Proceedings of the 1991 Winter Simulation Conference
Barry L. Nelson, W. David Kelton, Gordon M. Clark (eds.)

AN EFFICIENT CHANNEL-FEEDBACK-BASED ADAPTIVE PROTOCOL FOR SCHEDULING VARIABLE-LENGTH MESSAGES ON SLOTTED, HIGH-SPEED FIBER OPTIC LANs/MANs

Subrata Banerjee and Biswanath Mukherjee

Department of Computer Science
University of California
Davis, California 95616

ABSTRACT

This work builds up on the continuation-bit approach in conjunction with a probabilistic scheduling strategy called the p_i-persistent protocol. This combination has been proposed as an efficient multiple access protocol for transmitting variable-length messages on slotted, high-speed, fiber-optic, bus-structured local and metropolitan area networks (LANs/MANs). Specifically, our present work extends the previous static model on this approach to the dynamic case so that the protocol can be implemented. Now, the model considers network loading changes, changes in message length distributions, etc. in order to preserve the desired form of the protocol's operation (fairness), by adaptively adjusting the network operating parameters.

A dynamic simulation model of the protocol using the continuation-bit approach has been formulated, and channel feedback information as discussed above has been employed to calibrate the operating parameters in real time. Since the goal was to study the operation of the dynamic protocol with variable-length messages, variable-bit-rate (VBR) coded (i.e., compressed) video was chosen as the traffic to be carried by the network in the illustrative examples. The simulation model indicates that channel feedback is a good approach for dynamically adjusting the network parameters as the network loading and the message lengths fluctuate.

1 INTRODUCTION

Most of the proposed high-speed LAN/MAN protocols (e.g., DQDB (IEEE Standard 802.6, 1990), LOCOST (Limb 1989), p_i-persistent (Mukherjee and Meditch 1988)) operate on slotted channel(s), and, in Single-bus Unidirectional Broadcast System (SUBS) or Dual-bus Unidirectional Broadcast System (DUBS) environments (Fig. 1). The slot size is conveniently chosen to equal a packet's transmission time. However, since messages arriving at a station for transmission can have variable lengths, they need to be divided into fixed-length packets. This necessitates some overhead to be associated with each packet. This overhead usually carries control information such as source and destination addresses. Thus, while the synchronous nature of the slotted system provides easier control, a significant amount of overhead is also introduced. This limitation of slotted systems has been emphasized in several recent papers (Kim 1990, Roucher 1990).

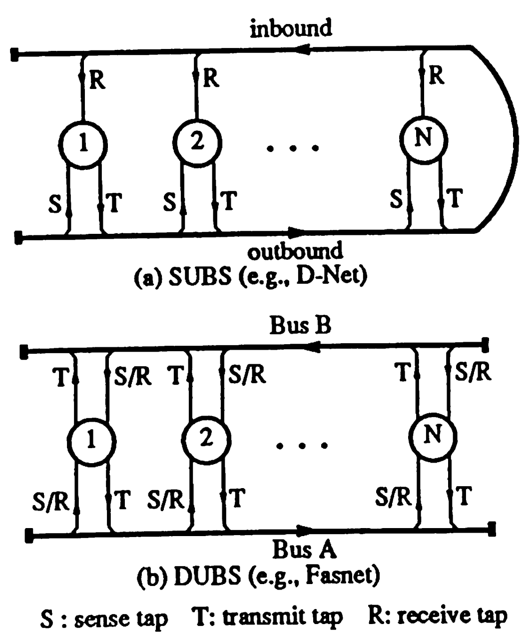


Figure 1: Topologies For Unidirectional Bus Networks

To overcome the slotted system's overhead for multi-packet messages, a method was proposed by

Mukherjee and Kamal (1991) which is, in a sense, a compromise between a totally-slotted system (e.g., DQDB) and an unslotted system (e.g., FDDI). The basic idea is that the entire addressing information be provided in the first packet of a message, as in any conventional multi-packet message scheduling method. However, none of the subsequent packets in the message contain any addressing information; instead, each of them has its first bit, called the *continuation-bit*, marked specially to indicate that this is a *continuation packet*. The message's last packet includes an *END delimiter* following the last message information bit. It has been shown by Mukherjee and Kamal (1991) that even though packets from several message sources can be interleaved, there is still enough intelligence in the system to properly reconstruct the messages. This method, which is called the *continuation-bit approach*, was originally proposed in the study of an efficient slotted-ring protocol (Kamal and Hamacher 1990).

