, labeled *Receiver*, *Analysis Module*, and *Transmitter*, func-

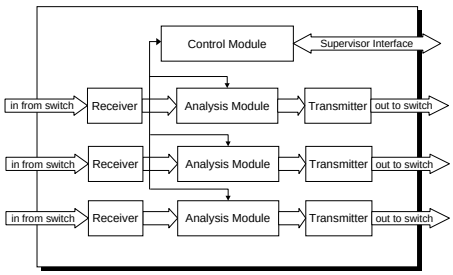

Figure 3: Block Diagram of ATM Switch Security Device

tion in sequence to capture, analyze, and retransmit the network traffic from one ATM network switch output Port: The Receivers queue the incoming data from the ATM network switch and present the data to the Analysis Modules in manageable pieces. The Analysis Modules capture the data from the Receivers and perform the necessary functions of detection, suppression and alert and pass this data to the Transmitters if it is found to be valid. The Transmitters capture the outgoing data from the Analysis Modules and transmit it to the subsequent switch in the ATM network.

Overseeing the operation of the Receivers, Analysis Modules and Transmitters is the Control Module. It is the responsibility of this module to accept data from the Supervisory Interface regarding new connections that need to be admitted in the ATM network and pass this data to the appropriate Analysis Module. Additionally, the Control Module must detect a traffic alert from any one of the Analysis Modules and, when it occurs, must capture the data regarding the cell which caused the alert from the appropriate Analysis Module. Then, the Control Module must transmit this data to the supervisory interface.

The end result is a device that can capture, analyze and retransmit the ATM network traffic on the multiple output ports of an ATM switch, update path information , and report traffic infractions under conditions of peak data rate transmission. The analysis portion of the device's function may be of two types. Under the first variant, arriving network traffic will be checked for validity in terms of whether or not the connection with which that traffic is associated does indeed pass through the network switch and port from which the data originated. The second variant will perform exactly the same verification as the first variant and, in addition, will also verify that traffic that has been found to be traveling across a valid connection has not exceeded the traffic limits placed on that connection.

3.1 Transmitter and Receiver

The receivers and transmitter capture and send the cell data from and to the physical outputs and inputs of the ATM switches between which the device lies and process it according to the particular physical interface characteristics of those switches. This includes any functions of decryption, decompression, and bit-level synchronization. The exact design of these units will be highly dependent on the physical media and beyond the scope of this description. The physical blocks comprising these modules is not a matter of choice since it is already described in the ATM forum literature ([2, 3]) and components for use in these modules are available.

The only design issue that needs to be noted with regard to the function of the receivers and transmitters is that they present data to the Analysis Module in parallel 16-bit words and synchronize the their presentation to the Analysis Module clock. The stipulation that data be presented to and read from the Analysis Modules in 16-bit words arises out of the need to have this device operate at clock speeds that are reasonable for implementation in integrated circuit designs that utilize the major logic families currently available. At the highest speed scenarios of data rates of 622.08 Mbps, it implies that 38.9 million 16-bit words need to be processed by every Analysis Module, which implies a maximum clock rate of 38.9 MHz for the Analysis Modules.

3.2 Analysis Module

The Analysis Module admits a new cell into a 16-bit shift register, word by word from the receiver. In parallel, as components of the VPI/VCI pair belonging to the cell in transit are received from the Receiver (contained in the cell header, consisting of the first five bytes of data) they will also be copied into six four-bit latches.

Once all 24 bits of the VPI/VCI pair associated with the cell in transit have been captured in these four-bit latches, the 24 bits of output from them will be presented to the *memory lookup module* in two 12-bit words, with one word being presented at a time. The control to present these two 12-bit words will be performed by a 12-bit by 4-input multiplexor.

The two words that are presented to the memory lookup module will be interpreted by this module as an address that is uses to perform the actual analysis of the cell's validity. Depending on the version of the Analysis Module

to be implemented, this function will change. Primarily, the memory lookup module will verify if the cell belongs to a connection that does indeed pass through the switch and port from which it originated. Optionally, the module will also verify if the network connection along which the cell in question is traveling has is within the envelope defined by the maximum and minimum traffic bounding function for that connection at that output port.

This result will be used by the sequence/detect module to determine if the cell is valid or not. If the cell is valid, it will enable the output from the last set of latches in the 16-bit shift register to be sent out to the transmitter. If the cell is not valid, the sequence/detect module will suppress output of the cell from the shift register to the transmitter by simply presenting null data (all zero bits) to the input state of the Receiver. In this case, the sequence/detect module will also trigger the interrupt logic in the Control Module. The Control Module will then know that an invalid cell has been detected and will perform the necessary operations to read the VPI/VCI pair of the offending cell from the outputs of the six 4-bit latches, which have been storing this information throughout the entire process.

