Application of FDDI/XTP Network Protocols to Distributed Simulation

M. Bassiouni
Dept. of Computer Science

J. Thompson
Inst. for Simulation and Training

University of Central Florida
Orlando, FL 32816 USA

Abstract

Recent breakthroughs in communications technology have led to the *emergence of new network protocols with unprecedented high* speeds. Two such protocols are the Fiber Distributed Data Interface (FDDI) and the eXpress Transmission Protocol (XTP). In this paper , we present the results of an ongoing performance *evaluation project to study/assess the* evolution of real-time training networks (e.g., SIMNET) to FDDI and XTP. Both the long-range and short-range design policies for the integration of these high speed protocols into simulation networks are considered. Gatewaying methods specially tailored for FDDI backbones connecting multiple local-area simulation networks are discussed. The implications of the results and the insight gained from our project for improving the networking of real-time simulators are discussed.

Introduction

The advent of direct computer-to-computer communications (computer networks) opened the possibility of interconnecting different types of computer based systems. Distributed simulation is an excellent example which manifests the importance and benefits of the networking technology to real-time systems. *Today, there is major emphasis* being placed on the development of efficient networkable simulation and training systems. In this context, the term *networkable implies* that each simulator is capable of communicating (transmitting and receiving), in real-time, information relative to the simulated exercise
Today, there is major emphasis being placed on the development of efficient networkable simulation and training systems. In this context, the term *networkable* implies that each simulator is capable of communicating (transmitting and receiving), in real-time, information relative to the simulated exercise
which can be understood by other networkable devices tied to the network, thus allowing for fast interaction among the simulation devices.

First generation networks in the field of *simulation and training (such as the* ETHERNET LAN used in the networking of SIMNET devices) have certainly served admirably and *demonstrated the legitimacy* of networking for group training. But the need for a second generation network (such as FDDI) to support distributed simulation *is now rapidly* emerging and is becoming more critical as new improvements are put into use on today's simulation and training devices. In fact, the continuous effort and ongoing intensive research to attain higher and higher levels of performance for networkable simulation systems (e.g., increasing the number of devices tied to the network, decreasing network delays to improve fidelity, incorporating real-time voice transmission, mandating stringent measures to ensure interoperability, etc.) have led many experts *to the realization that the availability* of high speed network protocols will be an absolute necessity for the continuous growth and the successful evolution of *networkable* simulation systems.

In this paper, the application of the FDDI and XTP protocols to distributed simulation is examined. FDDI [6, 7] is a high bandwidth (100 Mbits/sec) general-purpose local-area-network (LAN) that uses fiber optics as its transmission medium. Its architecture is primarily a token-passing, dual (counter rotating) ring configuration which is capable of connecting 1000 physical connections on a total fiber path of 200 kilometers. FDDI consists of the following entities:

1) A Physical Layer (PHY) [1] which fulfills the medium, optical bypassing, connectors, and drivers/receivers requirements.

2) A Data Link Layer (DLL) that is divided into two sublayers: i) A Media Access Control (MAC) [2] which implements the token-based algorithm for access to the transmission medium, and ii) A Logical Link Control (LLC) which provides the required interfacing to the Network layer running on top of FDDI.

3) A Station Management (SMT) function [3] which provides the control services required for station initialization, fault isolation and recovery, and scheduling procedures.

XTP [9] is a lightweight transport protocol designed to meet the requirements of real-time applications. The protocol is implemented in VLSI using a small number of chips, and has a sustained transfer rate of 100 Mbits/sec (scaleable to 1 Gbits/sec). Some of the functions provided by XTP include: stream services (connect, disconnect, send, receive), real-time datagram services, flow and error control, delivery confirmation, and reliable multicasting. XTP contains functions corresponding to OSI layers 3 and 4, and is expected to become one of the best choices for the implementation of layers 3 and 4 in FDDI-based real-time networks.

The Institute for Simulation and Training (IST) at the University of Central Florida has established a Network and Communications Laboratory dedicated to performing research for the purpose of enhancing the networking capabilities of real-time networked simulation systems. This laboratory houses a number of real-time networkable simulators and is the center of several research activities dealing with different aspects of real-time networked simulation. In this paper, we report on the findings of an ongoing project focusing on the application of high speed network protocols (e.g., FDDI and XTP) to distributed simulation. The implications of the results of our study and the insight gained from our research for improving real-time simulation networking are discussed.

