AN ADAPTIVE ALGORITHM FOR HIGHER PERFORMANCE IN A 1-PERSISTENT CSMA/CD LOCAL AREA NETWORK

M. Nazih El-Derini
Department of computer science and Automatic Control
Faculty of Engineering, Alexandria University

ABSTRACT

A simulator is built to examine the performance of a slotted 1-persistent CSMA/CD protocol. Three system parameters; the channel load, the number of users and the packet size, are taken into consideration. The effects of these parameters on four performance measures are examined. These four performance measures are the delay per slot, the channel utilization, the collision utilization, and the best response time to the user which is the primary performance metric in this paper. A new adaptive algorithm is introduced to access a common channel shared among a community of distributed users in a LAN environment. Information about the status of the channel is sensed by all users, and it can be used to improve the performance.

INTRODUCTION

This paper concerns with the performance of the bus architecture LAN under the slotted 1- persistent carrier Sense Multiple Access/Collision Detection (CSMA/CD) protocols [1-8]. Various CSMA protocols have been proposed and evaluated [7]. Some articles had examined the performance of token-bus, token-ring and CSMA/CD with the emphasis on throughput, while other articles had examined the performance of file server and timing problems on local network access unit [9-13].

The goal of this paper is to satisfy the network's user through a fast response time. In this analysis it is assumed that each user has a specific amount of queries or data to send over the channel but in no way he is responsible for the way it is sent, which is the job of the network protocols. The normal protocol divides the user's message into smaller units of fixed size, packets, and let them contend for acquiring the channel independent of the traffic intensity over the channel which results in longer delay and higher percentage of collisions between these contending packets.

An adaptive algorithm based on the feedback concept is proposed in this paper to solve this problem. Analysis starts by collecting some statistics concerning the channel status every period of time. These statistics include the current number of users, the packet size used in transfering data during this period and the percentage of channel bandwidth utilized by colliding packets. The traffic intensity can be determined from this percentage for the current number of users. Then the suitable packet size that gives the minimum average waiting time for that intensity can be chosen by all users in the same manner. The system status is updated every while by repeating this process.

Simulation is used to study a local network model. This model has no bias towards messages based on their length, and it does not cause too much collision. Thus it performs excellently with respect to the collision channel utilization and the normalized average transmitted message length. The performance measures are studied through the results, then the algorithm is explained with some recommendations for implementing it in the existing systems after slight modifications.

SIMULATION MODEL

To measure the performance of the protocol, a simulation model is developed. The simulation model is parameterized by the number of nodes, and the distributions of message arrival and transmission time. The simulation model has been built under the following assumptions.

1. There are N users contending for a common shared channel.
2. Messages are governed by a Poisson process with arrival rate λ, all users have the same distribution.
3. An error-free channel is assumed, where as if two or more messages collide, all messages must be in their entirety.

4. Time is divided into slots and each user synchronizes his packet transmission with the beginning of a slot. The packets, that arrive during the slot, will be held until the next slot.
5. A user can sense the channel and determine its status. The channel would be either "IDLE" which permits the ready user to transmit the packet, "BUSY" which means that the ready user will continuously sense the channel until it is sensed idle, or "COLLISION" which immediately makes the user truncates the remainder of the packet and reschedules it for retransmission. The queuing model is shown in Figure (1).
6. The slot size is equal to two channel end-to-end propagation delay.
7. The collision is detected after Half slot at most and the colliding user uses the other half to send a broadcast jamming packet to inform all users that a collision has taken place.
8. All used data are taken from the Ethernet technical data [14].

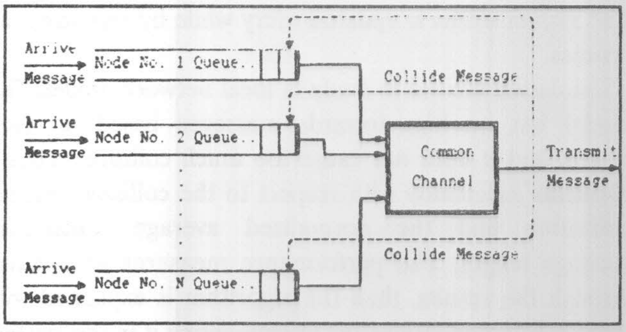

Figure 1. Queuing model.