In this work, we study the dynamic version of the continuation-bit approach in combination with the p_i-*persistent* protocol which uses a probabilistic scheduling technique. This combination has been proposed as an efficient multiple access protocol for transmitting variable-length messages on slotted, high-speed, fiber-optic, bus-structured (Fig. 1) local and metropolitan area networks (LANs/MANs) ((Mukherjee and Kamal 1991). The target network environment is shown in Fig. 1(a) with stations numbered 1, 2, ... , N, with Station 1 being the most upstream. Specifically, our current work extends the previous static model on this approach to the dynamic case so that the protocol can be implemented. The enhanced model takes into account variations in the network traffic pattern, including the individual stations' message arrival rates and their message length distributions. Thus, the dynamic model is capable of preserving the desired form of the protocol's operation (fairness), by adaptively adjusting the network operating parameters. These adjustments are performed at the network stations in a dynamic and distributed fashion using feedback information from the channel. The channel, because of its broadcast nature, contains a wealth of information. For example, by monitoring the channel activity and by considering a window of time in the past, a station can determine the number of currently-active stations, and it can estimate their utilizations, the utilization of the entire network, and the first two moments of the message lengths generated by the individual stations. All this information is required for controlling the network's operating parameters properly.

A dynamic simulation model of the protocol using the continuation-bit approach has been developed. Information from channel feedback, as discussed above, has been utilized to calibrate the network operating parameters in real time. Since the goal was to study the operation of the dynamic protocol with variable-length messages, variable-bit-rate (VBR) coded (i.e., compressed) video was chosen as the traffic to be carried by the network in the illustrative examples. Although a number of video models are appearing in the literature, we chose the "beta-distribution" video model (Koga et al. 1981, Fig. 7) because of its simplicity in modeling. Results from the simulation model indicate that channel feedback is a good approach for dynamically adjusting the network parameters as the network loading and the message lengths fluctuate.

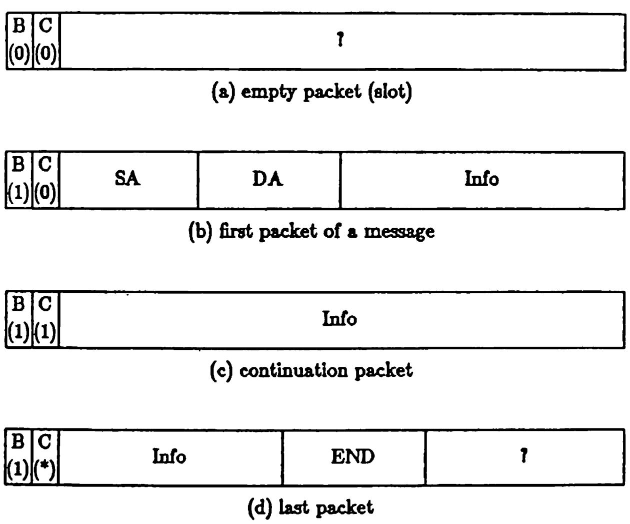
This paper is organized as follows. In Section 2 we describe the continuation bit approach and present some of the analytical results from previous work. The simulation model is presented in Section 3. Some representative numerical results are discussed in Section 4. The paper concludes with Section 5.

2 CONTINUATION BIT APPROACH

2.1 Message Transmission

Let us first describe the packet format which is also shown in Fig. 2. Additional packet information which are not relevant for our purposes here, e.g., a preamble for slot synchronization, are not shown. The first two bits in a packet are called the the busy-bit (B) and the continuation-bit (C) respectively. An empty slot has B=0, and although its C bit could be arbitrary, we choose C=0 for empty slots as well. A slot carrying a packet has B=1. Additionally, the first packet in a message uses C=0. It is then followed by the source and destination address fields, after which the message information starts. Subsequent packets in a message employ C=1, and they are called continuation packets. The last message information bit is followed by a special bit pattern, called the END delimiter, to indicate the end of the message. Obviously, in order to ensure that this pattern does not occur in the message information, bit stuffing must be used (Tanenbaum 1988).