The Network System Configuration

The architecture of FDDI specifies a dual (counter rotating) token ring configuration that provides point-to-point connections between every pair of adjacent nodes on the ring. In normal operation, data frames circulate between the nodes using a single ring. The purpose of the dual ring is to gracefully recover from failures due to breaks in the transmission medium. Such breaks would render a single ring useless, but can be easily handled in a dual counter-rotating ring. This is done by simply reconfiguring the network using the redundant ring to isolate the fault. In this paper, we concentrate on evaluating the performance of FDDI during normal operation; the aspect of error recovery will not be discussed.

Figure 1 shows a typical FDDI configuration that might be used to connect real-time simulation devices. Stations (i.e., simulators) are shown as nodes and connected in the form of a loop. Normally each simulator will be directly attached to the ring, but FDDI also allows the use of concentrators which can connect a number of stations to the ring. In simulation networks, these concentrators might be used to connect either a group of nodes which belong to the management and control of the training exercise, or to connect a group of regular nodes that are physically located near each other.

Access to the physical medium is controlled by passing a token around the FDDI ring. The token gives the station which receives it the opportunity to transmit a frame of data (or a sequence of frames) onto the ring. If a station wants to transmit data, it first strips the token from the ring (by not repeating it to the downstream station). After transmission of its eligible queued frames, the station issues a new token and transmits it onto the ring. Stations that are not transmitting merely repeat the incoming frame stream. A circulating frame is removed from the ring when it is received by the station that originated it, i.e., a station does not repeat its own frames.

The configuration shown in Figure 1 represents a second generation solution for simulation networks since it requires that all simulators be equipped with FDDI interfacing and FDDI ring attachments. This means that existing networkable simulators (normally equipped with ETHERNET interfaces) would

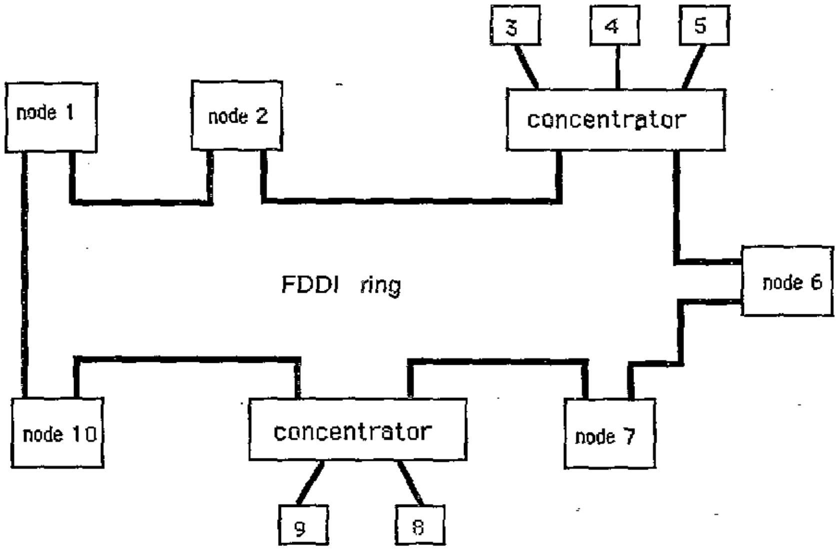

Fig. 1 An example of FDDI topology

have to undergo both software and hardware modifications in order to adapt to the FDDI technology. Fortunately, a short-term solution is possible and is quite suitable for the application of FDDI technology to existing simulation devices. Due to its high speed and its strict adherence to the OSI standards, FDDI is expected to be remarkably successful when used as a backbone [8] to connect standard (slower) LANs (e.g., the 10 Mbits/sec ETHERNET [4], the 4 Mbits/sec and the 16 Mbits/sec token ring [5]). Figure 2 shows an FDDI backbone configuration that connects two ETHERNET (IEEE 802.3) and two token-ring (IEEE 802.5) standard networks via special gateways. Existing simulators can be connected to the standard LANs, while newly developed (FDDI-based) simulators can be connected to the FDDI ring either directly or through concentrators. The design of the special gateways shown in Figure 2 will be briefly discussed at the end of this paper.