The main program of the simulator is responsible for inspecting the channel status and determining the occurent events. It calls a routine that inserts a packet into user i's queue if the next event is the arriving of a new packet for user i. In the collision case, the main program calls a routine to reschedule the colliding packets for later retransmission.

A two dimensional array, one row per node, is used to store all nodes' information such as starting and stopping transmission time. Another two dimensional dynamic array, one row per message, is used to store all messages' information such as number of collisions, arrival time and service time. The packets ready to be transmitted on the channel, either fresh or rescheduled for retransmission, are stored in a linked list.

Each user has a data stream or queries that have a given average message length. The data stream is taken long enough to be satisfactory for a simulation period (this data stream varies from 16 MB-320MB). To verify the simulator many experiments are built and confidence intervals for most parameters of the experiments are calculated.Results of these confidence intervals are within satisfactory ranges.

RESULTS

Three system parameters are taken into consideration; the channel load, the number of users, and the packet size. The effects of these parameters on four performance measures are the delay per slot, the channel utilization, the collision utilization, and the average waiting time or the average response time to the user.

The effect of the input load on the performance measures is shown in Figures ,(3),(4) and (5) for 10 users and packet sizes varying from 100 to 2000 bytes. The collision utilization increases with higher rate for higher intensity of the channel traffic. The channel utilization increases as the load increases but with a decreasing rate. The delay per slot increases as the load increases and we have to notice the abrupt increases that happens at point between low and high load for small packet size. The average waiting time increases also as the load increases. This increase is about one order of magnitude when the load increases from 2Mbps to 5Mbps or from 5Mbps to 8Mbps.

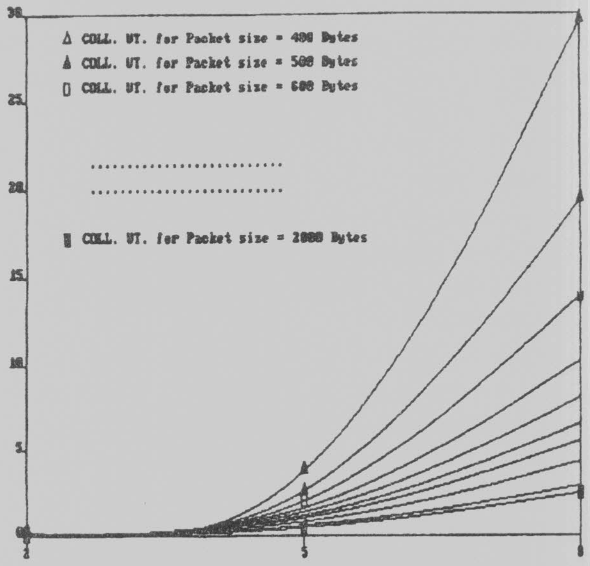

Figure 2. Collision utilization versus input load for 10 users and different packet sizes.

The effect of the number of users on the performance measures is shown in Figures (6), (7), (8) and (9). The collision utilization increases as the number of users increases but this increase is very small for any packet size. The delay per slot, the average waiting time and the response time decrease as the number of users increases. The reason for this behaviour is due to the length of the waiting queues at each node which decreases as the number of users increases. In case of heavy channel load this change will be notifiable.

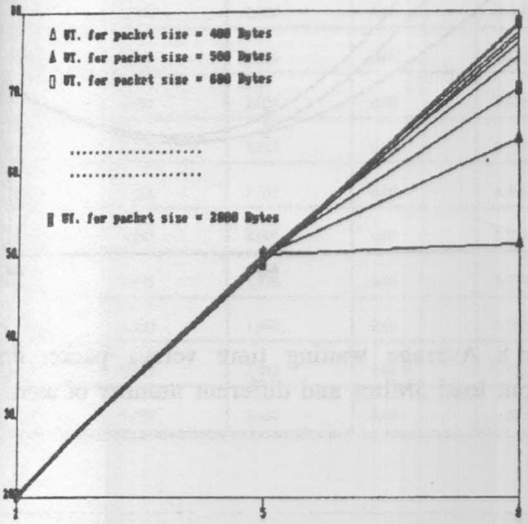

Figure 3. Utilization versus input load for 10 users and different packet sizes.

