[10] International Standards Organization. Basic Reference Model for Open System Interconnection. *ISO 7498*, 1984.

[11] William Stallings. *Data and Computer Communications*. MacMillan, 1988.

[12] Sun Microsystems, Inc. *LANCE 10Mbit/sec Ethernet interface*, June 1992. From the online manual page le(4).

[13] Andrew S. Tanenbaum. *Computer Networks*. Prentice-Hall, 1989.

[14] Fourad A. Tobagi. Multimedia: The Challenge Behind the Vision. *Data Communications*, pages 61–66, January 1993.

that is prevalent in much of the commercial and non-commercial organizations. By providing better methods to transfer multimedia data over an Ethernet, multimedia use is encouraged which leads to better, more productive, and friendlier interfaces in the workplace and in the academic environments.

Besides multimedia applications, these protocols can be used in applications (e.g, control systems, industrial process control, automated manufacturing, air-traffic control, program trading, etc.) where certain tasks must meet deadlines. Critical tasks can be assigned higher priority to ensure better service to them. In addition, these protocols can be used for priority scheduling in a general computing system to give priority to a user because of projects deadlines, importance of the user, or because a user pays higher fees for better service, etc.

Last but not the least, this research will result in the identification of the functions of new (intelligent) network interfaces that will perform most of the priority transmission related chores in hardware. Simulation study in this research will also reveal the performance trade-offs between having all these functions at the lowest layer and having some functions at higher layers.

References

[1] Dave Andrews. High Stakes for High-Speed Ethernet. *Byte*, pages 22 – 23, October 1993.

[2] Douglas Comer. *Internetworking With TCP/IP: Principles, Protocols and Architecture.* Prentice-Hall, 1988.

[3] E. Mafla, Bharat Bhargava. Communications Facilities for Distributed Transaction-Processing Systems. *IEEE Computer*, pages 61 – 66, August 1991.

[4] Domenico Ferrari. Client Requirements for Real-Time Communication Services. *IEEE Communications Magazine*, pages 65 – 72, November 1990.

[5] Khaled Amer, Ken Christiansen, Tora Toher. Experiments With Client/Server Multimedia On Token Ring. In *Proceedings of the 18th Conference on Local Area Networks*, 1993.

[6] Khaled Amer, Ken Christiansen, Tora Toher. Multimedia Applications on IBM 16-bit Token-Ring LANs. In *IBM Personal Systems Journal*, April 1993.

[7] R. Metcalfe and D. Boggs. Ethernet: Distributed Packet Switching for Local Computer Networks. *Communications of the ACM*, 19(7):395 – 404, July 1976.

[8] A.D. Narasimhalu and S. Christodoulakis. Multimedia Information Systems: The Unfolding Reality (Guest Editors' Introduction). *Computer*, 24(10):6 – 8, October 1991.

[9] The Institute of Electrical and Electronics Engineers. Carrier Sense Multiple Access with Collision Detections (CSMA/CD) Access Method and Physical Layer Specifications. *American National Standard ANSI/IEEE Std. 802.3-1985.*

respect to placing the support for the priority protocol at each level, there is an issue of technological feasibility.

For example, if want the network interface, i.e., the Ethernet hardware controller, to be able to implement a priority protocol completely at its own, we are expecting the network interface to be substantially "intelligent" and to be able to perform many additional chores much more quickly. Clearly, currently available network interfaces do not support those functions. Placement of functions at the lowest level, their feasibility (cost effectiveness), and trade-offs will be studied. Evidently, there is a very distinct alternative: implement priority protocols in the device driver, controlled by the node's operating system. (Note that the device driver is responsible for writing and reading data to and from the hardware interface buffer.) There is a very distinct trade-off here. An intelligent interface requires new, sophisticated hardware, and is likely to be much more expensive than the existing Ethernet controllers, while implementing priority protocols in the device driver calls for software modifications to the device driver only. However, if priority is implemented in the device driver, the device driver must interpret all the broadcast priority information packets, interrupting the CPU to process each of these packets. Clearly, the load on the CPU will be much greater in this case than if all the information were handled at the hardware interface level. Device drivers are interrupt driven and a context switch is required every time a packet is received and additional processing is required to pass information to and from the different protocol layers. These research issues and trade-offs will be explored in this research.