FDDI Access Protocol

FDDI uses a *timed token rotation protocol* that dynamically adjusts the amount of data which can be transmitted by a node during each token rotation. The protocol supports two major classes of services:

i) **Synchronous:** this class of service is used for the transmission of information *requiring real-time processing* (e.g., real-time voice transmission). The synchronous bandwidth for each station is pre-allocated (via the SMT) during the initialization of the ring.

ii) **Asynchronous:** this class of service is used for applications whose response time requirements are less critical (e.g., interactive applications with bursty traffic). Asynchronous bandwidth is instantaneously allocated from the unused or non-allocated portion of LAN bandwidth.

When a station captures the token, it transmits the pending synchronous frames (if any) without exceeding its pre-allocated synchronous bandwidth. A station may also transmit asynchronous frames if the response time requirements of synchronous data during the last token rotation have been satisfied (as will be shortly explained).

FDDI Timers and Counters

The implementation of the FDDI media access protocol is accomplished via a set of timers and counters that are used to control the transmission of data frames. The

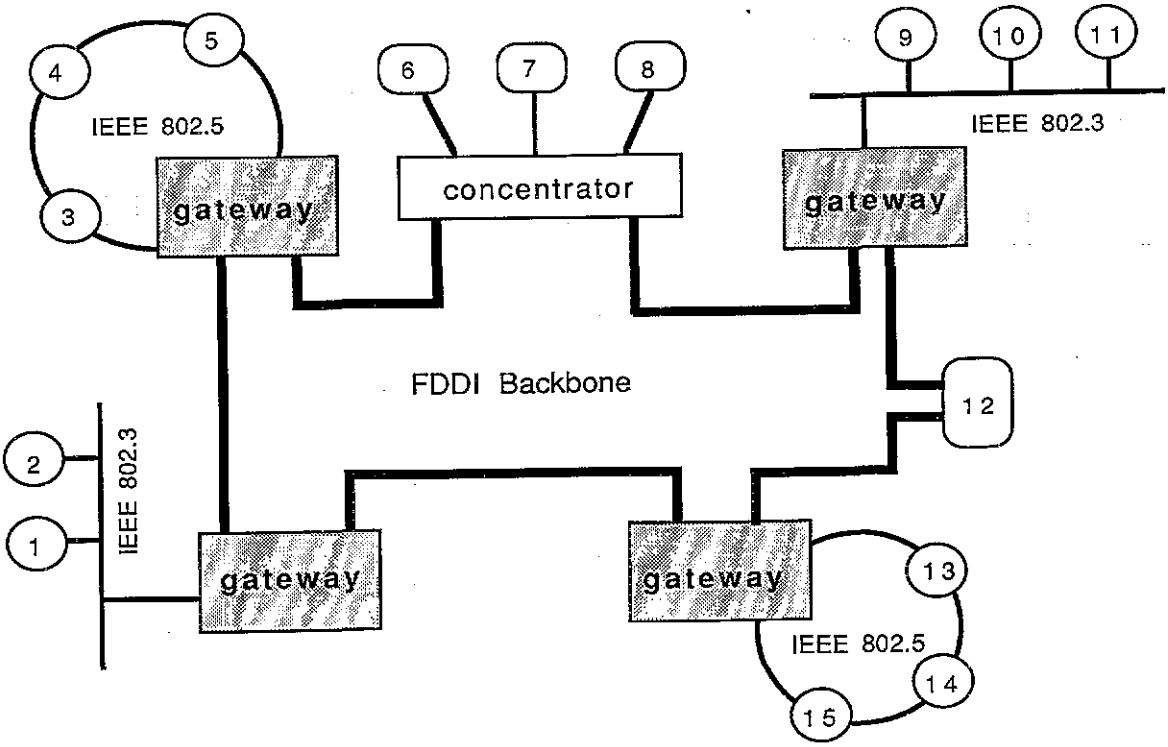

Fig. 2 An FDDI Backbone

following are the important timers/counters maintained (locally) by each node on the network.