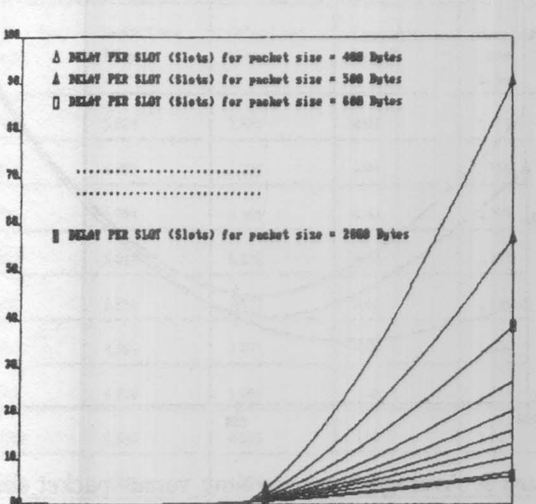

Figure 4. Delay per slot versus input load for 10 users and different packet sizes.

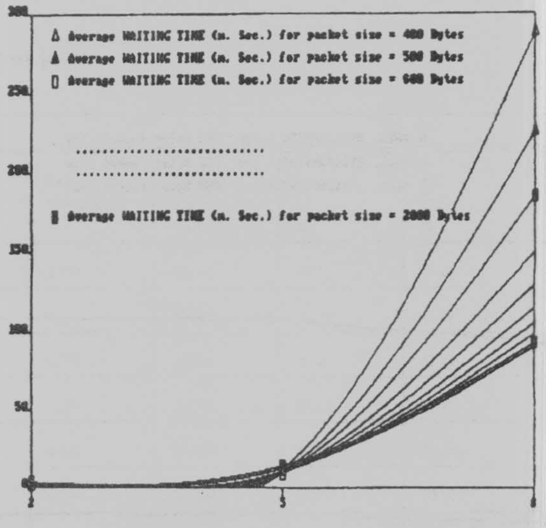

Figure 5. Average waiting time versus input load for 10 users and different packet sizes.

The most important new results are shown in Tables (1), and(3) for 10 users and input load of 2Mbps, 5Mbps and 8Mbps respectively. The effect of varying the packet size on the performance measures is really interesting particularly on the average waiting time that comes in the first place from the user's point of view, as shown in Figures (10),(11) and (12). The source of interest is that the average waiting time has a minimum value at certain packet size. For small packet size there will be many packets which make the collision utilization high and so the average waiting time. When the packet size increases the collision utilization decreases and so the average waiting time, till we reach a packet size which holds the channel more time while arriving packets at other stations will remain in their queues waiting for channel release. Thus this long period of packet size will increase the probability of arriving more than one packet during it and consequently increases the probability of collision and so the average waiting time. So a compromise action must be taken in choosing the packet size to lie at the minimum average waiting time.

ALGORITHM

The previous new result breaks the rule of using fixed length of packets for data transmission. The proposed adaptive algorithm relies on this new result. The algorithm starts by knowing the number of stations currently on line on the network, and for certain channel load, then it chooses the best packet size which corresponds to the minimum average waiting time from the corresponding curve which is similar to Figures(10),(11) and(12).

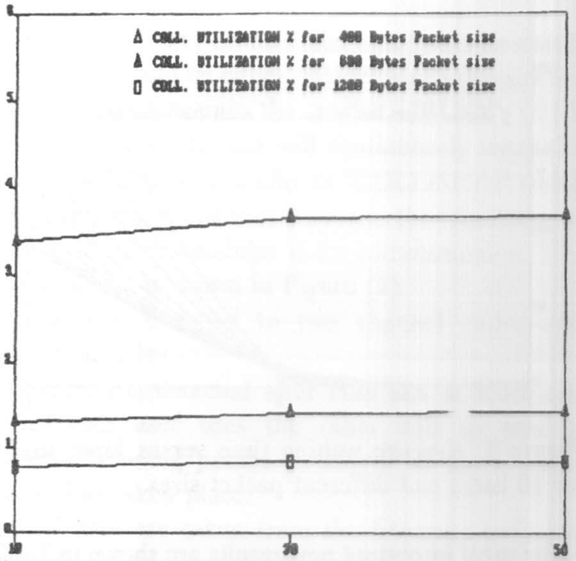

Figure 6. Collision utilization versus number of users for input load 5Mbs and different packet sizes.