6 Conclusions

Traditional protocols for local area networks, e.g., an Ethernet, were designed to handle data traffics which are bursty in nature and for which variable delay is acceptable. These protocols are not suitable for multimedia traffic which requires pretty much a constant delay in packet deliveries as well as fast (or real time) delivery of packets.

The protocols developed in this research will enable priority transmission of data on Ethernets. By giving multimedia data priority over regular data, the developed priority protocols can be used to deliver multimedia data over the Ethernet in a timely manner with low variance in delay. These protocols will allow a higher throughput for multimedia data with fewer packets dropped (due to missed deadlines). Since multimedia is becoming important in a large number of business, government, and academic organizations and since an Ethernet is a widely used local area network among most of these organizations, priority protocols for Ethernets will enable these organizations to integrate multimedia capabilities into their existing systems with little change. Other alternatives available to these organizations to integrate multimedia capabilities into their systems will cause an expensive hardware overhaul.

Thus, this research will allow a greater access to multimedia utilizing existing technology

allow priority transmission of data on Ethernets. Assign multimedia data priority over regular data and use the developed protocols for priority transmission to allow multimedia data to be delivered in a timely fashion over the Ethernet. We have outlined some preliminary approaches to develop priority transmission protocols for Ethernets. We will completely develop these protocols and will also investigate further approaches to achieve this objective. We will develop necessary data structures for keeping state information, and will study various alternative placements of the data structures and functions to various layers. As we argued before, the concept of priority must be supported (i.e., actual transmission of packets in priority order must be done) at lower layers, i.e., level 2, if we were to have a global, network wide priority. Still other chores such as collecting and interpreting priority information, and updating state information data structures can be implementing at higher layers. We will study the suitability of various function assignments to layers and the trade-offs involved.

Clearly, keeping exact information about node priorities will be expensive (in terms of message overhead and CPU overhead at each node). Therefore, methods to keep approximate priority information will also be investigated. These methods will lazily disseminate priority information about nodes. Consequently, nodes will make incorrect decisions occasionally, but the overhead on the Ethernet and CPUs will be reduced. Obviously, these methods have trade-off and their cost-performance ratio will be studied using simulation.

2. It is quite clear from the discussion so far that the design of priority protocols for Ethernets is full of trade-offs and various alternative design choices are present at different design steps. We will study the performance of promising design alternatives using discrete-event simulations. (The promising designs will be identified using bottleneck analysis or common sense reasoning.) Simulation models will be developed for the designed priority protocols for Ethernets. An extensive simulation study for various mix of workloads and various design alternatives will be performed evaluate the performance of the developed protocols and their suitability to multimedia traffic.

   However, the priority communication should not completely choke off the regular data traffic. It should be able to provide a better service to multimedia traffic without having a serious adverse effect on regular data traffic. This is a capacity planning issue and requires studying of the maximum number of concurrent multimedia streams that can be active at any time without degrading the quality of service to regular data traffic. We plan to address this issue in the simulation study.

3. We will investigate protocol structuring and functional placement issues. This will not only involve studying the performance trade-offs but also examining technological demands while placing functions in various layers in the layered architecture. Recall that while the basic mechanisms of a priority protocol must be supported at the lowest protocol level, other parts of it (e.g., maintenance and update of priority information about nodes) can be located at higher levels. Besides costs and benefits issues with

Third Approach

The third approach is to allow asynchronous broadcast of node state information. No distinct round of state information exchange occurs. Instead, the broadcast of state information is asynchronously embedded in the normal data transmission. That is, the priority state data compete with the actual data for network bandwidth using the CSMA/CD protocol to resolve conflicts. In this approach, when a node gets a data packet for transmission from a user level process, it immediately broadcasts the priority of the packet if that packet has a priority greater than anything pending transmission at that node, i.e., if the node now has a higher priority packet to send. The type and amount of traffic and utilization of the Ethernet will influence the effectiveness of this method.