The head station, which is the most upstream station on an unidirectional bus, continuously generates empty slots with B=0 and C=0. At any station, scheduling the transmission of the first packet in a message will depend upon the access algorithm being employed. In this study, we choose a variation of the p_i-*persistent* protocol (the regular p_i-*persistent* protocol, however, is described in (Mukherjee and Meditch 1988)). Under this protocol, which operates on a slotted channel, a station i, with a message to send, fragments the message into one or more fixed-length packets in accordance with the continuation-bit approach. (See Fig. 2 for the packet format.) Also, the station continuously checks the B bit of every slot that passes through its sense tap. If it finds that B=1, then it lets the slot go by since the slot is already carrying a packet. But if it finds an empty slot (B=0), and if it has not yet transmitted the first packet of its message, then it persists in transmitting that packet in such empty slots with probability p_i (where p_i is to be determined later). Or, alternately, Station i can pass along untouched a number of empty slots that it sees after its message arrival (say the first $\lceil 1/p_i \rceil - 1$ of them); it may then transmit into the next empty slot with probability one. When the station transmits the first packet, it sets B=1 and C=0 (i.e., this is not a continuation packet).



B:	busy bit	SA:	source address
C:	continuation bit	DA:	destination address
Info:	information bits	END:	END delimiter
?:	arbitrary bits		

*Last packet of a multi-packet message employs C=1. The last packet will also be the first packet in the case of short messages where the END delimiter fits into the first packet; then, this packet will have C=0. Note also that the END delimiter can span over two consecutive packets.

Figure 2: Packet Format

After transmitting the first packet, a station continually checks the following slots that arrive at its interface to see if they are empty or not. If the slot is empty, it sets B=1 and C=1, and transmits, with probability one, its information as a continuation packet. If the slot is busy, it simply waits for the next slot. It repeats this process until its message (including the END delimiter) is completely transmitted.

Note that (B,C)=(0,1) is an invalid combination, but this combination can be employed for control packets if stations are equipped at their network interfaces with buffers of size two bits or more, so that they can determine the status of a slot by examining two bits at a time.

2.2 Message Reception

It is quite apparent that the transmission process can interleave message packets from different sources. However, since the bus is unidirectional, the interleaving process follows a rather straightforward pattern.

An arbitrary station, say Station i, sees the following activity at its receive tap R. All packets corresponding to a message from Station 1 are always uninterrupted (call this a busy period (BP) initiated by Station 1). Station 2's activity is interrupted by Station 1's BP occasionally, so reconstruction of Station 2's message is a little involved Station 3's BP is in turn interrupted by BPs from Stations 1 and 2, and so on. Note that there is no idle slot in a BP. Even though no explicit addressing information is used in continuation packets, the use of the C bit along with an explicit END delimiter at the end of the message is sufficient to reconstruct all of the messages at any Station i. To do so, each station employs a *stack mechanism*. The stack is initially empty (corresponding to an idle channel), and it is used to keep track of the address of the station that is currently transmitting on the channel. Stack activity starts as soon as a BP is observed, and the stack grows with more BP interruptions, viz. when slots with (B,C) = (1,0) arrive. Packets collected from the channel are associated with only the address on top of the stack. The stack is decremented by one when it encounters an END delimiter. When the channel is idle again, the stack should have become empty.

For message reconstruction, simpler variations of the above mechanism can also be employed if desired. For example, instead of a stack, a station can be equipped with a counter that keeps track of the depth of nesting in the current BP. Of course, if the station is also the recipient of one or more of the nested messages, it must keep track of the corresponding nesting levels in order to reconstruct for itself the corresponding messages it is expected to receive.

2.3 Some Analytical Results

Because of the system's complexity, an accurate analytical model is difficult to formulate, except for trivially small systems. Hence, approximations are employed to analyze the system. Some of the approximate analytical results obtained by Mukherjee and Kamal (1991) are given below.