- Token-Rotation Timer (TRT): this timer is used to control the station's access to the network. TRT is initialized to TTRT (the target token rotation time) negotiated during the initialization of the ring. The value of TRT is reset to TTRT whenever the token arrives to the station or when TRT expires (after TTRT units of time). In the latter case, the counter Late_Ct is incremented by one.

- Late_Ct: is a counter used to indicate the number of times the timer TRT has expired since the token was last received by this station. Late_CT is initialized to zero and is incremented each time TRT expires. When the station receives the token, Late_Ct is reset to zero and TRT is reset to TTRT.

- Token-Holding Timer (THT): this timer is used to control the amount of asynchronous data that can be transmitted. When the station receives the token, the value of THT is initialized to the current value of TRT and TRT is then reset to TTRT. Asynchronous data may be transmitted by the station if the value of THT is less than the priority threshold value of the pending asynchronous frame(s).

Network Operation

A token arriving at a given station is considered to be an *early token* if Late_Ct = 0 and the value of TRT at this station is less than the (global) target token rotation time TTRT. Otherwise, the token is considered to be a *late token*. Thus, if Late_Ct is not zero when a station receives the token, the station concludes that the token has failed to circulate (in its last rotation) within the time allotted by the constraints imposed by synchronous traffic.

The high level FDDI access protocol can be simply described as follows. A token arriving before TRT reaches TTRT (i.e., an early token) can be used for both synchronous and asynchronous transmissions. A token arriving after TRT reaches TTRT (i.e., a late token) can be used only for synchronous transmission. The status of the token is dynamically adjusted in each station based on the token's most recent rotation.

Asynchronous bandwidth is further controlled by a two-tier allocation mechanism, enforced by two types of tokens: *restricted* and *nonrestricted.* In the *restricted* mode, the token is used to establish a single extended dialog between specific nodes on the LAN (depriving other nodes from access to the asynchronous bandwidth). The *nonrestricted* mode, on the other hand, is the normal mode of FDDI operation and is based on giving all nodes equal access to the asynchronous bandwidth. Transmission of synchronous data proceeds normally regardless of the token type. Accordingly, synchronous frames are always given priority over both restricted and nonrestricted asynchronous frames. Because of the broadcast nature of the majority of messages communicated within simulation networks, only the nonrestricted mode of FDDI operation is considered in this paper.

Multiple levels of asynchronous priority may optionally be distinguished by a station. For each implemented priority level **n**, a threshold value T_Pri(**n**) is established. A nonrestricted token can only be captured by a station for the transmission of an asynchronous frame of priority n if the value of the THT timer is less than the associated threshold T_Pri(**n**). By setting lower threshold values for lower priority levels, transmission of lower priority frames is deferred when the ring is more heavily loaded. For example, if T_Pri(**n**) is set to a very small value, then transmission for priority **n** is only possible if the token arrives early enough such that TRT and hence THT are less than T_Pri(**n**). If the priority feature is not supported in a given station, the MAC protocol in that station assumes that all asynchronous frames have an equal effective priority threshold of TTRT.

The Simulation Model

Detailed simulation programs (written in Ada) are being developed and used at IST to study the suitability and impact of implementing real-time simulation networks using newly-emerging network protocols such as FDDI and XTP. In this section, we give a high-level description of the simulation model used in evaluating and predicting the performance of FFDI-based simulation networks.

Ada, with its flexible synchronization and concurrency features, has been employed in our simulation of the FDDI/XTP network. The process interaction model maps the different entities and activities of the real-time network to its corresponding Ada tasks. The following is a brief description of the major generic Ada task types used in the FDDI simulation.

- Task FDDI_NODE: this task is used to represent the interface of FDDI with higher network layers. A task of this type is created for each node on the simulated network. The FDDI_NODE task can be executed in one of two modes: isolated or integrated. The isolated mode is used to evaluate the performance of FDDI separately (i.e., independent of higher network layers). In this mode, the FDDI_NODE task acts as the source of local traffic. It generates frames according to a specified input method such as using traces of real-data or stochastically generated interarrival times (e.g., exponential, uniform, fixed with jitters, etc.). The integrated mode is used to simulate FDDI in conjunction with higher network layers (e.g., XTP or TCP/IP). In this case, the FDDI_NODE task receives traffic through its interaction with the load-generating entity of the simulated higher network layer.

