3-1994

A Topology-Aware Collision Resolution Algorithm

Lewis Barnett III
University of Richmond, lbarnett@richmond.edu

A Topology-Aware Collision Resolution Algorithm

Lewis Barnett
Department of Mathematics and Computer Science
University of Richmond
Richmond, VA 23173
email: barnett@cs.urich.edu

Abstract

A new collision resolution algorithm called the Space Division Multiple Access protocol (SDMA) is presented. SDMA gains a performance advantage over similar protocols by using information about the positions of stations on the network. The protocol can operate asynchronously on a broadcast bus, allowing variable sized packet traffic. Through simulation the protocol is demonstrated to have better performance than Ethernet and the Capetanakis Tree protocol, a similar collision resolution protocol, under some traffic conditions. In particular, under heavy loads, SDMA displays better average throughput and lower variance of delay than Ethernet. The protocol demonstrates a performance bias based on the location of stations, but in most cases this bias is less severe than that experienced by Ethernet.

1 Introduction

Collision Resolution Algorithms (CRAs) are an alternative method for broadcast bus Medium Access Control which have several desirable properties. These algorithms use feedback from the channel to resolve collisions, reducing the average delay and variance of delay experienced by other access mechanisms. Offsetting these desirable properties is the requirement that all stations need complete information on channel history. These algorithms have usually been discussed in terms of slotted transmission channels and fixed-length packet transmission.

The basic method of operation for these protocols can be stated as follows: initially, all ready stations on the network transmit in a slot. If a collision occurs, the stations are divided into two groups according to some criterion. The stations in the "first" group are allowed to transmit in the next slot, and if a collision results, the group is further subdivided and the resolution proceeds recursively. When the first group has been resolved, the second group is allowed to resolve. Several different algorithms have been proposed in the literature based on different criteria for performing the subdivision. The Capetanakis Tree protocols used a uniquely assigned or randomly generated address for each station [2, 3]. The First-Come First-Served protocol uses the arrival time of packets within an enabled transmission "window" as the subdivision criterion [4, 15]. Recently, interest in packetized voice and other time-constrained applications has led to the proposal of variants of the First-Come First-Served protocol which take transmission deadlines into account [11].

Practical application of these techniques is limited by the necessity of synchronous operation of the stations participating in the protocol. The required level of synchronization is difficult to achieve in a bus structured Local Area Network using the Carrier Sense Multiple Access with Collision Detection (CSMA/CD) technique [8]. This question has been investigated by Towsley [14] and Molle [9]. In both cases, asynchronous operation used knowledge of the maximum propagation delay on the network to determine the end of "steps" in the algorithm. This technique can be problematic when counting successive idle steps if there is a significant difference in the clock rates of the participating stations. To avoid this potential problem, Molloy presented a mechanism for signaling the end of idle steps in a Collision Resolution Algorithm by causing an intentional collision which is consistently observable by all stations on the network [10].

Variable length messages were considered by Wu and Chang [16]. Their work describes a protocol where messages are divided into fixed length packets, with the First-Come First-Served algorithm being used to arbitrate control of the bus for transmission of trains of such packets.

The remainder of the paper is organized as follows. Section 2 describes the Space Division Multiple Access protocol, including comments on implementation and limitations. Section 3 discusses the simulation software and the characteristics of the simulated network and workloads. Section 4 describes the performance of the protocol and compares it to that of Ethernet and the Capetanakis Tree protocol. Section 5 summarizes the results.

2 SDMA protocol description

The SDMA protocol is a specialization of the class of protocols which use a unique address to perform collision resolution, such as the Capetanakis Tree protocol. These protocols operate by initially allowing all stations on the network to attempt to transmit without restriction. When a collision occurs between two stations attempting to transmit at the same time, the set of stations are divided into two subsets based on some globally agreed-upon criterion. The typical criterion for subdivision is the address of stations, though other schemes have been proposed [4]. After the subdivision, the resulting subsets are alternately allowed to attempt transmission. If subsequent conflicts occur, the subdivision is repeated recursively until the resulting enabled subset contains only one station with a packet to transmit, at which time that packet is successfully transmitted.

Such protocols do not require any correlation between the address of a station and the location of the station on the network. This scheme simplifies adding new stations to a network; it is only necessary to find a unique address to assign to the new station. However, performance improvements can result from using information about the relative positions of stations on the network in the collision resolution process. The protocol presented here exploits one such improvement. (Various unidirectional broadcast protocols such as Expressnet [13], DQDB [6] and Hymap [12] have made use of upstream/downstream positional information. These protocols are intended for use on folded bus or dual bus topologies.)