All the devices used in this circuit are currently feasible in TTL and HC logic families. In addition, a number of tri-state buffers are implicitly used in this design to allow the Control Module to select between the data inputs and outputs of the different Analysis Modules to which it is attached. The interconnection of the functional blocks of the Analysis Module is shown in Figure 4.

The sequence/detect module is a simple sequential state machine with external decode logic. It controls all the inputs and outputs required to perform the functions just described. This state machine is designed using the same type of edge triggered D-type latches and combinatorial logic used to construct the other component blocks of the Analysis Module.

The reasoning behind the design of the Analysis Module was to be able to take advantage of the large number of operations that can be performed in parallel in order to reduce the number of clock cycles necessary for the device to perform its function.

The effect on the performance of the physical communication link passing through this device will be that any cell in transit will be delayed by the amount of time necessary to read in the cell's header and perform the lookup of the VPI/VCI pair contained in these five bytes in the memory lookup module. This means that the controlling factor of the transmission delay a cell will experience in every security device through which it passes will be the sum of these two periods of time, in addition to delays incurred due to link-level synchronization at receivers and transmitters.

The Control Module's logic will be triggered within less than one cell transmit time if the transiting cell is found to be invalid (nine clock cycles, to be precise). This means that the Control Module will know about the violation in less than one cell time and can begin sending data bout the violation to its supervisory control interface within less than one cell time.

3.3 Control Module

The Control Module performs its job asynchronously from the Analysis Module. It handles communication with the supervisory interface and with the hardware in the Analysis Module to which it is attached. To the supervisor interface, the Control Module must report traffic violations detected and read from the Analysis Modules and get information about new valid data paths that have been created in the network. When the Control Module receives data about a new valid path, it must be able to distinguish through which Analysis Module the path passes and must update the valid path information within this module.

A Motorola 68PM302 Integrated Multiprotocol Processor is an ideal candidate because of its current availability at reasonable cost and its capability to provide a broad range of built-in features that closely match the needs of this application. It provides sufficient I/O to be able to perform all the necessary read and write operations to and from the Analysis Module hardware. It offers the interrupt circuitry necessary for the Analysis Module to alert the Control Module of a traffic violation. Finally, it provides a high-speed serial interface, which could be used in conjunction with a DS1 compliant transceiver in order to communicate with the Control Module's supervisory interface.

4 Final Remarks

In this work we propose an intrusion detection mechanism for high-speed networks based on the characterization of the traffic arrival patterns on connections. This approach well complements other traffic characterization schemes. Its strength lies in the fact that the man-

