ON THE USE OF POPULATION-BASED INCREMENTAL LEARNING IN THE MEDIUM ACCESS CONTROL OF BROADCAST COMMUNICATION SYSTEMS

G.I.Papadimitriou[1], *M.S.Obaidat*[2] *A.S.Pomportsis*[1]

1.Department of Informatics, Aristotle University, Box 888, 54124 Thessaloniki, Greece.
2.Department of Computer Science, Monmouth University, West Long Branch, NJ 07764,U.S.A.
E-mail:{gp, apombo}@csd.auth.gr, obaidat@monmouth.edu

***Abstract*—The Time Division Multiple Access protocol suffers from poor performance when the offered traffic is bursty. In this paper, an adaptive Time Division Multiple Access protocol, which is capable of operating efficiently under bursty traffic conditions, is introduced. According to the proposed protocol, the station which is granted permission to transmit at each time slot is selected by means of a variation of the population-based incremental learning (PBIL) algorithm. The choice probability of the selected station is updated by taking into account the network feedback information. In this way, the proposed protocol is always capable of being adapted to the sharp changes of the station's traffic.**

***Keywords*—Time Division Multiple Access, Bursty Traffic, Population-Based Incremental Learning, Learning Medium Access Control.**

1. INTRODUCTION

An adaptive time division multiple access protocol which is capable of operating efficiently under bursty traffic conditions is introduced. According to the proposed protocol, the station which is granted permission to transmit is determined by means of learning automata [1] that implement a variation of the Population Based Incremental Learning (PBIL) algorithm [2].

At each time slot, the learning automata take into account the network feedback information in order to update the choice probability of the selected station. The probability updating scheme is designed in such a way, that the choice probability of each station asymptotically tends to be proportional to the probability that this station is not idle. In this way, the number of idle slots is minimized and the network performance is significantly improved. When the traffic conditions of a station change, this leads in a change of the choice probability of this station. Therefore, the protocol is capable of being adapted to the sharp load changes of a bursty traffic environment.

The proposed Learning Medium Access Control (LMAC) protocol is applicable to a broad range of broadcast network architectures, including bus, star and wireless LANs. This paper focuses on the theoretical aspects of LMAC rather than on its application to specific network architectures.

The paper is organized as follows: The problem formulation is given in Section 2. The proposed LMAC protocol is presented in Section 3. In Section 4, extensive simulation results are presented which indicate the superiority of the LMAC protocol over other well-known TDMA protocols. Finally, concluding remarks are given in Section 5.

2. PROBLEM FORMULATION

Let $U = \{u_1, \ldots, u_N\}$ be the set of stations, where N is the number of stations. All the stations are connected to a broadcast transmission medium (e.g. a copper wire or an optical coupler). Data are transmitted in the form of packets. All packets are of equal length. The time axis is slotted, with the slot duration being equal to the packet transmission time. Packet transmissions are synchronized with the time slots. Thus, no packet transmission is allowed to start in the middle of a time slot.

Each station is provided with a waiting queue where the packets are temporarily stored while waiting to be transmitted. For all stations, the queue capacity is assumed to be equal to Q packets. When a packet arrives at a station u_i while the waiting queue of this station is full, the packet is discarded. Otherwise, if the queue is not full, it is stored in the queue. It remains there, until it is transmitted.

The traffic, which is offered to the stations, is assumed to be bursty [3]. Packets arrive at the stations in long bursts. Thus, each station can be in one of two states: active or idle. When a station is active, it has one packet arrival at

0-7803-8163-7/03/$17.00

each time slot. On the other hand, when a station is idle, it has no packet arrivals. The mean burst duration is B slots. In order to avoid collisions, only one station is allowed to transmit at each time slot.

In a bursty traffic environment, the bandwidth demands of stations are asymmetric and time-variable. The main challenge in such an environment, is to share the available bandwidth among the stations according to their needs. The key problem is to determine which station is granted permission to transmit at each time slot. In order to reduce the number of idle slots, stations which have packets to transmit must be granted permission to transmit more frequently than other stations. This is not an easy task, because each station operates independently from the others and has no knowledge of their states.

A second problem is to guarantee that - although there is no centralized coordination between the stations - all the stations arrive at the same conclusion on which station is granted permission to transmit at each time slot.

