ELSEVIER

Computer Networks and ISDN Systems 29 (1997) 717–722

COMPUTER NETWORKS and ISDN SYSTEMS

Dynamic p_i-persistent protocol with reduced station hardware [1]

M. Hamdi *, W.C. Ming

Department of Computer Science, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

Abstract

The p_i-persistent protocol has received a lot of attention from the research community and is considered to be a very suitable candidate for multiple-access communication over high-speed metropolitan area networks (MANs). The p_i-persistent is a unit-capacity and distance-insensitive protocol. Further, it is a very *fair* protocol. In order to make the p_i-persistent protocol adaptive to changing load conditions, the inventors of the protocol proposed a novel dynamic algorithm that continuously adjusts the stations' probabilities p_i's at their proper levels as governed by the offered traffic. While this dynamic algorithm achieves extremely good results, it requires $N+1$ counters at the interface of each station to the network for an N-station network. Moreover, it requires a large processing time for updating the p_i's of all stations which can have a big effect on the behavior of the protocol. In this paper, we propose a *variant* of this dynamic algorithm that solves these two problems while yielding *exactly* the same results as the original algorithm. Our new algorithm requires only a constant of 4 counters per station no matter how many stations are attached to the network. Further, the processing time for updating the p_i's is a lot less than that of the original dynamic algorithm. Thus, with our new algorithm, the p_i-persistent protocol seems to be the ideal candidate for future generation MANs. © 1997 Elsevier Science B.V.

Keywords: Dynamic control; Hardware cost; MAC protocols; MAN; p_i-persistent

1. Introduction

There have been two schools of thought in the networking research community towards designing future generation Local/Metropolitan Networks (LANs/MANs). One school of thought suggests the design of these networks to be based on the Asynchronous Transfer Mode (ATM) of the B-ISDN world [1]. The second school of thought suggests the design of these networks to be based on broadcast fiber-optic channels primarily due to the unidirectional nature of a fiber-optic medium and its cost-effectiveness [2,3]. Both camps have shown some of the advantages and disadvantages of both schemes. However, we do believe that both schemes will co-exist together especially given the fact that ATM cells can be naturally mapped onto the transmission slots of broadcast fiber-optic LANs/MANs. This, in turn, would make the interface between ATM LANs/MANs with a Broadcast fiber-optic LANs/MANs quite easy. Moreover, in CCITT Recommendations I.211 [4], a comprehensive list of broadband applications supported by B-ISDN is given. It is argued in [5] that the same list can be taken as a guide for applications that may be supported by broadcast fiber-optic LANs/MANs even if the development emphasis for these networks is the reverse of that of B-ISDN, with connectionless data communications coming first and video services last. With that said, we take the second line of thought in this paper and assume the

* Corresponding author. Email: hamdi@cs.ust.hk.

[1] This research work was supported in part by the Hong Kong Research Grant Council under the grant RGC/HKUST 100/92E.

PII S0169-7552(96)00114-6

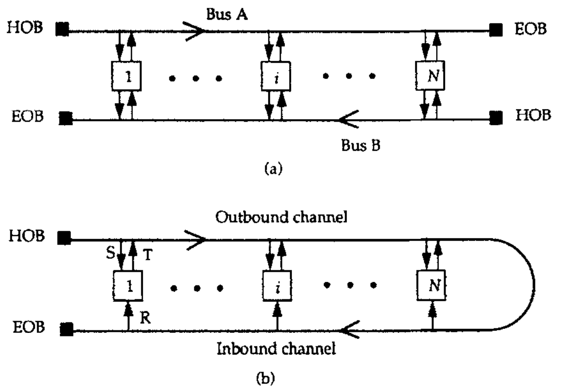

Fig. 1. Unidirectional bus network topologies for MANs.

design of LANs and MANs to be based on broadcast fiber optic networks as shown in Fig. 1. However, with this design choice come the challenge of designing Medium Access Control, or MAC, protocols for these networks.