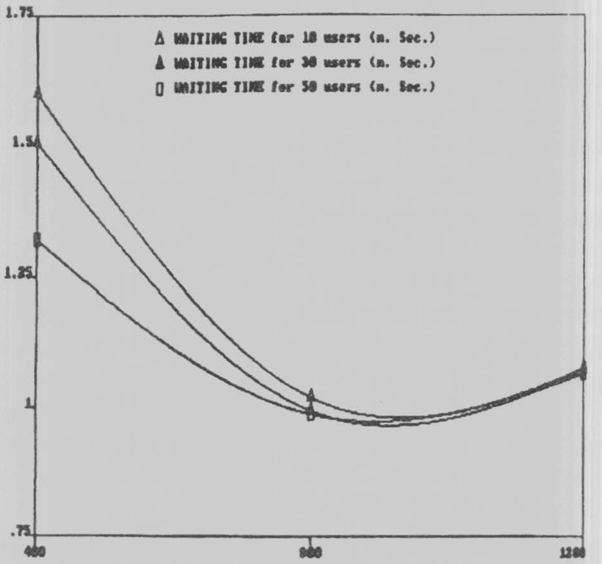

Figure 8. Average waiting time versus packet size for input load 5Mbps and different number of users.

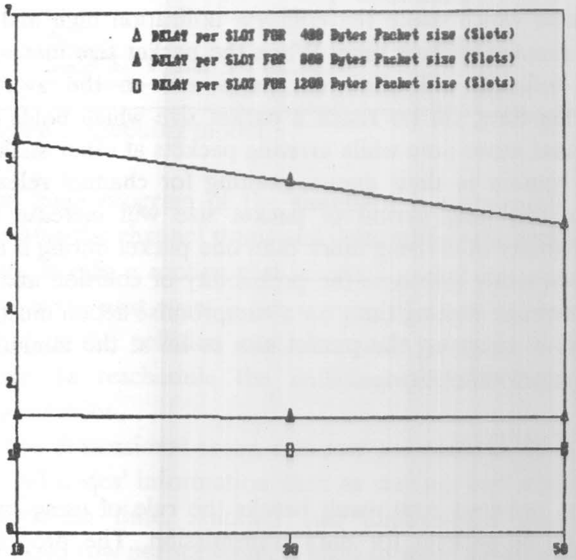

Figure 7. Delay per slot versus number of users for input load 5Mbps and different packet sixes.

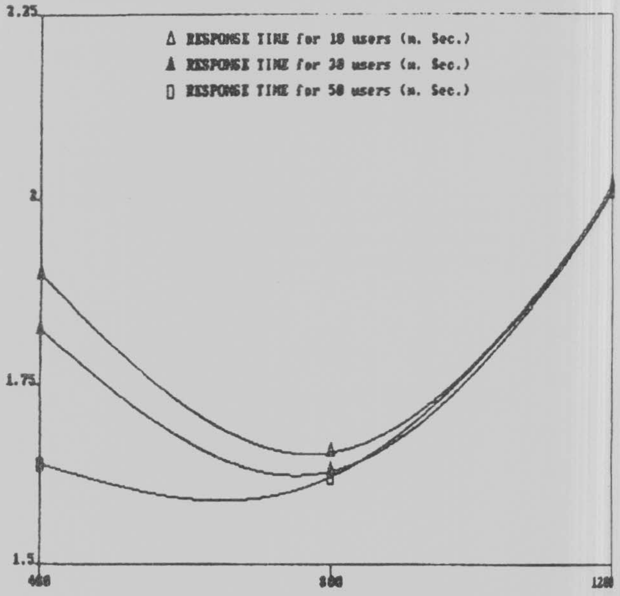

Figure 9. Average response time versus packet size for input load 5Mbps and different number of users.

Table 1. Performance measures for 10 nodes and 2 Mbps input load.