The proposed LMAC protocol copes with the above problems by using a variation of the PBIL algorithm.

3. THE LMAC PROTOCOL

The LMAC protocol is based on a variation of the population-based incremental learning (PBIL) algorithm [2]. The PBIL algorithm is a combination of competitive learning citeLA10 and genetic algorithms [4]. The PBIL algorithm attempts to explicitly maintain statistics about the search space to decide where to sample next. The objective of the algorithm is to create a real valued probability vector, which when sampled reveals high quality solution vectors with high probability. A probability vector which is denoted by $P = \{P[1], \ldots, P[K]\}$ is maintained, with $P[j](j = 1, \ldots, K)$ being the probability of obtaining a "1" in the j-th position. A number of sample vectors are generated according to the probability vector P. Then, the sample vectors are evaluated and the probability vector is updated towards the best sample vector. The algorithmic description of the PBIL algorithm is presented below [2]:

```
Procedure PBIL;
begin
 (* Initialize the probability vector *)
 for j:=1 to K do P[j]:=0.5;
 repeat
  (* Generate samples according to probabilities P *)
  for i:=1 to S do
   for j:=1 to K do
    if RND[j]<P[j] then smpl_vect[i,j] := 1
                   else smpl_vect[i,j] := 0;
  (* Update Probability Vector *)
  for i:=1 to S do eval[i]:=evaluate(smpl_vect[i]);
  best_vect:=find_best_vect(smpl_vect, eval);
  for j:=1 to K do P[j]:=P[j] * (1 - L) + best_vect[j] * L;
 forever;
end;
```

Where:
S is the number of sample vectors generated before update of the probability vector.
L is the learning rate, how fast to exploit the search performed.
K is the number of bits in a generated sample vector.
$RND[j]$ for $j = 1, \ldots, K$, are random numbers which are chosen from the (0,1) interval according to the uniform probability distribution.

The proposed LMAC protocol uses a variation of the PBIL algorithm. The number of bits in a vector is taken to be equal to the number of network stations, thus, K=N. A sample vector schedule is generated according to the probability vector P . Thus, S=1.

The generated sample represents a transmission schedule. The presence of a "1" in the j-th position of the $schedule$ ($schedule[j] = 1$) implies that station u_j is granted permission to transmit during this transmission schedule. On the other hand, the presence of a "0" in the j-th position of $schedule$ ($schedule[j] = 0$) implies that station u_j is not granted permission to transmit during this schedule. After all stations u_j with $schedule[j] = 1$ are granted permission to transmit, the probability vector is updated according to the network feedback information. Then, a new sample vector is selected according to the new probability vector, and so on.

The presented algorithm differs from the PBIL algorithm in that the probability updating scheme is not based on a total evaluation of each schedule, but on a separate evaluation for each element of the schedule. If $schedule[j] = 1$ and u_j has no packets to transmit (idle slot), then $P[j]$ decreases. On the other hand, if $schedule[j] = 1$ and u_j has a packet to transmit (successful transmission), then $P[j]$ increases. Let $u(t)$ be the station which is granted permission to transmit at time slot t and $slot(t)$ be the channel status during this time slot. The following probability updating scheme is used (where: $L, a \in (0, 1)$ and $P[j] \in (a, 1)$):

$$P[j]{:=}P[j]+L(1-P[j]) \text{ if } u(t){=}u_j \text{ and } slot(t)\neq idle \tag{1}$$

$$P[j]{:=}P[j]-L(P[j]-a) \text{ if } u(t){=}u_j \text{ and } slot(t)=idle$$

At each station u_j, the above learning algorithm is implemented by means of a learning automaton. Since the offered traffic is bursty, when the selected station u_j has a packet to transmit, it is probable that this station will have packets to transmit in the near future. Therefore, its choice probability $P[j]$ is increased according to relation (1). On the other hand, when the selected station u_j is idle, it is probable that this station will remain idle in the near future. Therefore, its choice probability $P[j]$ is decreased according to relation (1).

When the choice probability of a station converges to 0, then this station is not granted permission to transmit for a long period. During this period, it is probable that the station transits from idle to busy state. However, since the station is not granted permission to transmit, the protocol is not capable of "sensing" the transition. The role of parameter a, is to prevent the choice probabilities of the stations from taking values in the neighborhood of 0, in order to increase the adaptivity of the protocol.