MAC protocols have been the subject of a vast amount of research over the past two decades. One reason for this research is that all higher layer services are built on the fundamental packet transfer service which is provided by the MAC sub-layer, and it is the MAC protocol which determines the characteristics of this fundamental service. Hence, improvements to MAC services result in improved system performance, while the provision of new MAC services means that new applications can be developed. Advances in technology have led to significant changes in the constraints for communication networks. With the advent of optical fiber as a transmission medium which is projected to have speeds beyond 1 Gbits/sec, spanning distances in the hundreds of kilometers, and proving a wide spectrum of new services, more strict constraints on message/packet/slot processing times are imposed at the intermediate nodes, thereby providing a new challenge to MAC-level protocol designers.

Consequently, a MAC-level protocol designed to operate in this environment must satisfy certain requirements.

(1) It must be simple enough to be directly implemented in hardware, so that it can exploit the bandwidth capacity offered by optical fiber.
(2) It must be fair; i.e., the throughput and transmission delay of a station must be independent of its location within the network.
(3) The throughput of the protocol must be independent of the network's size and transmission rate. As a result, the MAC protocol should be equally suitable for either a LAN or a MAN networking environment.
(4) The access delay of the stations should be bounded. This ensures that a process will always be able to transmit at regular intervals regardless of the current network load. This is especially important for real-time applications such as voice and/or video transmission.
(5) It must be flexible enough to satisfy heterogeneous traffic demands which are typical in an integrated services networking environment.
(6) The format of the transmission slot should be compatible with an ATM cell so that the interface of these networks to ATM networks becomes easy.

Recently, several new MAC protocols for high-speed LANs and MANs have been proposed with the aim of meeting some or all of the requirements listed above, including *Distributed Queue Dual Bus* (DQDB) [6], *P_i-Persistent* [7], *Load-Controlled Scheduling of Traffic* (LOCOST) [8], and *Cyclic Reservation Multiple Access* (CRMA) [9]. Of particular interest to us in this paper is the p_i-persistent protocol since it can theoretically achieve a bus utilization of nearly the bus capacity while providing fair access to all stations under *any* loading scenario. Because of these characteristics, it has been receiving a lot of attention from the research community [10–13]. In the *static* case [7], the p_i-persistent protocol assumes the traffic characteristics of each station and that of the whole network to remain unchanged. Hence, it has no mechanism to properly adjust the stations' probabilities p_i to respond to changing load conditions (where the p_i's are parameters of the p_i-persistent protocol). The *dynamic* version of the p_i-persistent protocol [10] is a probabilistic scheduling scheme that relies on the station's ability to monitor the channel and observe feedback on the bus in order to adapt to changing network conditions. Under dynamic control of the protocol [10,11], the stations use overlapping time windows to estimate the mean packet arrival rates at the other stations based on traffic that is seen by all stations, so the effects of the network characteristics is minimal.

This dynamic control algorithm is a very novel scheme and is extremely effective such that it continuously adjusts the stations' probabilities p_i at their proper levels as governed by the offered traffic. For this reason it has been adopted by various researchers [13]. However, as was indicated by the inventors of this algorithm [10,11], this dynamic control algorithm has two drawbacks. First, since this algorithm is required to monitor the traffic of each station in the network in addition to the whole traffic of the network, each station must have $N+1$ counters, one counter to keep track of the traffic of the whole network and N counters to keep track of the traffic generated by the N stations attached to the network. As a result, the station's hardware can be costly especially when a very large number of stations are attached to the network. Further, upon the addition of each new station, a new counter has to be inserted into the stations' hardware which can be very unpractical. Second, updating the p_i's using this dynamic algorithm can take a considerable amount of processing time. This, in turn, can have a big effect on the behavior of the protocol.