The SDMA protocol is similar to the address-based Capetanakis protocol with one important difference; the address space for stations corresponds directly to the physical length of the network. If the address of the station is interpreted as a location (e.g. distance from one end of the cable), then the collision clear time can be reduced with each address space subdivision. To see why this is so, consider the situation when a collision occurs in such a network. Let L be the length of the network. When a collision occurs, the stations are divided into two subsets, one subset containing addresses less than $L/2$ and the other subset containing stations whose address is greater than or equal to $L/2$, first enabling the stations with addresses less than $L/2$ to transmit at the next opportunity. Since addresses correspond to locations, this corresponds to a physical partitioning of the cable, with all the stations in the enabled subset on the same half of the network. This means that the maximum propagation delay between enabled stations will be half of the full propagation delay of the network. So, should another collision occur, the collision itself will be of shorter duration due to the proximity of the colliding stations (each detects the signal of the other and aborts transmission sooner), and the subsequent channel clear time will be half what it was for the previous step.

A method for implementing Collision Resolution Algorithms on an asynchronous (unslotted) broadcast bus was presented by Molloy [10]. The difficulty with implementing most CRAs on an asynchronous bus is that they require the ability to recognize an idle "slot" where no station will attempt transmission. Since there are no slot boundaries on an asynchronous bus, some other mechanism must be employed to signal the end of an idle step in the protocol. The method presented by Molloy requires that when a ready station which collided in the previous interval observes an idle period equal to twice the maximum propagation delay of the network, it should transmit. Since there must be at least two such stations, both observe the idle period and transmit, causing an intentional collision. We refer to a collision of this type as an *idle marking collision.*. This collision is visible to all stations on the network and acts as a signal to proceed with the next step of the algorithm. In networks which conform to the the 802.3 standard in terms of network length and allowed packet sizes, the bandwidth wasted by these intentional collisions is small. The use of positional information in the SDMA protocol further reduces the impact of these collisions, since at each step of the resolution process the time needed for the channel to clear after a collision is halved.

There are several restrictions imposed by the SDMA scheme. As presented, SDMA imposes priorities by location; stations with lower addresses will transmit earlier. Under extreme load, the protocol will result in a round-robin ordering of transmissions. It is possible to alleviate this to some degree by alternating the order in which the subsets are resolved. Stations are required to continuously monitor the channel at all times as in the First-Come, First-Served protocol. SDMA also expects the network to be a single linear cable, a significant topology limitation. The protocol can be adapted to handle branching topologies by mapping addresses to create a logical linear cable with length equal to the sum of the lengths of the branches. On sparsely populated cables, further gains can be made by keeping track of the unused station positions. If an enabled range of addresses consists wholly of unused positions, a step can be skipped.

These limitations restrict the range of applications for which SDMA is appropriate. There are obvious difficulties involved in relocating stations on a network, so the best use of SDMA would be in applications where the locations of stations are fixed. Such applications might involve networks of computing elements, device controllers or intelligent sensors embedded in structures or vehicles.

3 Simulation

SDMA, Ethernet, and the Capetanakis Tree protocol are compared in this paper. The Capetanakis protocol was chosen because it is the CRA which SDMA most closely resembles. The three protocols were simulated using a custom simulation package called Netsim, written by the author. Netsim uses the discrete event simulation method to simulate a finite network of stations. Modeling of collisions is very detailed, taking into account the positions of all stations involved in the collision and the effects of propagation delay on the detection time of the beginning and end of collisions at all stations on the network. This point is particularly important in accurately modeling SDMA, which depends on reducing the length of collision bursts for the performance improvement it achieves over similar algorithms. Netsim also provides very flexible specification of packet length and inter-arrival time distributions, allowing realistic traffic loads to be simulated. Netsim is described in more detail in [1].

The simulation parameters specified a 10 megabit per second contention bus network. There were 32 regularly spaced stations producing identical traffic loads. We consider three packet length distributions: fixed length maximum sized packets (1526 bytes), fixed length minimum sized packets (76 bytes), and a discrete packet distribution abstracted from observations of the University of Richmond campus Ethernet. The distribution is shown in Table 1. The average packet size for experiments run with the packet mixture distribution was 252 bytes. These three packet length distributions represent the best case for overall efficiency (maximum sized packets), the worst case (minimum sized packets) and a realistic intermediate packet length distribution. All experiments used an exponential distribution of packet inter-arrival times. The simulation concentrates on the behavior of the media access mechanism, so no buffering in the stations is modeled.