All the stations use the same probability updating scheme and due to the broadcast nature of the network, the network feedback information is common for all the stations. Consequently, all the stations always contain the same choice probabilities. Furthermore, since the same random number generator and the same seed is used by all the stations, it follows that all the stations select the same station which is granted permission to transmit [6]. Therefore, although there is no centralized coordination between the stations, the protocol is collision-free. The algorithmic description of the LMAC protocol is presented below:

```
Procedure LMAC;
begin
 t := t + 1;
 (* Initialize the probability vector P *)
 for j:=1 to N do P[j]:=0.5;
 repeat
  (* Use vector P to generate a schedule *)
  for j:=1 to N do
   if RND[j]<P[j] then schedule[j]:=1
                  else schedule[j]:=0;
  (* Update the probability vector P *);
  for j:=1 to N do if schedule[j]=1 then
  begin
   t := t + 1;
   u(t) := u_j (* u_j is granted permission for slot t *)
   if slot(t)<>IDLE then P[j] := P[j] + L * (1 - P[j])
                    else P[j] := P[j] - L * (P[j] - a);
  end;
 forever;
end;
```

Let us consider an example in order to clearly demonstrate the operation of the proposed protocol. Assume for example, that N=7 and $P = \{0.7, 0.2, 0.5, 0.1, 0.4, 0.6, 0.2\}$. The transmission schedule is constructed by selecting seven random numbers $RND = \{RND[1], \ldots, RND[7]\}$ from the (0,1) interval according to the uniform probability distribution and comparing them with the corresponding probabilities P. Assume, for example, that: $RND = \{0.1, 0.3, 0.9, 0.2, 0.3, 0.2, 0.5\}$. If RND[j]<P[j] (for j=1,...,7) then station u_j is included in the current transmission schedule. Otherwise, it is not included in the schedule. In our example, the resulting transmission schedule consists of stations u_1, u_5 and u_6. These stations are sequentially granted permission to transmit at the next three slots. Then, the probability vector P is updated according the network feedback information (successful transmission or idle slot) during these slots by using the probability updating scheme (1). Then, a new transmission schedule is generated and the same procedure is repeatedly executed.

4. SIMULATION RESULTS

In this section, the proposed LMAC protocol is compared to two representatives of time division multiple access protocols, namely, the time division multiple access, TDMA [5] and the random time division multiple access, RTDMA [6].

The protocols which are under study are compared by simulation using two different networks (N_1 and N_2) and under bursty traffic conditions. The bursty traffic was modeled in the following way (which is identical to the one presented in [3]): Each node can be in one of two states S_0 and S_1. When a node is in state S_0 then it has no packet arrivals. When a node is in state S_1 then at each time slot it has one packet arrival. Given a station is in state S_0 at time slot t, the probability that this station will transit to state S_1 at the next time slot is P_{01}. The transition probability from state S_1 to state S_0 is P_{10} . Clearly, the mean number of time slots that the station spends in state S_1 is given by $\frac{1}{P_{10}} = B$, where B is the mean burst length (in slots). The periods that each station stays in states S_0 and S_1 are assumed to be geometrically distributed with means of $\frac{1}{P_{01}}$ and B, respectively. The probability for a station to be in state S_1 is given by: $\frac{P_{01}}{P_{01}+P_{10}}$. All the stations are assumed to have the same load. The total load which is offered to the network is R packets/slot. Therefore, each station has a load of $\frac{R}{N}$ packets/slot and consequently, P_{01} can be calculated as follows: $\frac{R}{N} = \frac{P_{01}}{P_{01}+P_{10}} \Leftrightarrow P_{01} = \frac{\frac{R}{N}P_{10}}{1-\frac{R}{N}} = \frac{R}{B(N-R)}$.

The number of stations N, the queue size Q and the mean burst length B, were taken to be as follows:

a) Network $N_1 : N = 10, Q = 10, B = 10$

b) Network $N_2 : N = 4, Q = 2, B = 1000$

The above values of the number of stations and the queue size are realistic for local area networks. The same values have also been used in [7].