In (Mukherjee and Kamal 1991) two types of traffic are analyzed. In the light traffic approximation model, it is assumed that all slots arriving at a station are equally likely to be either full or empty, independent of their past status, and their probability of occupancy depends only on the station's index (or location). Station j's average message delay for light traffic is given by

$$\overline{D}_j = \left[\sum_{k=1}^{j-1} u_k \left\{ 1 + \frac{(\overline{R}_k - 1)}{1 - \sum_{i=1}^{k-1} u_i} + \frac{1}{1 - \sum_{\substack{i=1 \\ i \neq k}}^{j-1} u_i} \right\} \right] + \frac{1/p_j + \overline{M}_j - 2}{1 - \sum_{i=1}^{j-1} u_i} \quad (1)$$

where j = 1, 2, $\cdots$, N, $\overline{M}_i$ and $\overline{M_i^2}$ are the first two moments of the message length in slots at Station i,

$$\overline{R}_k = (\overline{M_k^2} + \overline{M}_k) \,/\, 2\overline{M}_k$$

is the mean residual length of Station k's message at light loads, and

$$u_i = \frac{\overline{M}_i}{\dfrac{1}{1 - e^{-\lambda_i}} + \overline{D}_i} \quad (2)$$

is Station i's utilization, where λ_i is the message arrival rate at Station i. Thus, for a given set of operating conditions $\{\lambda_i, p_i, \overline{M}_i, \overline{M_i^2}\}$, eqns. (1) and (2) can be solved numerically to obtain the mean message delays.

Equation (1) can be understood as follows. Since $\Sigma_{i=1}^{j-1} u_i$ is the combined utilization of stations upstream of Station j, $z_j = (1 - \Sigma_{i=1}^{j-1} u_i)$ is the fraction of bandwidth seen by Station j. Since Station j persists in transmitting the first packet of its message with the geometric parameter p_j and subsequent packets with probability one, it takes $1/p_j$ empty slots on the average to transmit its first packet and $\overline{M}_j - 1$ empty slots on the average for its subsequent packets. Thus, if Station j always found empty slots, its average message delay would be $Y_j = (1/p_j + \overline{M}_j - 1)$ slots. However, since it finds only a fraction z_j of the arriving slots to be empty, Y_j is expanded to $w_j \times 1 + (Y_j - 1)/z_j$. The second and subsequent slots in Y_j are explained by the second term in Equation (1); the first slot in Y_j, however, deserves special treatment. Specifically, it takes Station j w_j slots on the average from the instant of its message arrival until it encounters the first empty slot. Note that $w_j \neq 1/z_j$ as explained below.

The message arrival (i.e., the first slot in Y_j) is expected to occur in a longer duration of busy slots from upstream, rather than in an average one. This is accounted for by the first term which captures the fact that an upstream station (say Station k) is transmitting with probability u_k in the slot of the tagged message's arrival. The average residual number of slots remaining in Station k's message transmission is $\overline{R}_k$, but, again, the second and subsequent residual slots can be interrupted from upstream. And, finally, when Station k's transmission is over, other stations upstream of Station j can still contend for empty slots.

A heavy traffic approximation model is also developed in (Mukherjee and Kamal 1991) with the assumption that all of the stations always have packets to send. Station j's average message delay for heavy traffic is given by

$$\overline{D}_j = \left[\frac{\overline{x}_j}{\overline{M}_j - 1} \right] \frac{1}{p_j} + \overline{x}_j \quad (3)$$

where x_j is defined as the number of additional slots needed by Station j to complete its message transmission *after* it has already transmitted its *first* packet. (See Fig. 3.) The above equation indicates the following:

(1) out of $\overline{x}_j$ slots that Station j sees, it finds only $\overline{M}_j - 1$ of them to be empty on the average (which follows from the definition of x_j) so that the fraction of empty slots arriving at Station j is $(\overline{M}_j - 1)/\overline{x}_j$;

(2) it needs $1/p_j$ empty slots on the average to transmit its first packet; and

(3) finally it needs an additional $\overline{x}_j$ slots to complete its message transmission.

It has been found in (Mukherjee and Kamal 1991) that the light traffic model is a reasonably good approximation for busses with total offered loads of up to 0.8 packets/slot. For higher loads, the heavy traffic model provides more accurate results.

2.4 Need for Fair and Adaptive Model

In order to provide fair and efficient service, the network must adapt to the changing traffic patterns. As a fairness criterion either *equal average message delay* or *equal average message throughput* or *equal average message blocking*, at all the stations, can be used. In our illustrative examples, we chose *equal average message delay* as our fairness criterion. By equating $\overline{D}_i$ with $\overline{D}_j$ for i,j = 1,2, ... ,N and by using p_N=1, a system of equations can be formed, solving which, the proper set of $\{p_i\}$ that achieve fairness can be obtained. The other fairness criteria can also be handled similarly, if desired.