- Task FDDI_MAC: this task is used to implement the FDDI media access control. A task of this type is created for each node on the simulated LAN. The FDDI_MAC task keeps track of the different timers and counters used in the implementation of the FDDI MAC protocol.

- Task FDDI_PHY: this task is used to implement the physical layer of the FDDI protocol. It is responsible for receiving data from the upstream station, transmitting data to the downstream station, and applying the FDDI encoding functions.

- Task SERVER: this task is used to simulate the propagation of traffic and the corresponding delay between successive pairs of FDDI nodes. A task of this type is created for each pair of adjacent nodes on the FDDI ring.

- Task SCHED: this single task is used to time order events and control the sequencing of the entire simulation. The SCHED task maintains a simulated clock and uses it to determine the time when each waiting task should be reactivated. The simulated clock is advanced only when all tasks are in a waiting state; thus any computation done by a task takes place in zero simulated time.

When the FDDI_MAC task receives a token (via its FDDI_PHY task), it checks to see if there are any eligible queued frames (delivered by the FDDI_NODE task). If such frames exist, the station strips the token from the ring and starts the transmission of synchronous frames. The FDDI_MAC also examines its appropriate variables and, in the case of an early token, transmits the eligible asynchronous frames according to the priority mechanism explained earlier. All frames are transmitted through the FDDI_PHY task to the SERVER task which delivers them to the downstream station after simulating the appropriate propagational delay.

Performance Evaluation

The FDDI scheme uses an underlying token passing mechanism with appropriate modifications to handle synchronous data. If the FDDI parameters (thresholds and initial timer values) are carefully chosen, the FDDI LAN operates smoothly without (or with very few) reinitializations. In this case, the performance of FDDI is basically that of a high speed token ring. Unlike ETHERNET whose performance deteriorates at high traffic loads due to excessive packet collisions, the overhead of FDDI token management does not result in throughput decline when the traffic load on the ring increases. Figure 3 shows a typical relationship between the throughput and traffic load in an FDDI ring. Since FDDI uses a collision-free protocol, it does not suffer from the problem of declining performance at high loads. Throughput around 90% of the transmission medium bandwidth can be easily obtained in FDDI LAN's (compare this to the ETHERNET protocol whose throughput is usually limited to about 65% of the medium bandwidth).

Packet delays in FDDI LANs depend to a large extent on the frequency of interruption of normal network operation due to network reinitializations. When the lateness of the token is judged to be serious, a ring reinitialization procedure is invoked by the SMT function in one of the nodes. The purpose of this procedure is to adjust the ring parameters in order restore stability and allow faster token rotation time. During reinitialization, a new value for TTRT is established by means of a bidding process. The ring then resumes normal operation and the transmission of data frames proceeds using the new parameters. If the ring parameters are carefully selected, it would be possible to operate the ring without significant overhead due to reinitialization. In this case, the delay characteristics are similar to those of a token ring. Figure 4 shows a typical relationship between traffic load and the average time required for a packet to be successfully communicated through the network.

A major advantage FDDI offers the real-time user is that it guarantees upper bounds on the time between successive arrivals of the token at a station. Specifically, the FDDI