Recall that in order for the second highest priority node to begin transmitting, it must be informed when the highest priority node has finished transmitting its packet. We have identified two approaches to disseminating the information that a node has finished transmitting its data. The first approach is to have every node to monitor the Ethernet and update their state vector every time a node successfully transmits data. For current Ethernet interfaces, this requires the hardware to be set in "promiscuous mode" which may put a heavy load on the CPU because it must do context switches and pass the packets up and down the protocol layers. However, if the interface were intelligent and could update the state vector without the CPU having to process all the packets, this can be a viable approach. Since current interfaces do not support such features, this approach can only be tested through simulation.

In the second approach a node, after successfully transmitting data, transmits a broadcast packet indicating that it finished transmitting data. New priority information about the node can be included in this packet. For the first approach to disseminating the state data with synchronous state information exchange phases (with N = 1), the phase would be initiated by the node that finished transmitting data, and there would be no additional overhead. For the other approaches, there is an overhead of a broadcast packet that is transmitted after every data message. The more packets per data message, the smaller the overhead of the broadcast packets. However, in the worst case when each data message occupies one physical Ethernet packet or frame, the number of messages transmitted on the Ethernet are doubled.

5 Future Research

Future research will involve the following three issues: (i) developing solutions to the problems associated with transfer of multimedia data on an Ethernet, (ii) study of the performance of these solutions and their suitability to multimedia traffic, and (iii) investigating the performance trade-offs and technological demands in placing functions in various layers in the layered architecture (i.e., protocol structuring and placement issues). Below we elaborate on each of these.

1. To develop solutions to the problems associated with transfer of multimedia data on an Ethernet, we will follow the following approach: Develop a set of protocols that

4 Approach Taken

Once the priority information of packets at all nodes is disseminated, the rules for transmitting packets on Ethernets are straightforward, i.e., a node waits until it has the highest priority. However, dissemination of the priority information of nodes to all the nodes efficiently is a complex issue. Dissemination of priority information should take precedence over the transmission of data on an Ethernet. This is difficult to implement on the Ethernet that does not support priority communication to start with. There are several factors involved that may influence the performance, including the number of nodes, the amount of network traffic, and the distribution of the traffic among various priorities involved.

There are two issues involved in disseminating the priority information. The first is transmitting the priority of the highest message for every node on an Ethernet. The second is disseminating the information that a node is finished transmitting its data packet or packets so that the node with the next highest priority can start transmission.

We have identified the following preliminary approaches for the disseminating node priority information for the priority protocol.

First Approach

The first approach to distributing the state information is a synchronous method. After transmission of each data packet (or after N packets), there is a round of priority information exchange. Each node sends a broadcast packet informing other nodes of the highest priority packet it wishes to transmit. Nodes broadcast their status information in an *a priori* fixed order. For example, node i broadcasts its information after it receives node $i-1$'s state information, after which node $i+1$ broadcasts its state. Provisions will be made to handle adding and dropping nodes. This method should virtually eliminate Ethernet contention during the priority information exchange phase, however, it may suffer from efficiency problems, some channel bandwidth is taken up by exchange of priority information and processing overhead of the broadcast packets up and down the protocol layers.

Second Approach

The second approach is a variant of the first. The state information exchange phase occurs after a fixed number of data messages are transmitted. However rather than having a synchronous exchange of the state information, all nodes on the Ethernet transmit their state messages in the state information exchange phase, using the existing CSMA/CD with exponential backoff to handle collisions.