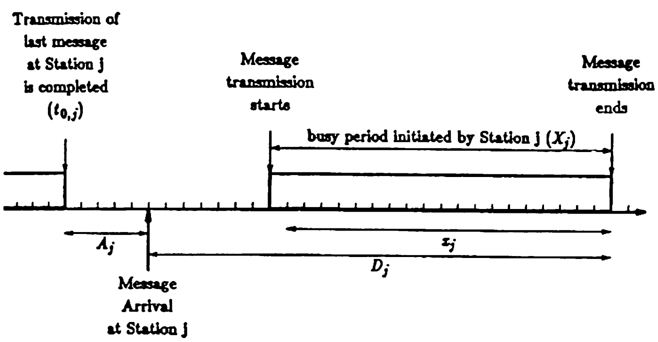


Figure 3: Components of Message Delay at Station *j*

In practice, the network traffic pattern is expected to change with time. To keep up with the fluctuating traffic, the network's operating parameters (i.e., the set of $\{p_i\}$) must be dynamically updated (by using the light or heavy traffic model, depending on the load), preferably in a distributed fashion such that the desired operating condition is maintained.

An adaptive version of the protocol, which regularly updates the network's operating parameters in a decentralized manner, is considered in the next section.

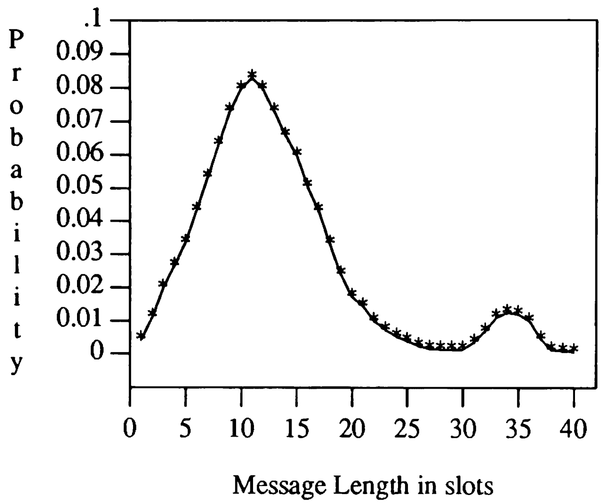
3 SIMULATION MODEL

So far, a static view of the network has been considered. But what happens when the network loading changes, some stations leave the network, others join it, or message length distributions change? Obviously, in order to preserve the desired form of fairness, the network operating parameters, viz. the $\{p_i\}$, must be adjusted accordingly.

An attractive possibility is to adjust the p_i in a dynamic and distributed fashion using feedback from the channel, as in a SUBS example (Mukherjee et al. 1991). The channel, because of its broadcast nature, contains a wealth of information. For example, by monitoring the activity on the inbound channel and by considering a window of time in the past, a station can determine the number of currently-active stations, and it can estimate their utilizations, the utilization of the entire network, the first two moments of the message lengths of the individual stations, etc. Measurements from the outbound channel activity can be used to determine the distribution of time between empty slots arriving at Station *i*.

A dynamic simulation model of the protocol using the continuation-bit approach has been formulated for the SUBS structure (Fig. 1(a)), and channel feedback information discussed above has been employed to calibrate the p_i in real time. Specifically, if the estimated network utilization is lower than 0.8, the light-traffic approximation model is used for adjusting the p_i so that the station will achieve a proper $\overline{D}_i$ which will be fair. For higher measured utilizations, the heavy-traffic model is used to adjust the p_i. The simulation programs are written in "C" and are run on a Sun Microsystems SPARCstation 1. *Equal message delay* is used as the criterion for determining the p_i. A station only employs local information at its receive tap on the inbound channel, and it employs the stack mechanism to parse the fragments of various messages to determine the distributions of the message lengths generated by the various stations. For estimating these various quantities, each station updates its p_i after every τ slots, and each such update is based on the estimates of the various quantities over the most recent W slots, which is called the *window of estimation*. Note that, although the analytical models assume no propagation delays between stations, the latter will have an effect on the dynamic simulation model because the windows used by the various stations for estimating the various quantities will not exactly be synchronized. But we find from the simulation results (as also in (Mukherjee et al. 1991)) that, if the offered traffic is stationary, the effect of the propagation delay is not significant.