Packet Size Bytes	Input Load Mbps	Offer Load Mbps	Lost /Arr. Msg %	Response Time m.Sec.	Waiting Time m.Sec.	Delay per Slot (Slots)	Carried /Input load %	Tx. Ut. %	Eff. Ut. %	Coll. Ut. %
100.0	2.032	2.255	0.00	0.275	0.195	2.434	100.00	20.32	15.03	3.05
199.7	2.030	2.087	0.00	0.281	0.121	0.757	100.00	20.30	17.66	0.81
299.5	2.040	2.068	0.00	0.360	0.120	0.502	100.00	20.40	18.63	0.41
399.5	2.012	2.030	0.00	0.444	0.125	0.391	100.00	20.12	18.81	0.25
498.2	1.993	2.005	00.00	0.532	0.134	0.335	100.00	19.93	18.89	0.18
599.1	2.015	2.024	0.00	0.625	0.146	0.304	100.00	20.15	19.27	0.14
697.0	2.009	2.017	0.00	0.717	0.159	0.285	100.00	20.09	19.34	0.11
792.0	2.024	2.031	0.00	0.805	0.171	0.271	100.00	20.24	19.58	0.10
892.2	2.003	2.008	0.00	0.895	0.182	0.255	99.95	20.02	19.44	0.08
986.4	1.971	1.976	0.00	0.979	0.190	0.241	100.00	19.71	19.19	0.07
1180.8	1.993	1.997	0.00	1.163	0.218	0.231	100.00	19.93	19.49	0.06
1569.9	2.010	2.013	0.00	1.525	0.269	0.215	100.00	20.10	19.77	0.04
1945.3	2.030	2.032	0.00	1.883	0.326	0.210	100.00	20.30	20.03	0.03

Table 2. Performance measures for 10 nodes and 5 Mbps input load.

Packet Size Bytes	Input Load Mbps	Offer Load Mbps	Lost /Arr. Msg %	Response Time m.Sec.	Waiting Time m.Sec.	Delay per Slot (Slots)	Carried / Input Load %	Tx. Ut. %	Eff. Ut. %	Coll. Ut. %
399.4	5.014	5.325	0.00	1.451	1.132	3.542	99.98	50.13	46.87	3.85
498.0	4.993	5.205	0.00	1.356	0.958	2.404	99.98	49.92	47.31	2.69
600.8	5.024	5.188	0.00	1.393	0.912	1.898	100.00	50.24	48.06	2.09
698.4	5.018	5.150	0.00	1.473	0.914	1.637	100.00	50.18	48.31	1.69
792.2	5.010	5.122	0.00	1.576	0.943	1.487	100.00	50.10	48.45	1.44
892.7	4.981	5.075	0.00	1.666	0.952	1.333	100.00	49.81	48.36	1.22
987.1	4.978	5.061	0.00	1.764	0.975	1.234	100.00	49.78	48.47	1.07
1179.8	4.929	4.993	0.00	1.967	1.023	1.084	99.98	49.28	48.20	0.84
1571.9	4.933	4.979	0.00	2.476	1.219	0.969	99.98	49.32	48.51	0.60
1943.2	4.910	4.946	0.00	2.943	1.389	0.893	100.00	49.10	48.10	0.47

Table 3. Performance measures for 10 nodes and 8 Mbps input load.

Packet Size Bytes	Input Load Mbps	Offer Load Mbps	Lost /Arr. Msg %	Response Time m.Sec.	Waiting Time m.Sec.	Delay per Slot (Slots)	Carried / Input Load %	Tx. Ut. %	Eff. Ut. %	Coll. Ut. %
399.1	8.015	11.891	33.74	29.084	28.765	90.121	64.25	51.50	48.15	29.62
497.2	7.939	10.237	18.39	23.000	22.603	56.843	80.89	64.22	60.85	19.48
599.4	8.005	9.574	11.39	19.089	18.609	38.805	88.13	70.55	67.49	14.09
698.8	7.968	9.056	6.43	15.656	15.096	26.990	93.31	74.35	71.58	10.28
791.4	7.943	8.771	3.98	13.660	13.027	20.583	95.71	76.02	73.53	8.13
893.0	7.933	8.596	2.97	12.275	11.561	16.319	96.84	76.82	74.59	6.63
988.9	7.922	8.477	2.29	11.418	10.627	13.432	97.58	77.30	75.27	5.63
1182.6	7.966	8.395	1.74	10.877	9.931	10.494	98.18	78.21	76.49	4.41
1572.3	7.951	8.236	1.05	10.346	9.089	7.224	98.88	78.62	77.32	3.01
2876.8	7.933	8.072	0.79	12.249	9.948	4.321	99.16	78.66	77.95	1.49
3870.1	8.025	8.130	0.96	14.810	11.714	3.783	98.99	79.44	78.91	1.11