In this paper, we present a *variant* of this novel dynamic control algorithm that requires only 4 counters per station to continuously adjusts the stations' probabilities p_i at their proper levels as governed by the offered traffic no matter how many stations are attached to the network. Moreover, we get *exactly* the same results as that of the original dynamic control algorithm. Also, the processing time for updating the p_i using our algorithm is a lot less than that of the original dynamic control algorithm.

This paper is organized as follows. Section 2 overviews the p_i-persistent protocol. In Section 3, we present the *original* dynamic control algorithm and the *variant* algorithm proposed in this paper, and we compare them together. Finally, we conclude the paper in Section 4.

2. The p_i-persistent protocol

The p_i-persistent protocol described in this paper is for folded bus networks as shown in Fig. 1(b). There are N stations attached to the network which are numbered sequentially from 1 through N, with station 1 being at the head of the outbound channel. Each station has a transmitter T on the outbound channel and a receiver R on the inbound channel. Further, each station has the ability to sense on both channels; for the outbound channel a station can sense the transmissions made by all upstream stations using a separate tap S, and for the inbound channel a station can sense all transmissions made on the bus using its receive tap R as illustrated in Fig. 1(b). Using this network, the time is slotted, with each slot equaling a packet's transmission time. If a slot contains a packet transmitted by a station, it is *full*, otherwise, it is *empty*. Empty slots are generated at the head of the outbound channel and traverse the entire length of the bus, visiting each station in sequence. The p_i-persistent protocol provides for fully distributed control of access to the bus network. Next, we review the protocol's static algorithm and then its dynamic control version.

2.1. Static p_i-persistent protocol

The static p_i-persistent protocol is applied by each station as follows [7]:

Step 1: At the beginning of each time slot, the station checks to see if the slot is empty.

Step 2: If the slot is empty and if the station has a packet to transmit, the station will transmit the packet in this slot with probability p_i. If no transmission occurs, the packet is saved, and the station goes back to Step 1. If the slot is not empty, or if the station does not have a packet to transmit, the station goes straight back to step 1.

Each attached station i has its own transmission probability p_i which is chosen to satisfy some desired performance criterion on the network. Given a network such the one in Fig. 1(b), it is clearly apparent that there is an inherent unfairness in the system if p_i is not chosen carefully. For example, if all of the p_i's were chosen equal to 1, station 1 would receive the best service from the network since it can potentially transmit in every slot. On the other hand, station N is clearly at the greatest disadvantage, since it has access to only those slots remaining empty after all of the other stations have attempted transmission. To create a fair system, the p_i's can be chosen to impose equal average packet delay for all stations on the network. The delay is the time from a packet's arrival at a station's queue under it has completed transmission.

Approximate formulas for satisfying this fairness criteria were derived in [10,11]. They compute the p_i's based solely on the average arrival rate λ_i (in packets/slot) at each station. To achieve equal average packet delay, the p_i's should be chosen such that

$$p_i = \frac{2(1-\lambda) + \lambda_i(1+\lambda-\lambda_N)}{(2-\lambda_N)\left(1-\sum_{j=1}^{i-1}\lambda_j\right)}, \quad (1)$$

where $i = 1, 2, \ldots, N$, and $\lambda = \sum_{j=1}^{N}\lambda_j$ is the total offered traffic to the network in packets/slot.

2.2. Dynamic p_i-persistent protocol

In a real network the traffic loads will be varying, so a fixed setting of p_i at each station will not yield the best results. Consequently, it is necessary to dynamically alter the p_i's in response to changing traffic conditions. In a distributed system such as the one being considered here, it is not possible for each station to precisely know the arrival rates at all of the other stations. However, since each station can sense all the traffic on the outbound channel, it can approximate these rates by estimating the traffic from each station on the inbound channel as a percentage of the total traffic seen on the bus.