Since our goal was to study the operation of the dynamic protocol with variable-length messages, we chose variable-bit-rate (VBR) coded video as the traffic to be carried by the network in the following illustrative examples. Although a number of video models are appearing in the literature, we chose the "beta-distribution" video model developed by Koga et al. (1981) because of its simplicity in modeling. Specifically, we assume the following video model at each station. A video message is generated at each station periodically every 200 slots. Each message can be of length between 1 and 40 slots (ignoring overhead) according to the measured results of the video "beta distribution" in (Koga et al. 1981). This distribution is the long-term average video distribution (Fig. 4) with $\overline{M}_i = 12.167$ and $\overline{M_i^2} = 179.997$ for all *i*. We remark that, although the measured video output was specified in Mbps (between 1 and 40) in (Koga et al. 1981), we are keeping our study sufficiently general by making no assumptions on the channel rate or slot length. This is accomplished by changing the video output units from Mbps to slots.



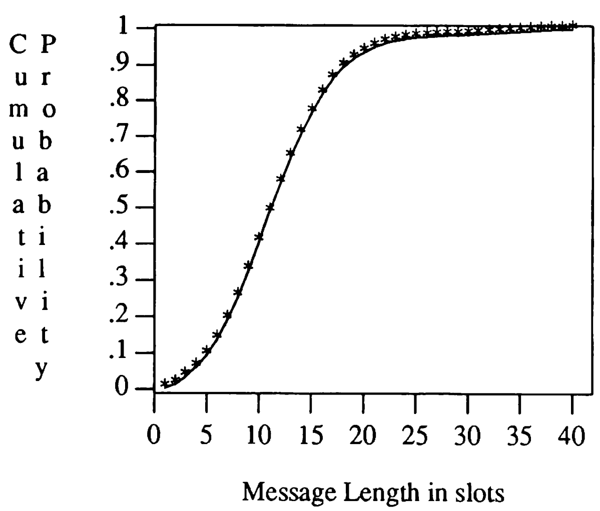


Figure 4: Beta Distribution Used In Modeling Video (Koga et al. 1981)

For determining the performance of the protocol with video service, we will also measure the fraction of a station's traffic that is not delayed from transmission beyond a certain maximum D_{max}. Thus, although the analytical model assumes a single-message buffer at each station, this dynamic video simulation model can buffer more than one video message at each station as long as none is older than D_{max}; and as soon as a message gets older than D_{max}, it is discarded.

In the simulation model, we use W=150,000 slots, τ=100,000 slots, and the interstation distance equal to one slot on both the inbound and outbound channels. These values of W and τ resulted in quite accurate estimation of the system parameters, increasing these numbers did not improve the performance significantly. Messages that have been completely transmitted as well as completely received within the window are used for building the estimates. Also, the arrivals of the first video messages at the various stations are randomized; otherwise, since the arrivals are periodic, the performance results are lop-sided, in favor of the upstream stations.

4 NUMERICAL RESULTS

Table 1 shows the measured performance of the system with five video stations and D_{max}=200 slots. In the analytical part (Table I(a)), note that the estimates on the station utilization, station message arrival rate, and the first two moments of the station message lengths are shown for Station i only; actually, these statistics are compiled at Station i for all of the other stations as well. However, the statistics compiled by Station i for all stations are not shown to conserve space; only Station i's own compiled statistics from channel feedback are shown. The estimates shown are quite representative of those that are not shown. Although the target average analytical message delay was 17.1 slots in this example, the corresponding simulation value turned out to be lower. This could possibly be due to the well-behaved periodic arrivals, while the analytical model assumes random arrivals. The range on the average delays across the individual stations was observed to be quite wide, however. It was also observed that the range, the station number with the maximum delay, and the station number with the minimum delay would change when a different random pattern of first arrivals at the various stations was chosen. This was because of the differences in the observed traffic within the windows at the various stations employed for the estimation purposes.