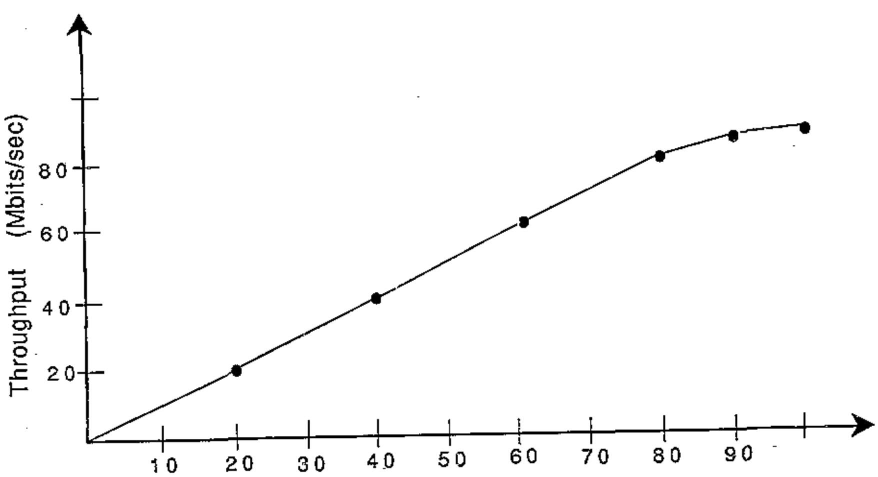

Figure 3

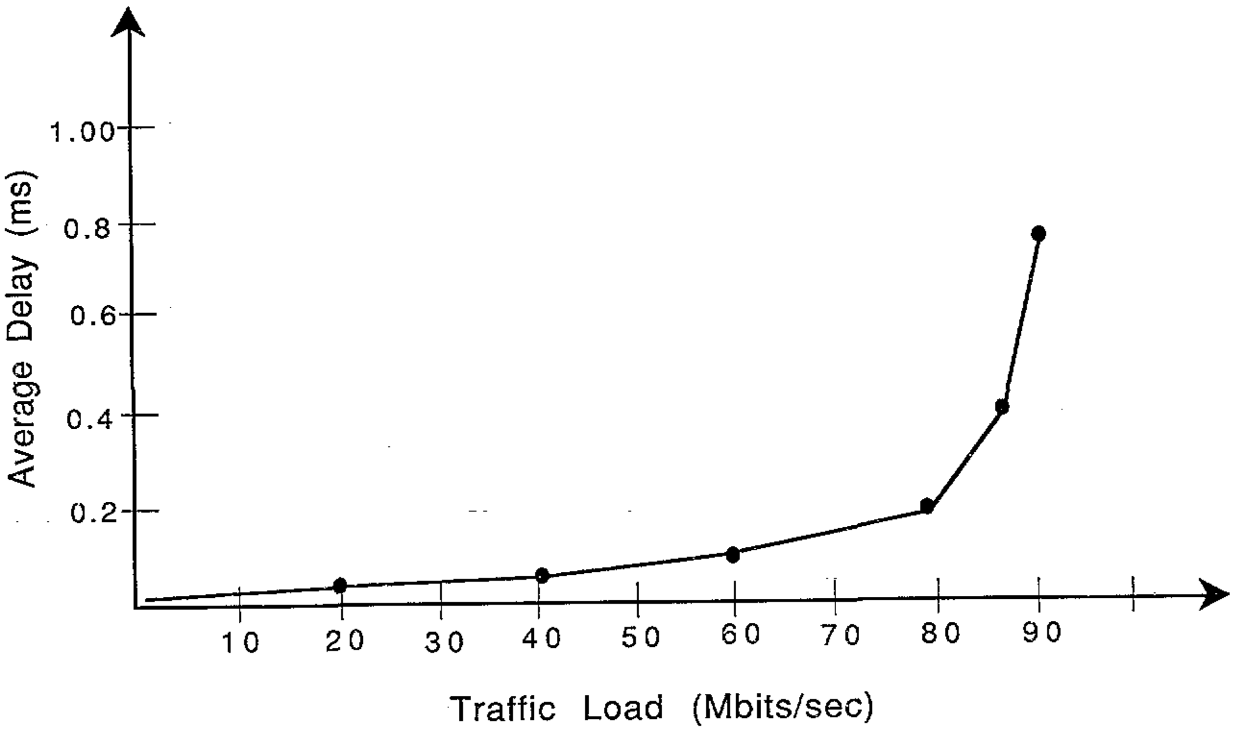

Figure 4

MAC protocol guarantees an average TRT (also called average synchronous response time) of not greater than TTRT and a maximum TRT (also called maximum synchronous response time) of not greater than twice TTRT. The following table gives the upper bounds on the average and maximum synchronous response times, assuming that the minimal bidding TTRT is proportional to the number of stations on the ring and is equal to 4 milliseconds when 500 stations are connected.