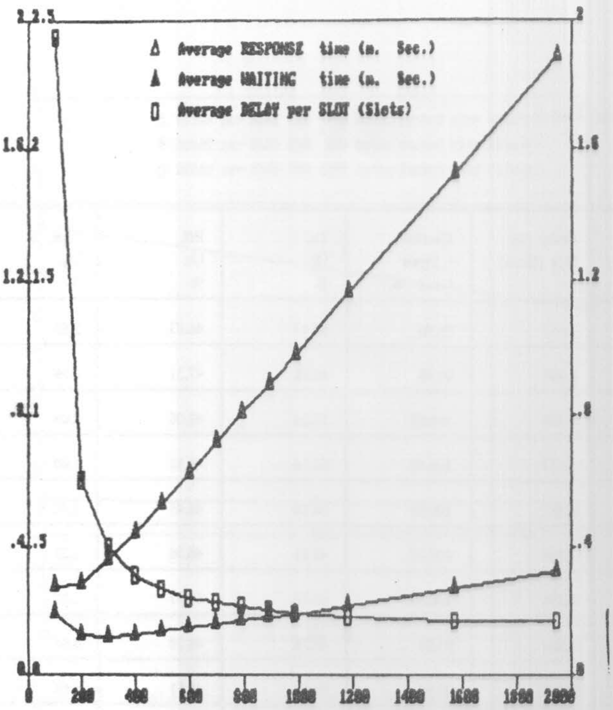

Figure 10. Average response time, average waiting time and average delay per slot versus packet size for 10 users and input load 2Mbps.

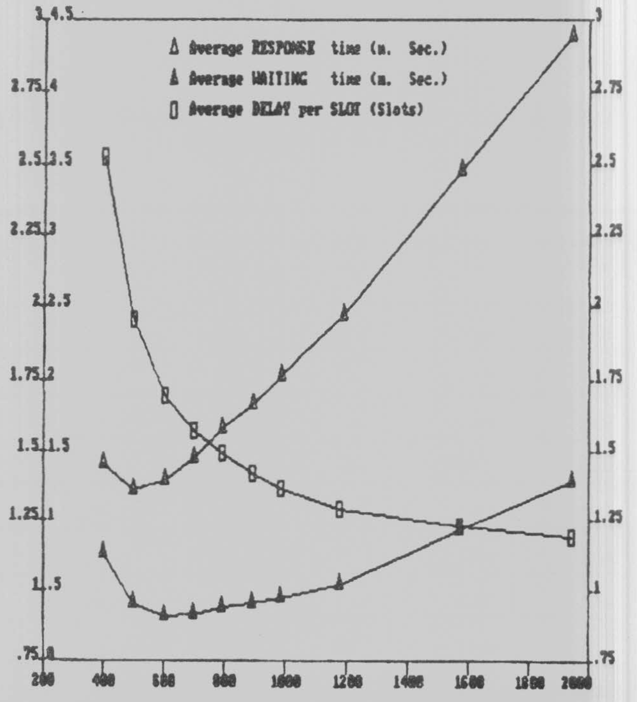

Figure 11. Average response time, average waiting time and average delay per slot versus packet size for 10 users and input load 5Mbps.

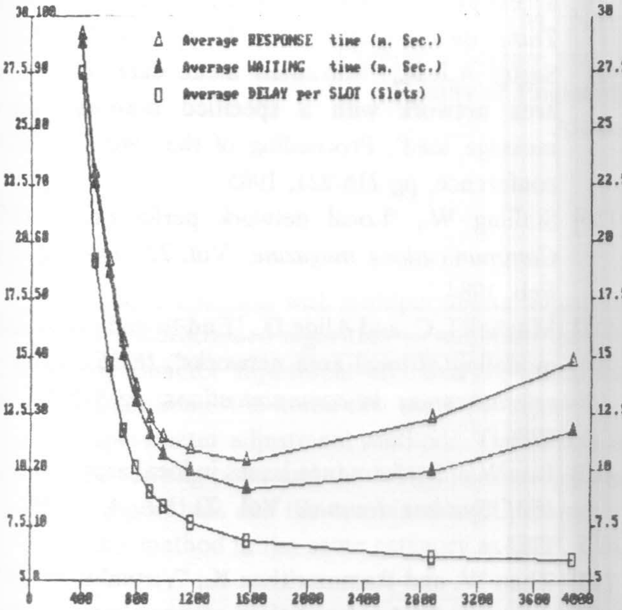

Figure 12. Average response time, average waiting time and average delay per slot versus packet size for 10 users and input load 8Mbps.

A table could be built from these figures for the minimum points for easy of use by using double interpolation with respect to the number of nodes and the load, hence the appropriate packet size is determined. This process is repeated every while, as the channel status changes due to users login and logout or due to traffic rate change, to keep the delay at a minimum.