Table 2 shows the measured performance of the system with different numbers of video stations (5, 10, and 15) and different values of D_{max} (200, 400, and 600 slots). Shown are the average system-wide message delay, the identity of the station with the highest fraction of delayed messages, and the value of this fraction (denoted by $Drop_i$). For 5 or 10 video stations, the system is sufficiently lightly loaded (corresponding to approximate system throughputs equal to 0.3 and 0.6, respectively) so that almost all messages are cleared within a 200-slot delay. For 15 video stations, however, the system load is sufficiently high so that about 97.5% of the messages at Station 2, and larger percentages at other stations, are able to meet the 200-slot delay deadline. An additional 1% of the messages at Station 2 can beat the 400-slot deadline. And less than 0.9% of the messages cannot beat the 600-slot deadline. These results are encouraging, although we feel that the protocol may need to be properly reconstructed for accommodating real-time traffic.

Table 1: Performance of A Dynamic System with 5 Video Stations and $D_{\max}$ = 200 slots

(a) analytical results at last p_i-update

Stn. No. (i)	p_i	target $\bar{D}_i$	estimated quantities u_i	λ_i	$\overline{M}_i$	$\overline{M_i^2}$
1	0.169	17.1041	0.0558	0.00498	12.189	181.951
2	0.214	17.1041	0.0553	0.00497	12.107	179.530
3	0.284	17.1041	0.0548	0.00498	11.981	173.603
4	0.606	17.1041	0.0578	0.00499	12.614	192.667
5	1.000	17.1041	0.0545	0.00496	11.963	171.385

(b) simulation results

Stn. No. (i)	u_i	D_i
1	0.06110	18.4157
2	0.05999	17.2768
3	0.06076	15.8456
4	0.06217	14.6789
5	0.06091	13.1876

Average system delay = 15.8733
Range of delay = 18.4157 – 13.1876 = 5.2281
System throughput = 0.30495

Table 2: Performance of A Dynamic System with Different Numbers of Video Stations And Different Values of $D_{\max}$

no. of video stns.	$D_{\max}$ (slots)	$\bar{D}$ (slots)	S	$Drop_{i,max}$ (percent)	corres-ponding stn. no. (i)
5	200	15.8733	0.30495	0.000011	1
	400	15.8731	0.30495	0	-
	600	15.8731	0.30495	0	-
10	200	26.7507	0.60745	0.000011	1
	400	26.7507	0.60746	0	-
	600	26.7507	0.60746	0	-
15	200	109.3889	0.77599	0.024107	2
	400	191.0566	0.83298	0.014991	2
	600	241.2068	0.87048	0.008558	2

Results with larger interstation propagation delays did not show any significant changes. The effect of different window sizes, different update periods, changing traffic intensities, etc., were studied in (Mukherjee et al. 1991) for the single packet per message case. Their effect for the continuation-bit case were not significantly different; hence, they are not shown here.

5 CONCLUSION

In this paper we have studied the dynamic version of the continuation-bit approach employed with the p_i-persistent protocol. We have also outlined a method for implementing this approach by utilizing feedback information from the channel. From the feedback information, network loading conditions are estimated in a distributed fashion. These estimates, which are used to adjust the p_i, are updated periodically such that the protocol can follow the variations in the offered traffic patterns.

A dynamic simulation model for the protocol employing the continuation-bit approach was also formulated. In our illustrative examples, network traffic was chosen to be variable-bit-rate (VBR) coded video. Even at a network utilization of 0.6, it was found that almost all video messages were transmitted from the station buffers before the next video message arrived. Although these performance results are encouraging, we feel that the protocol does not provide good support for real time traffic. This is due the large window size required for accurate estimation of the system parameters. However, smaller window sizes can also be used for faster response to real time traffic at the expense of accuracy. However, future studies are to be conducted in this direction to make the protocol fully integrated with support for real-time and prioritized traffic.

Finally, we remark that the continuation-bit approach has also been applied to the DQDB network (Kamal 1991, Banerjee and Mukherjee 1991).

ACKNOWLEDGMENTS

This work has been supported by the US Air Force Office of Scientific Research (AFOSR) under Grant No. 89-0292.

REFERENCES

Banerjee, S., and Mukherjee, B. 1991. "Incorporating Continuation-of-Message Information, Slot Reuse, and Fairness in DQDB Networks," *Computer Networks and ISDN Systems*, to appear, 1991. (Also

Division of Computer Science, University of California, Davis, Tech. Report No. CSE-90-42, October 1990.)

IEEE Standard: Distributed Queue Dual Bus (DQDB) Metropolitan Area Network (MAN), P802.6, 1990.