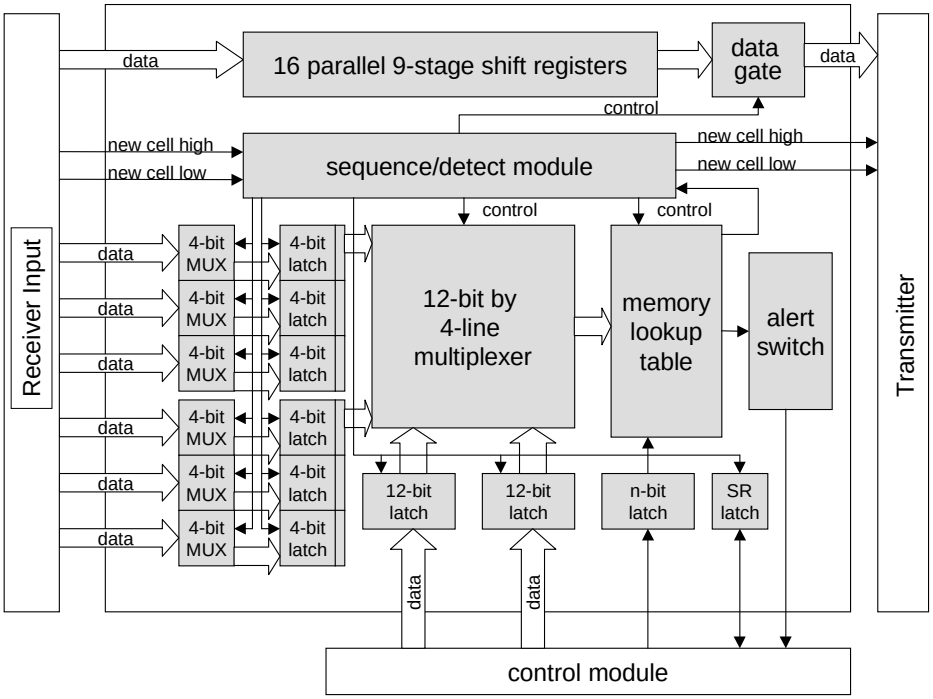

Figure 4: Analysis Module Block Diagram

agement of characterization parameters is greatly simplified. Indeed, we take full advantage of the fact that applications must declare the traffic specification during connection setup if they want their QoS requirements to be met. This traffic specification provides the upper bound of the traffic envelope. An additional minimum traffic specification would provide the lower bound. No sophisticated anomaly detector is needed ([6, 10]) to determine whether an intrusion occurred, since the envelope parameters have been provided during connection setup. For example, if a maximum traffic function is exceeded for a connection, an alert must be triggered to indicate that an intrusion may have happened and that the system is operating at a higher load than what is safe for the given QoS guarantees.

Our traffic model in terms of maximum and minimum traffic functions then gives us a flexible method to formulate the envelope for each connection at an arbitrary point in the network. This allows for a targeted deployment of our proposed ATM security devices across the network, and for an accurate methodology to determine the parameters for the traffic envelopes for connections.

We presented a module level description of a detection device and have shown it to be implementable with currently available off-the-shelf components and custom ASICs available at current levels of integration technology. The performance of the device has been evaluated under worst-case conditions for network traffic. It has been shown that the delay experienced by network traffic in existing virtual connections in the network is trivial when compared to its expected transit time within the network and that the management functions of creating and destroying virtual connections are not a function of the creation/destruction rate of these connections. Through the description of its operation, it is evident that, while utilizing such a framework of traffic security enforcement, the full bandwidth of the network is available to all users for authorized utilization and that traffic delays network cells will experience are constant even under sustained peak traffic conditions.

The details for the components of the device presented here have concentrated primarily on the mechanisms by which actual enforcement should occur and how to limit the impact which it has on overall network performance. Many portions of the larger issues of this method of security enforcement have been glossed over. Foremost among these issues is the topology and physical architecture that should be used to implement the network by which supervisory control data is transferred between the modules that actually provide the enforcement and the workstations that keep the operators of the security body appraised of the state of the network. An integral component of this decision will be an assessment of ex-

actly what criteria to use in order to derive the level of enforcement that the modules designed in this document will be required to perform. Based on this, assessments may be made with regard to what the overall bandwidth and worst-case delays of the overlaying network must be in order to provide an interface to the individual enforcement modules that is deemed to be acceptable from the network management perspective.

References

[1] ABIS Task Force, DoD Advanced Battlespace Information Systems Task Force Report, 1996.

[2] ATM Forum, "DS3 Physical Layer Specification," ATM Forum, January 1996.

[3] ATM Forum, "622.08 Mbps Physical Layer Specification," ATM Forum, January 1996.

[4] R. L. Cruz. "A calculus for network delay," *IEEE Transactions on Information Theory*, 37(1):114–131, Jan. 1991.

[5] R. L. Cruz, "Quality of Service Guarantees in Virtual Circuit Switched Networks," *IEEE Journal of Selected Areas in Communication*, Aug. 1995.

[6] J.S. Javitz, A. Valdes, "The SRI IDES Statistical Anomaly Detector," *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, CA, May 1991.

[7] C. Li, R. Bettati, W. Zhao, "Static Priority Scheduling for ATM Networks." *Proceedings of the Real-Time Systems Symposium*, San Francisco, CA, Dec. 1997.

[8] J. Liebeherr, D.E. Wrege, and D. Ferrari. *"Exact admission control in networks with bounded delay services."*, *IEEE/ACM Transactions on Networking*.

[9] A. Raha, S. Kamat, and W. Zhao. "Admission control for hard real-time connections in ATM LAN's," In *Proceedings of the IEEE Infocom'96*, Mar. 1996.

[10] K. Tan, "The Application of Neural Networks to UNIX Computer Security," *Proceedings of the International Conference on Neural Networks '95*, Perth, Australia, 1995.

[11] Xylan Corporation. *Xylan announces participation in SmartShip program*. Xylan Press Release. URL: http://www.xylan.com/news/pr-053097.html.

[12] H. Zhang and D. Ferrari, "Rate-Controlled Service Disciplines." *Journal of High-Speed Networks,* 3(4):389-412, 1994.