It is to easy to determine the channel load, from figures similar to Figures-(5) or their corresponding tables, once a station inpects the percentage of channel utilized by unsuccessfully transmitted packets, i.e. collision utilization. The number of collisions within a given period can be detected by each station using a counter incremented by one as it receives the jamming signal. The collision utilization is calculated by multiplying this number of collisions by the collision duration and dividing the result by the simulated clock.

So every time period, this process is repeated. The application of the feedback concept is now clear since the obtained information about the channel status is used to update the packet size parameter and, hence, improving the performance.

To realize this algorithm practically, three points are suggested. First, table are formed for the packet size that gives the minimum delays for each channel load for different number of users. Second, the updating time period will be event driven, i.e. whenever either channel status occurs due to a change in the number of on-line users, or the traffic intensity. Third, the set of tables and curves will be generated either through simulation as it is done in this paper, or from real measurements as far as the system exists.

CONCLUSION

A simulator is built to examine the performance of a slotted 1-presistent CSMA/CD protocol. Three system parameters are taken into consideration; the channel load, the number of users, and the packet size. The collision utilization, the delay per slot, the channel utilization, and the average waiting time increase for higher intensity of channel traffic. Thee effect of changing the number of users on the performance measures is minor for light load, and this effect is notifiable at heavy load. The effect of varying the packet size on the performance measures is really interesting, particularly on the average waiting time measure that comes in the first place from the user's point of view. For certain channel load and number of users, the average waiting time is minimum at an optimal packet size.

A new adaptive algorithm, that relies on the previous result, is introduced in this paper to access a common channel shared among a community of distributed users in a load network environment. For each period of time, each user collects some statistics about the channel status including the current number of users, the packet size during this period, and the percentage of channel utilized by collisions from which he can determine the channel load. In turn the optimal packet size is chosen from some predescribed curves, tables, or equations. This packet size is used in the next period giving the minimum delay per message. Thus the performance of the channel is improved from the user's point of view.

REFERENCES

[1] Kleinrock L. and Lam S., "Packet switching in a multi-access broadcast channels: performance evaluation",*IEEE T rans. on Communications,* COM-23, pg 410-423, 1975.

[2] Hayes J.F., "An adaptive technique for local distribution", *IEEE Tranes. on commuunications,* COM-25, No. 8, pg 1178-1186, Aug. 1978.

[3] Kurose J.F., Schwartz M. and Yemini T., "Multiple access protocols and time-constrained communication", *ACM Computing Surveys,* Vol. 16, pg 40-70, 1984.

[4] Tannenbaum A.,"Network protocols", ACM Computing surveys, Vol.13, pg 453-489, 1981.

[5] Tobagi F., "Multi-access protocols in packet communication systems", *IEEE Trans. on Communications,* COM. 28, pg 468-488, 1980.

[6] Tobagi F. and Hunt V.,"performance analysis of carrier sense multiple access with collision detection protocols", *Comput. Networks,*vol.4, pg 245-260, 1980.

[7] Tobagi F.and Kleinrock L., "Packet switching in radio channels, part I: Carrier sense multiple access model and their throughput delay characteristics", *IEEE Trans. on communications,* COM. 23, pg 1417-1433, No. 12, Dec. 1975.

[8] Towsley D. and Venkatesh D., "Window random access protocols for local computer networks", *IEEE Trans. on Computers,* Vol. 31, pg 715-722, 1982.

[9] Sastry A.R.K., "Maximum mean data rate in local area network with a specified maximum source message load", Proceeding of the 1985 INFOCOM conference, pg 216-221, 1985.

[10] Stalling W., "Local network performance", *IEEE Communications magazine,* Vol. 22, no.2, pg 27-36, Feb. 1984.

[11] Mitchell L.C. and Alide D., "End-to end performance modeling of local area networks", *IEEE Journal on selected areas in communications,* pg 975-985, sep. 1986.

[12] Bux W., " Performance issues in local area networks", *IBM Systems Journal,* Vol. 23, No. 4, pg 351-374, 1984.

[13] Zhao W. and Ramamrithan K., "Virtual time CSMA protocols for hard real time communications", *IEEE Trans. on software engineering,* SE-13, no. 8, pg 938-952, 1987.

[14] Shock J., Data Y. and Redell D., "Evolution of the Ethernet local computer network", *IEEE Trans. on computer,* vol. 15, pg 1-27, 1982.