Kamal, A. E., and Hamacher, V. C. 1990. "Utilizing bandwidth sharing in the slotted ring," *IEEE Trans. Comput.*, vol. 39, pp. 289-299, March 1990.

Kamal, A. E. 1991. "Efficient multi-segment message transmission with slot reuse on DQDB," *IEEE INFOCOM '91*, Bal Harbor, FL, April 1991, pp. 869-878.

Kim, B. G. 1990. "Packet delays in the IEEE 802.6 DQDB protocol," *Proc., IEEE Intl. Conf. Commun. '90*, pp. 1692-1696, Atlanta, GA, April 1990.

Koga, T., Iijima, Y., Iinuma, K., and Ishiguro, T. 1981. "Statistical performance analysis of an interframe encoder for broadcast television signals," *IEEE Trans. Commun.*, vol. COM-29, pp. 1868-1875, Dec. 1981.

Limb, J. O. 1989. "Load-controlled scheduling of traffic on high-speed metropolitan area networks," *IEEE Trans. Commun.*, vol. COM-37, pp. 1144-1150, Nov. 1989.

Mukherjee, B., Lantz, A. C., Matloff, N. S., and Banerjee, S., 1991 "Dynamic control and accuracy of the p_i-persistent protocol using channel feedback," *IEEE Trans. Commun.*, to appear (June 1991). (Also, in *Proc., IEEE INFOCOM '89*, Ottawa, Canada, April 1989, pp. 858-865, April 1989.)

Mukherjee, B., and Kamal, A. E. 1991. "Scheduling Variable-Length Messages On Slotted, High-Speed Fiber Optic LANs/MANs Using The Continuation-Bit Approach," *IEEE INFOCOM '91*, Bal Harbor, FL, April 1991, pp 678-687.

Mukherjee, B., and Meditch, J. S. 1988. "The p_i-persistent protocol for unidirectional broadcast bus networks," *IEEE Trans. Commun.*, vol. COM-36, pp. 1277-1286, December 1988.

Roucher, E. Y. 1990. "MAN overboard?" *Data Communications*, vol. 19, no. 6, pp. 202, May 1990.

Tanenbaum, A. S. 1988. *Computer Networks, 2nd ed.*, Englewood Cliffs, NJ: Prentice-Hall, 1988.

AUTHOR BIOGRAPHIES

SUBRATA BANERJEE received the B.Tech. degree in Computer Science and Engineering from Indian Institute of Technology, New Delhi, India, in 1988. Currently, he is pursuing his doctoral degree in Computer Science at the University of California, Davis. During the summer of 1987, he was an intern at the International Computers Limited, Delhi, India. At UC Davis, he served as a Teaching Assistant, Research Assistant and Post Graduate Researcher. His current research interests are in modeling, analysis, simulation and performance enhancements of high-speed local and metropolitan area networks and lightwave networks. Mr. Banerjee has been a recipient of tuition fellowships at Davis, the Institute Merit Award (twice) at IIT, Delhi and the National Talent Search Scholarship in India.

BISWANATH MUKHERJEE received the B.Tech. (Hons) degree in Electronics and Electrical Communications Engineering from Indian Institute of Technology, Kharagpur (India) in 1980, the M.S. degrees in Electrical Engineering (1983) and in Computer Science (1984), both from Southern Illinois University, Carbondale, and the Ph.D. degree from University of Washington, Seattle, in 1987. He has been a Customer Engineer at Operations Research Group Systems, India; a Graduate Assistant and an Instructor at Southern Illinois University; an Instructor at Southeastern Illinois College, Harrisburg, Illinois; and a Teaching Assistant and a Research Assistant at the University of Washington, Seattle. At the University of Washington, he held a GTE Teaching Fellowship and a General Electric Foundation Fellowship. Presently, he is an Assistant Professor of Computer Science at the University of California, Davis. Dr. Mukherjee served on the Technical Program Committee of the IEEE INFOCOM 89 and 90 conferences. He is also serving in the INFOCOM 92 and GLOBECOM 92 Technical Program Committees. His research interests include high-speed local and metropolitan area networks, lightwave networks, integrated services networks, network interconnection, and network security monitoring. He is a member of the Tau Beta Pi and Phi Kappa Phi Honor Societies.