The Capetanakis Tree protocol results are for an asynchronous version of the protocol constructed using the idle marking collision technique described in Section 2. The simulation is essentially the same as the SDMA simulation, except that the collision duration is not halved with each subdivision of the interval of enabled addresses. This simulates the version of the Capetanakis protocol which uses unique addresses rather than coin flips to generate address bits. Addresses are not restricted to correspond to network locations.

The quantities reported were produced by averaging the results of three runs for each set of parameters. At least 100,000 packets were transmitted in each run.

Bytes	**Percentage of packets**
75	47.5
105	4.4
135	18.0
165	16.3
444	2.6
889	0.8
1065	3.4
1305	0.4
1515	6.6

Table 1: Discrete packet length distribution.

4 Performance of SDMA

It was expected that the Ethernet protocol would prove superior to the SDMA protocol under light traffic conditions, the situation for which the Ethernet protocol was designed. However, it was expected that the collision resolution strategy employed by the SDMA protocol would result in higher overall throughput and lower overall delay under conditions where the network was more heavily loaded. It was also expected that SDMA's performance would be slightly better than an asynchronous implementation of the Capetanakis Tree protocol, the CRA to which it is most similar, in both overall throughput and average delay.

Figure 1 shows the performance of the protocols for a traffic load consisting completely of maximum sized packets. Figure 2 shows performance for the mixture of packet lengths described in Table 1. Figure 3 shows the performance when a homogeneous load of minimum sized packets are generated.

4.1 Average network utilization

Figures 1(a), 2(a) and 3(a) show the average network utilization as a percentage of the total capacity