There is a potential delay in the first approach due to gaps in the transmission of the priority information packets. The gaps are caused by the time it takes the device driver of node's operating system to process the priority information packets before the node can transmit its priority state information. The second approach alleviates the gaps by allowing the network interface to transmit the priority information packets as soon as the Ethernet is free during the state information exchange phase. However, delays can be incurred due to packet collisions and retransmissions during the state information exchange phase. The number of nodes on the Ethernet, the data traffic patterns and the latency from the device drivers in the first method will affect the relative performance of the first two methods.

Ethernet. In order to decide which node has the highest priority packet, each node must have knowledge of the highest priority packet of the other nodes on the Ethernet. Therefore, timely dissemination of this information is an important issue. Each node maintains a vector containing the priority of the highest priority packet every other node wishes to transmit.

(b) *Transmission of packets in priority order.* When a node wants to transmit a packet, in addition to the existing CSMA/CD checks, it compares the priority of its packet with that of the other nodes on the Ethernet. This is done by examining the vector containing the priorities. If it has the highest priority, it is allowed to transmit the packet immediately. Ties are broken by an *a priori* ranking of the nodes.

2. *Structuring issues:* These issues deal with assigning protocols functions to various layers (determining what layers are responsible for what). Priority cannot be implemented at the Network Layer (level 3) or higher layers because implementation of priority at these layers will guarantee enforcement of priority only locally. It will not guarantee transmission of packets in the order of priority at global level (among all nodes). Also, in this case the processing cost and time required to pass information up and down the protocol layers is likely to be high. Therefore, the concept of priority needs to be implemented at the lowest level possible, specifically, the Data Link Layer, which corresponds to the Ethernet device drivers.

   Ideally, the network interface, the Ethernet hardware controller, should be "intelligent" and be able to implement a priority protocol on its own (a hardware-oriented solution). However, if that is not possible, priority protocol can be implemented in the device driver controlled by the node's operating system (a software solution). An intelligent interface requires new, sophisticated hardware, and will be more expensive than the existing Ethernet controllers, while a device driver change only requires a software modification. There is a trade-off, however. If priority is implemented in the device driver, the device driver must then interpret all the broadcast priority information packets. This requires interrupting the CPU to process each of these packets. Clearly, the load on the CPU will be much greater in this case than if all the information were handled at the hardware interface level. Device drivers are responsible for writing and reading data to and from the hardware interface buffer. Since the device drivers are interrupt driven, the amount of work they do must be minimized. A context switch is required every time a packet is received and additional time is required to pass information to and from the different protocol layers.

   While the basic mechanisms of a priority protocol must be supported at the lowest protocol levels, other parts of it can be located at higher levels. There are costs and benefits with respect to placing the support for the priority protocol at each level and it is a research issue that will be explored in this research.

3 Priotiry Protocol for Ethernet

3.1 Objectives

Existing Ethernets have two main problems with regard to the integration of multimedia streams. First, the existing bandwidth capability for a standard Ethernets (10Mbps) is not enough to support (future) multimedia transmission requirements. Second, there is no priority mechanism for transmitting data on Ethernets. Therefore, if an Ethernet is used for multimedia, transmission of multimedia data must compete equally with the transmission of regular data. Note, however, that multimedia streams must be received at the presentation node in time such that there is no interruption (or gap) in presentation from user's point of view (consider a moving picture presentation here). That is, the Ethernet must be able to deliver multimedia packets with a small and constant delay (in real time). Clearly, the communication protocols of the Ethernet that are designed to efficiently transfer regular data traffic which are bursty in nature and can tolerate variable delay, must be modified to suit multimedia applications.