No. of Nodes	Avg. Synch. Response Time (milliseconds)	Max. Synch. Response Time (milliseconds)
60	0.48	0.96
70	0.56	1.12
80	0.64	1.28
100	0.80	2.00
280	2.28	4.56
450	3.60	7.20

Conclusions

Application of newly emerging high speed network protocols (such as FDDI and XTP) to distributed simulation systems will have a great impact on improving the overall effectiveness of these systems. In the long-term, FDDI (and perhaps XTP) might be fully incorporated into the simulators of future training networks. In addition to improving the real-time performance of these networks and increasing their capacity to support larger number of simulators, FDDI and XTP might also permit the addition of critical interoperability features that are currently absent in existing simulation systems. Many of the design features of FDDI and XTP (e.g., synchronous transmission, delivery confirmation, asynchronous multiple priority, etc.) can be utilized in the design of state-of-the-art networkable simulation systems in order to enhance their capabilities for providing such features as real-time digital voice/video services over the network.

In the short term, FDDI backbones appear to be ideal for connecting existing networkable simulation systems (e.g., ETHERNET based simulation networks) and would therefore provide an effective tool for increasing the capabilities of these systems without introducing any changes to the individual simulator units. Such a solution, however, requires the employment of intelligent gatewaying and filtering algorithms in order to reduce traffic into each of the slower local networks.

Two possible filtering schemes that are

appropriate to FDDI backbones are:

- *Filtering at Reception* - This scheme requires all frames to be transmitted onto the FDDI backbone and are later filtered by each receiving gateway in order to select those frames that are pertinent to the nodes served by this gateway.
- *Filtering at Transmission* - This scheme requires that frames originated by a node at a local network are filtered at the local gateway and are sent to other gateways only if they carry information relevant to other (non-local) nodes.

Filtering at reception has the advantage of simplicity and ease of implementation while filtering at transmission can potentially reduce the traffic load on the FDDI backbone. Further research is needed to evaluate the performance of these two schemes as well as other hybrid methods that use a combination of the two schemes.

Acknowledgements

This work has been supported by the Defense Advanced Projects Research Agency (DARPA) and the Army's Program Manager for Training Devices (PM TRADE) under Broad Agency Announcement #88-01, Contract N61339-89-C-0043. The views and conclusions herein are those of the authors and do not represent the official policies of the funding agency, the Institute for Simulation and Training, or the University of Central Florida.

References

[1] ANSI Standard X3.148 "FDDI token-ring physical layer protocol (PHY)" American National Standards, 1988.

[2] ANSI Standard X3.139 "FDDI token-ring media access control (MAC)" American National Standard, 1987.

[3] ANSI Standard X3.100 "FDDI token-ring station management (SMT)" American National Standard, 1988.

[4] IEEE/ANSI Standard 8802/3 "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specification" IEEE Computer Society Press, 1985.

[5] IEEE/ANSI Standard 8802/5 "Token-ring access method" IEEE Computer Society Press, 1985.

[6] Kolnik, I. and Garodnick, J. "First FDDI local area network" Proc. 12th IEEE Conference on Local Computer Networks, 1987, pp. 7-11.

[7] Sevcik, K. and Johnson, M. "Cycle time properties of the FDDI token-ring protocol" IEEE Transactions on Software Engineering, Vol. SE-13, No. 3, 1987, pp. 376-385.

[8] Marrin, K. "Emerging Standards, hardware, and software light the way to FDDI" Computer Design, Vol. 28, No. 7, 1989, pp. 51-57.

[9] Chesson, G. "XTP/PE Overview" Proc. 13th IEEE Conference on Local Area Networks, 1988, pp. 292-296.

About the Authors

M. A. Bassiouni received his Ph.D. degree in Computer Science from Pennsylvania State University in 1982. He is currently an Associate Professor of Computer Science at the University of Central Florida, Orlando. His current research interests include computer networks, distributed systems, databases, and performance evaluation. He has authored several papers and has been actively involved in research on local area networks, concurrency control, data encoding, I/O measurements and modeling, file allocation, and relational user interfaces. Dr. Bassiouni is a member of IEEE and the IEEE Computer and Communications Societies, the Association for Computing Machinery, and the American Society for Information Science.

J. Thompson received his BS degree in Electrical Engineering from the University of Central Florida in 1978. He is currently a Research Associate for the Institute for Simulation and Training (IST) at the University of Central Florida, Orlando. Mr. Thompson has technical responsibility for all IST research activities involving computer and simulation networking.