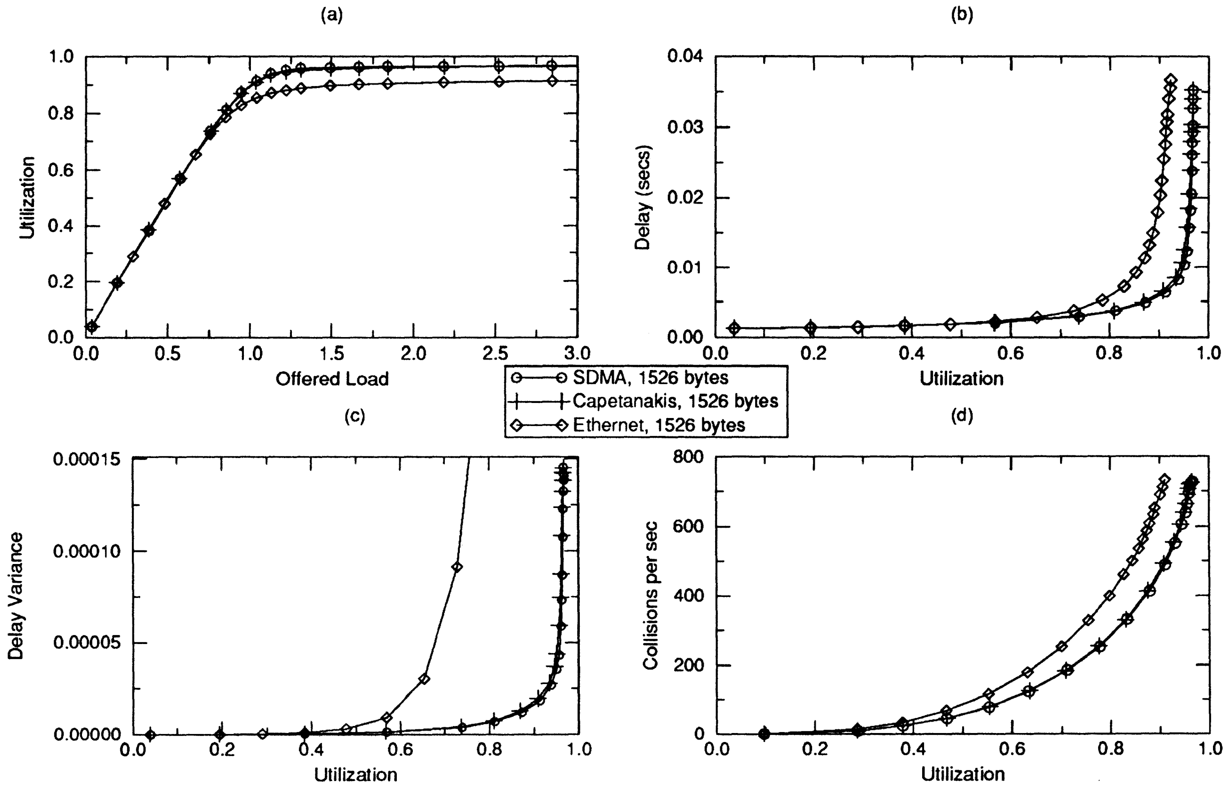

Figure 1: Protocol performance for 1526 byte fixed length packets with exponentially distributed transmission intervals.

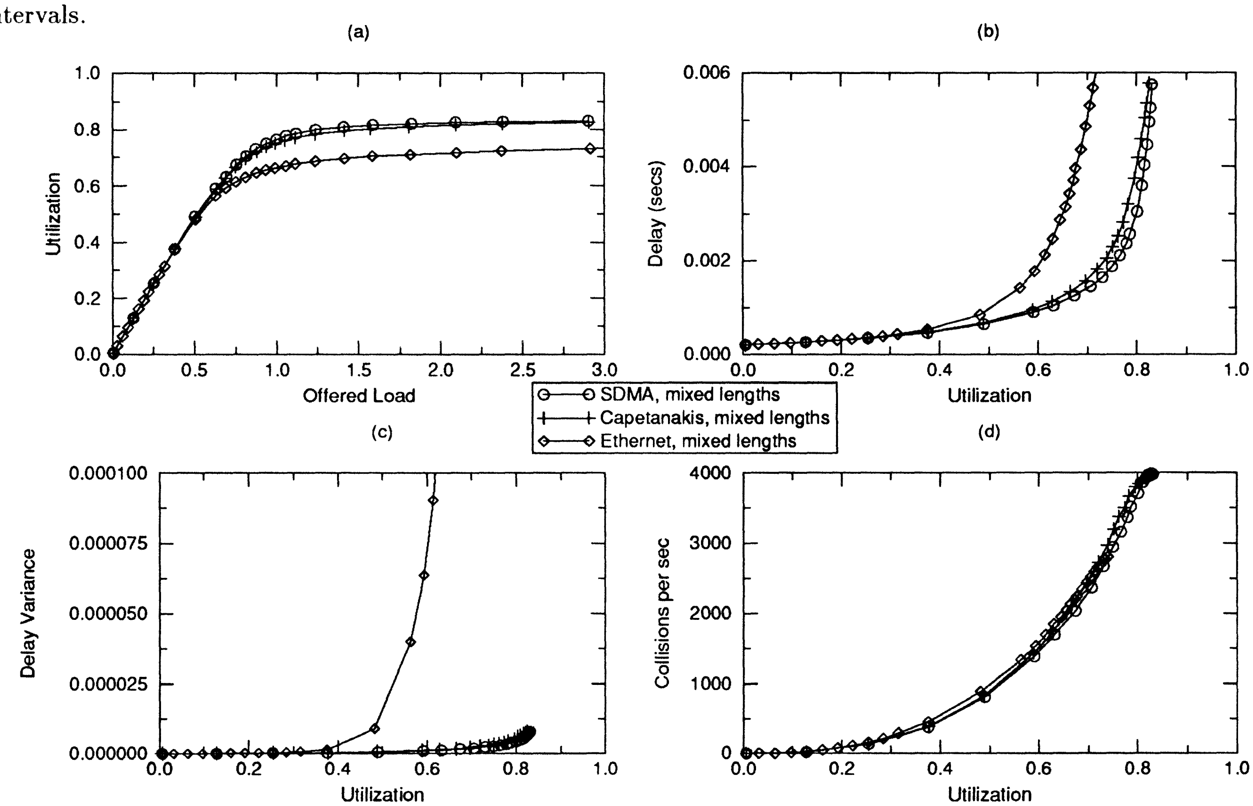

Figure 2: Protocol performance for the packet mixture with exponentially distributed transmission intervals.