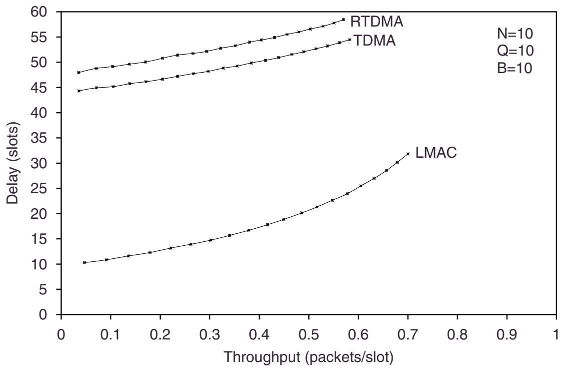

Figure 1: The Delay versus Throughput characteristics of LMAC, TDMA and RTDMA when applied to network N_1.

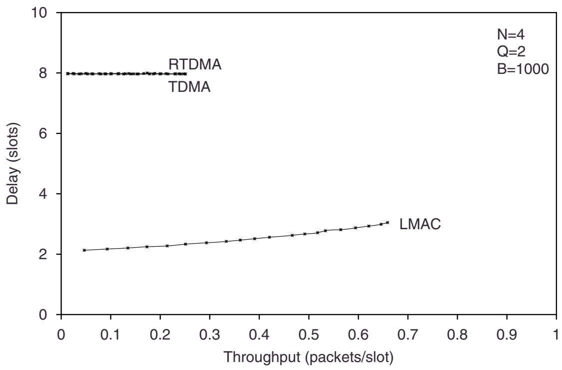

Figure 2: The Delay versus Throughput characteristics of LMAC, TDMA and RTDMA when applied to network N_2.

We have used the delay versus throughput characteristic as a performance metric in order to compare the three protocols The delay versus throughput characteristics of the compared protocols when they are applied to networks N_1 and N_2 are depicted in Figures 1 and 2, respectively.

From the above graphs, it becomes clear that, LMAC achieves a significantly higher delay-throughput performance than the TDMA and RTDMA protocols, when operating under bursty traffic conditions. The performance improvement which is achieved by the use of LMAC is higher when the offered traffic is more bursty.

5. CONCLUSION

This paper has presented a new medium access control protocol for broadcast networks. According to the proposed LMAC protocol, the station, which is granted permission to transmit at each time slot, is selected by means of learning automata, which are capable of being adapted to the changes of the stations' traffic. Therefore, the new protocol is capable of achieving a low delay and a high throughput in the dynamic bursty traffic environment.

The main characteristics of the LMAC protocol are summarized below:

1. It achieves a high performance, even when the offered traffic is bursty.
2. The protocol is self-adaptive and each station tends to take a fraction of the available bandwidth proportional to its needs.
3. No centralized control of the stations is required, since the protocol is fully distributed.
4. Its operation is not affected by a possible node failure.
5. No significant increase in the implementation cost is introduced.

References

[1] G.I.Papadimitriou and A.S.Pomportsis, "Learning-Automata-Based TDMA Protocols for Broadcast Communication Systems," IEEE Communications Letters, Vol. 4, No. 3, pp.107-109, March 2000.

[2] S. Baluja, "Genetic Algorithms and Explicit Search Statistics," Advances in Neural Information Processing Systems 9, (M. C. Mozer, M.I. Jordan and T. Petsche, eds.), pp. 319-25, MIT Press, Cambridge, MA. 1997.

[3] S.L.Danielsen, C.Joergensen, B.Mikkelsen and K.E.Stubkjaer, "Analysis of a WDM Packet Switch with Improved Performance Under Bursty Traffic Conditions Due to Tuneable Wavelength Converters," IEEE Journal of Lightwave Technology, vol.16, no.5, pp.729-735, May 1998.

[4] M.Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, 1998.

[5] I.Rubin and Z.Zang, "Message Delay Analysis of TDMA Schemes Using Slot Assignments," IEEE Transactions on Communications, vol.40, no.4, pp.730-737, April 1992

[6] A.Ganz and Z.Koren, "Performance and Design Evaluation of WDM Stars," IEEE Journal of Lightwave Technology, vol.11, no.2, pp.358-366, February 1993.

[7] B.Mukherjee and J.S.Meditch, "The p_i-Persistent Protocol for Unidirectional Broadcast Bus Networks," IEEE Transactions on Communications, vol.36, no.12, pp.1277-1286, December 1988.