The goal of the proposed research is to develop a set of protocols that enable priority transmission of data on an Ethernet. That is, if a set of nodes are trying to transmit packets, the node with the highest priority packet succeeds. Development of such protocols involves changes to the Data Link protocol layer of the Ethernet to carry out packet transmission in priority order, and changes to higher layer protocols to allow user processes to define, specify, and interpret the priority of data. In addition, the performance of the possible alternatives will be studied to evaluate the suitability of the developed protocols for multimedia traffic. Simulation models will be developed for the designed priority protocols and will be run to extensively evaluate the performance of these protocols.

Thus, the proposed research focuses on the problems associated with supporting multimedia on Ethernet environments as well as studying the performance aspects of the possible alternatives. By using a protocol for priority transmission, multimedia data can be given a higher priority than regular data and thus allowing multimedia data to be delivered in a timely fashion over the Ethernet.

3.2 Research Issues Involved

There are two main issues involved in the design of priority protocols of an Ethernet, namely, functional decomposition of the protocol and the structuring of the protocol in the layered architecture.

1. *Functional decomposition of the protocol:* This deals with decomposing the protocol functions in components and identifying what functions the various components of the protocol must perform. There are two distinct functions of the protocol.

   (a) *Dissemination of state information of nodes.* At any time, the node with the highest priority packet on an Ethernet is allowed to transmit its data on the

other. The lowest level is the Physical Layer, which is concerned with the actual physical data transmission and the hardware involved. This layers deals with the specifications electrical signals and their timing sent over a physical wire. Above this level is the Data Link Layer or the level 2 layer. This layer specifies how data is sent from one host to another host over a single physical network. It includes the concept of data frames, checksums and some error handling. Ethernet communication falls in this layer. Internet Protocol (IP), Transmission Control Protocol (TCP), as well as most other protocols involved in interprocess communication, represent higher level protocols. When several networks are connected together, IP provides point-to-point delivery of packets [2]. IP addresses are divided into two parts, the network address and the host address. In theory, a maximum of 254 hosts can be attached to a single Ethernet. However, in practice, the number of hosts is much smaller.

Ethernets are a baseband bus communication medium that use CSMA/CD (carrier sense multiple access/collision detect) [7]. CSMA essentially means that a node will listen to the channel before it attempts to transmit, to avoid transmitting while another node is transmitting. If a node finds the channel is currently busy, it waits until the Ethernet becomes free and then begins transmitting. This scheme is known as 1-persistence. A collision occurs when two or more nodes attempt to transmit packets on the Ethernet simultaneously. All packets involved in a collision cannot be used and must be retransmitted. Collision detection allows a node to detect if its packet going out on the bus is being corrupted by another node that started transmitting almost simultaneously. When a collision occurs, each node waits for a random amount of time (called an exponential backoff, due to the mean time of the delay or backoff doubling with every consecutive collision) and then tries again. This paradigm attempts to give all nodes an equal chance to transmit their data, while avoiding subsequent collisions due to repeated retransmission attempts [9, 7, 11, 13]. There is no notion of priority at this level. Thus, transmission of multimedia data on an Ethernet does not get any preference and must compete equally with the transmission of regular data traffic.

Ethernet hardware, i.e., network interface, is designed to minimize the load on the CPU. The interface delivers only those packets to its node that are sent to it or broadcast to the entire net and ignores packets intended for other nodes. Broadcast packets are received by all nodes on an Ethernet, but are not propagated to adjoining Ethernets when the system consists of multiple interconnected Ethernets. The network interface can be configured to be in "promiscuous mode" in which it receives all packets sent on the bus [12]. However, this is likely to cause an unacceptable load on the CPU to process and discard all of the undesired packets.

Thus, Ethernet is designed to efficiently transfer regular data traffic which are bursty in nature and for which variable delay is acceptable (e.g., transaction processing [3]). It is not suitable for multimedia traffic which requires constant delay, fast (or real time) delivery, and a large bandwidth.

Token ring networks support the concept of priority data transfer [6] and suitability of priority data transfer on a token ring to multimedia traffic has been studied by Amer et al. [5]. Their study reveals that the priority mechanism on a token ring considerably improves the quality of multimedia data transfer.