The inventors of the p_i-persistent protocol proposed a novel algorithm to enable stations to periodically update their p_i values using the above equation (1) based on the traffic seen at the receiver of each station [10,11]. The algorithm includes two design parameters, τ and K, which are used as follows. Each station i updates its p_i synchronously after every τ units of time, where τ equals an integer multiple of a slot duration. Each such update utilizes estimates of the current λ_j, based on the traffic observed by station i over its most recent time window, of length $\eta = K\tau$. Specifically, at each instant of time $n\tau$ ($n = 0, 1, \ldots, K-1, K+1, K+2, \ldots$), Station i ($i = 1, 2, \ldots, N$) executes the following steps:

Step 1: Compute for $j = 1, 2, \ldots, N$:
η_j = number of packets transmitted by Station j during the time interval $\eta = [(n-K)^{*}\tau, n\tau]$, where $(n-K)^{*} = \max(n-K, 0)$; and
$\Lambda_j = \eta_j/\eta$, the estimate of the current value of λ_j.

Step 2: Compute $\Lambda = \sum_{j=1}^{N}\Lambda_j$, the estimated current total offered load to the network.

Step 3: Compute p_i by substituting the Λ_j and Λ for the λ_j and λ in equation (1).

This algorithm provides a means for each station to dynamically learn the status of the network traffic and adapt to it so as to preserve the desired form of fairness. It was shown through series of experiments that this dynamically controlled p_i-persistent protocol is an effective means of access control for a unidirectional broadcast bus network [10]. The stations execute the algorithm to quickly adjust their p_i's to the proper levels to compensate for traffic changes or sudden increases in queue length. Moreover, the algorithm is relatively simple and is executed locally and independently at each station, so that access control is fully distributed.

However, as pointed out by the inventors of this algorithm, this dynamic control algorithm has two drawbacks [10,11]. First, since this algorithm is required to monitor the traffic of each station in the network in addition to the whole traffic of the network, each station must have $N+1$ counters, one counter to keep track of the traffic of the whole network and N counters to keep track of the traffic generated by the N stations attached to the network. As a result, the station's hardware can be costly especially when a very large number of stations are attached to the network. Further, upon the addition of each new station(s), new counters have to be inserted into the stations' hardware which can be very unpractical. Second, updating the p_i's using this dynamic algorithm can take a considerable amount of processing time. This, in turn, can have a big effect on the behavior of the protocol.

In the next section, we will present a *variant* of this novel dynamic control algorithm that requires only 4 counters per station to continuously adjusts the stations' probabilities p_i at their proper levels as governed by the offered traffic no matter how many stations are attached to the network. Moreover, we get *exactly* the same results as that of the original dynamic control algorithm. Also, the processing time for updating the p_i using our algorithm is a lot less than that of the original dynamic control algorithms.

3. Hardware-efficient dynamic p_i-persistent protocol

In this section, we present a *variant* algorithm of the dynamic controlled p_i-persistent algorithm presented in the previous Section. This new algorithm requires only 4 counters per station no matter how many stations are attached to the network while getting *exactly* the same results as that of the original dynamic control algorithm.

Let us recall Eq. (1) which is used to satisfy the fairness criteria in a folded bus network to make our explanation more clear:

$$p_i = \frac{2(1-\lambda) + \lambda_i(1+\lambda-\lambda_N)}{(2-\lambda_N)\left(1-\sum_{j=1}^{i-1}\lambda_j\right)}.$$

According to the p_i-persistent protocol, in order to calculate the value of p_i, we need to know all λ_j where $j = 1, 2, 3, \ldots, N-1$. However, by examining closely the above equation, we do not need to estimate the values of all λ_j. There are four values that are needed by the above equation in order to calculate p_i, which are:

(1) λ,
(2) λ_i,
(3) λ_N, and
(4) $\sum_{j=1}^{i-1}\lambda_j$.

If a station i knows all these 4 values, it can calculate p_i according to equation (1). Consequently, we need only to estimate 4 values by observing the traffic on the network. In order to do this, each station i needs 4 counters at its interface with the network. Denote these counters by C, C_i, C_N, and C_Σ. Each of these counters is used to estimate one of the values which are all needed by Eq. (1) to find p_i as follows:

(1) C is used to estimate λ. Consequently, this counter is increased by 1 if the receiver R (see Fig. 1(b)) of a station senses a nonempty slot.
(2) C_i is used to estimate λ_i. Consequently, this counter is increased by 1 if a station i generates a new packet.
(3) C_N is used to estimate λ_N. Consequently, this counter is increased by 1 if the receiver R of a station senses a nonempty slot sent by station N.
(4) C_Σ is used to estimate $\sum_{j=1}^{i-1}\lambda_j$. Consequently, this counter is increased by 1 if the tap S (see Fig. 1(b)) of a station senses a nonempty slot.

Like the original dynamic algorithm, our *variant* algorithm includes two design parameters, τ and K, which are used as follows. Each station i updates its p_i synchronously after every τ units of time, where τ equals an integer multiple of a slot duration. Each such update utilizes estimates of the current λ, λ_i, λ_N, and $\sum_{j=1}^{i-1}\lambda_j$ based on the traffic observed by Station i over its most recent time window, of length $\eta = K\tau$. Specifically, at each instant of time $n\tau$ ($n = 0, 1, \ldots, K-1, K+1, K+2, \ldots$), Station i ($i = 1, 2, \ldots, N$) executes the following steps:

Step 1: Compute

η_i = number of packets transmitted by Station i (as indicated by the counter C_i) during the time interval $\eta = [(n-K)^*\tau, n\tau]$, where $(n-K)^* = \max(n-K, 0)$; and

η_N = number of packets transmitted by station N (as indicated by the counter C_N) during the time interval $\eta = [(n-K)^*\tau, n\tau]$, where $(n-K)^* = \max(n-K, 0)$; and

η_Σ = number of packets transmitted by upstream stations (as indicated by the counter C_Σ) during the time interval $\eta = [(n-K)^*\tau, n\tau]$, where $(n-K)^* = \max(n-K, 0)$; and

η_T = number of packets transmitted by all stations (as indicated by the counter C) during the time interval $\eta = [(n-K)^*\tau, n\tau]$, where $(n-K)^* = \max(n-K, 0)$; and

$\Lambda_i = \eta_i/\eta$, the estimate of the current value of λ_i.

$\Lambda_N = \eta_N/\eta$, the estimate of the current value of λ_N.

$\Lambda_\Sigma = \eta_\Sigma/\eta$, the estimate of the current total offered load of upstream stations, λ_Σ.

$\Lambda = \eta_T/\eta$, the estimate of the current tal offered load to the network, λ.

Step 2: Compute p_i by substituting the Λ_i for λ, Λ_N for λ_N, Λ_Σ for $\sum_{j=1}^{i-1}\lambda_j$ and Λ for λ in equation (1).

Now, a point-by-point comparison of this algorithm with the original dynamic algorithm is in order:

- Both algorithms yield *exactly* the same results.
- The original dynamic algorithm requires $N+1$ counters per station for an N-station network, while our variant dynamic algorithm requires a

constant of 4 counters per station no matter how many stations are attached to the network.

- Upon the addition of each new station(s), new counters have to be inserted into the stations' hardware which can be very unpractical if the original dynamic algorithm is being used. Using our new algorithm, no new counter may be added or deleted.
- Unlike our algorithm, updating the p_i using the original dynamic algorithm can take a considerable amount of processing time. This, in turn, can have a big effect on the behavior of the protocol.

4. Conclusion

We have presented a *variant* of the original dynamic algorithm for the p_i-persistent protocol. Both algorithms yield *exactly* the same results. However, our new algorithm has the advantage that it requires only 4 counters per station as compared to the requirement of $N+1$ counters per station for the original dynamic algorithm for an N-station network. This results in a considerable savings in hardware cost for each station interface to the network. Moreover, our new algorithm requires a lot less processing time for the updating of the p_i's. This, in turn, can have a considerable impact on the behavior of the protocol. Consequently, using our new dynamic control algorithm, the p_i-persistent protocol seem to be an ideal protocol for high-speed MANs.