1 Introduction and Motivations

Multimedia is becoming important in several domains [8]. By allowing the flexibility of presenting more than one type of information, such as voice, music, still and motion video, in addition to normal text, multimedia computing is entering into office space, businesses, and other professional, non-academic environments.

There are many potential uses of multimedia in the business realm including presentations, lectures, conferences, discussions, and shared work environments. Multimedia provides more information and interaction than that is possible or practical with conventional facilities. Conferences can link people that are too distant to meet face to face and provide a better forum for the exchange of ideas than a 3-way telephone call. Shared applications allow multiple people to work on a project.

While the types of multimedia applications are diverse, they share a common underlying trait: "The multimedia data that is being exchanged from node to node is real-time in nature and needs to be delivered in a timely fashion. Multimedia data is isochronous, that is, it is sensitive to delays and requires a constant amount of bandwidth, as opposed to bursty traffic". Different media types of course have different tolerances, but they all have deadlines and data past its deadline can not be used [4, 14]. Current approaches of transmitting regular data traffic which is bursty in nature and can tolerate a high variance in delay [11, 13] are not suitable for multimedia traffic which is isochronous [14]. In multimedia systems, intelligent transmission of packets based on a data priority or deadline that will result in a much lower number of discarded packets is required [4, 5].

Most business organizations and academic institutions already have LANs (local area networks) installed. A typical environment consists of workstations or PCs connected by a LAN, typically an Ethernet. Multimedia is getting more important to these organizations. As far as the computing power is concerned, the nodes possess sufficient computing resources to support multimedia applications. However, as far as communication is concerned, the current Ethernets do not support high speed multimedia traffic [1]. Current Ethernet protocols do not support real-time, isochronous traffic.

These organizations and institutions have two options for adding multimedia capabilities. First is to attempt to utilize their existing networks by adding some devices and/or software in order to successfully transmit multimedia streams. The other option is use an ATM type network. ATM protocol is designed such that voice and video can be transmitted effectively along with data packets. ATM switches and interfaces will be needed which would add a lot of cost to these organizations. Therefore, a practical solution is to add capabilities to the existing LANs in order to successfully transmit multimedia streams.

2 State of the Art

The ISO Reference Model of Open System Interconnection [10] (also known as the ISO 7-layer reference), divides communication into seven different conceptual layers. The highest is the Application Layer, where different programs or applications communicate with each

Priority Ethernets: Promises and Challenges

Frank Adelstein
Mukesh Singhal
Dept. of Computer and Information Science
The Ohio State University
2036 Neil Avenue Mall
Columbus, OH 43210-1277

Phone: (614) 292-4634
FAX: (614) 292-2911
Internet: {frank,singhal}@cis.ohio-state.edu

February 7, 1994

Abstract

This proposal will address the issues related to the delivery of multimedia traffic on local area networks. Multimedia integrates voice and video along with text and images into existing systems. Traditional local area networks, e.g., Ethernet, were designed to handle regular data traffic which are bursty in nature and for which variable delay is acceptable. Multimedia traffic, on the other hand, requires constant delay in addition to fast (or real time) delivery. Multimedia traffic also requires large amount of bandwidth compared to regular traffic. Multimedia is becoming important in commercial as well as non-commercial organizations and Ethernet is a widely used local area network among most of these organizations; therefore, modification of Ethernet protocols to integrate multimedia streams is an important issue. In this research, the investigator will develop solutions to the problems associated with transfer of multimedia data on Ethernet and will study the performance of these solutions and their suitability to multimedia traffic. Specifically, priority protocols will be developed that will enable priority transmission of data on Ethernets. Multimedia data can be assigned priority over regular data, and the developed priority protocols can be used to deliver multimedia data over the Ethernet in a timely manner. The performance of these protocols will be studied using extensive simulation.