References

[1] I.M. Leslie, D.R. McAuley and D.L. Tennenhouse, ATM everywhere?, *IEEE Network* (1993) 40–46.

[2] I. Cidon, J. Derby, I Gopal and B. Kadaba, A critique of ATM from a data communications perspective, *J. High Speed Networks* (1992) 315–336.

[3] M.A. Marsan, C. Casetti and G. Panizzardi, On the performance of topologies and access protocols for high-speed LANs and MANs, *Computer Networks ISDN Systems* 26 (6–8) (1994) 873–893.

[4] CCITT Study Group XVIII, Rep. 34, B-ISDN service aspects, Draft Recommendation, I.211, 1990.

[5] Z.L. Budrikis, Working Document for IEEE 802.6 Services Standard, IEEE 802.6 MAN Working Group, IEEE 802.6-92/, 1992.

[6] IEEE Standard 802.6-1990, Distributed queue dual bus (DQDB) subnetwork of a metropolitan area network (MAN), *IEEE Publications* (July, 1991).

[7] B. Mukherjee and J.S. Meditch, The p_i-persistent protocol for unidirectional broadcast bus networks, *IEEE Trans. Commun.* 36 (12) (1988) 1277–1286.

[8] J.O. Limb, Load-controlled scheduling of traffic on high-speed metropolitan area networks, *IEEE Trans. Commun.* 37 (1989) 1144–1150.

[9] M.M. Nassehi, CRMA: An access scheme for high-speed LAN and MANs, in: *Proc. IEEE Internat. Conf. on Communications* (1990) 115–121.

[10] B Mukerjee, A.C. Lantz, N.S. Matloff and S. Banerjee, Dynamic control and accuracy of the p_i-persistent protocol using channel feedback, *IEEE Trans. Commun.* 39 (6) (1991) 887–898.

[11] B. Mukherjee, Algorithm for the p_i-persistent protocol for high-speed fibre optic networks, *Comput. Commun.* 13 (7) (1990) 387–398.

[12] B. Mukherjee and J.S. Meditch, Integrating voice with the p_i-persistent protocol for unidirectional broadcast bus networks, *IEEE Trans. Commun.* 36 (12) (1988) 1287–1295.

[13] G.S. Miller and M. Paterakis, A Dynamic bandwidth-allocation-based priority mechanism for the p_i-persistent protocol for MANs, *IEEE J. Selected Areas Commun.* 11 (8) (1993) 1229–1239.

Mounir Hamdi received the B.Sc. degree with distinction in Electrical Engineering from the University of Southwestern Louisiana in 1985, and M.Sc. and Ph.D. degrees in Electrical Engineering from the University of Pittsburgh in 1987 and 1991, respectively. While at the University of Pittsburgh, he was a Research Fellow involved with various research projects on interconnection networks, high-speed communication, parallel algorithms, switching theory, and computer vision. In 1991 he joined the Computer Science Department at Hong Kong University of Science and Technology as an Assitant Professor. His main areas of research are High-Speed Networks, ATM Packet Switching Architectures, Wireless networking, and Parallel Computing. Dr. Hamdi has published over 50 papers on these areas in various journals and conference proceedings. He Co-founded and Co-chairs the *International Workshop on High-Speed Network Computing*, is on the editorial board of the *IEEE Communications Magazine*, and has been on the Program Committee of various international conferences. Dr. Hamdi is a member of IEEE and ACM.

Woo Chat Ming received the B.Sc. in Computer Science from the Hong Kong University of Science and Technology in 1995. The research work in this paper was done as part of his final year project thesis. His areas of research are high-speed networking and parallel computing.