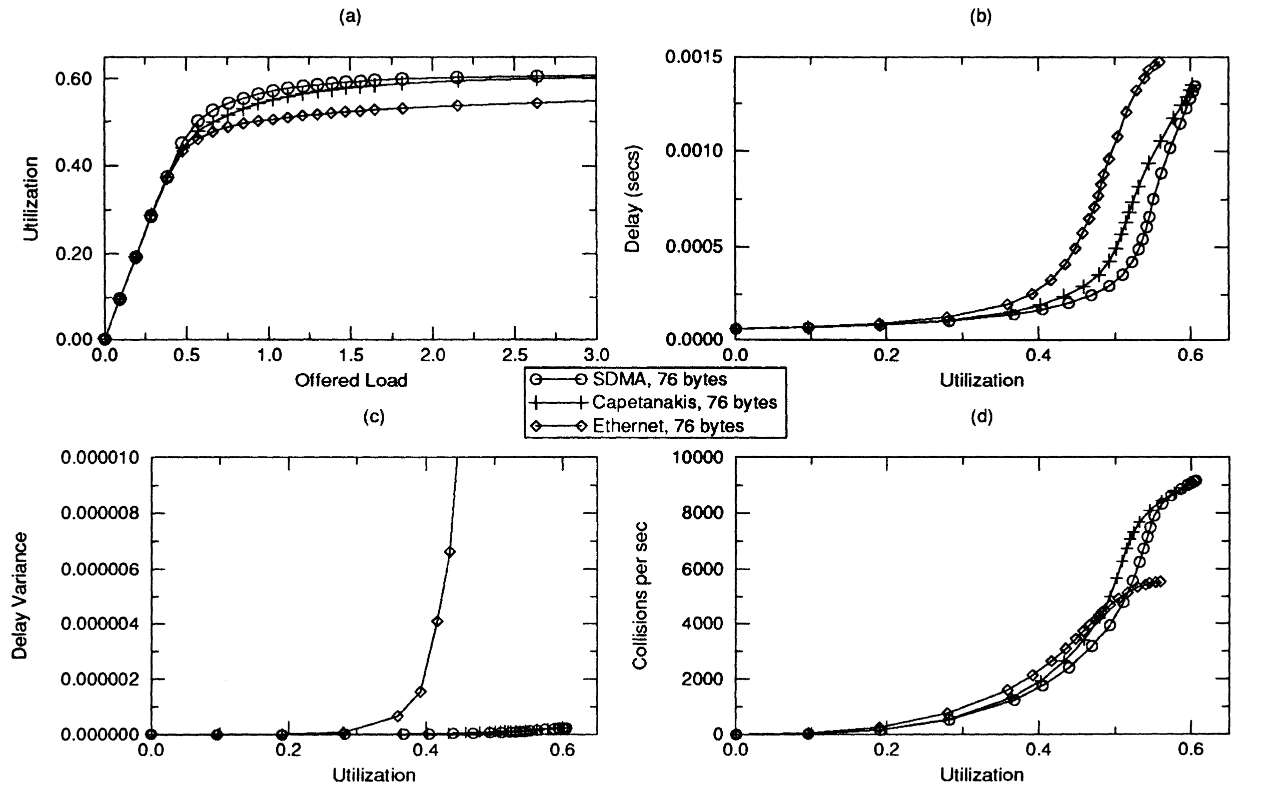

Figure 3: Protocol performance for 76 bytes fixed length packets with exponentially distributed transmission intervals.

of the network versus the offered traffic load, also presented as a percentage of the capacity of the network. Offered load is the sum of the loads presented by each station running on an otherwise idle network and is a measure of the traffic that the stations are attempting to introduce onto the network. The figures show that, contrary to expectations, under light loads (Offered load < 1), SDMA does achieve slightly higher utilization of the network. This situation becomes more pronounced when the offered load approaches and exceeds the capacity of the network. In the experiments using 1526 byte fixed length packets, at an offered load of 200% of capacity, SDMA achieves an overall utilization of 0.963, Capetanakis achieves 0.960, and Ethernet achieves 0.898. The difference between the protocols is even more pronounced in experiments run with the realistic mixture of packet sizes. In those experiments, at an offered load level of 200% of capacity, SDMA achieves a utilization of 0.824, as compared with 0.814 for Capetanakis and 0.717 for Ethernet. With 76 bytes fixed length packets, SDMA achieved a utilization of 0.593, Capetanakis achieved 0.587, and Ethernet achieved 0.538.

4.2 Average delay

Figures 1(b), 2(b) and 3(b) show the average queueing delay for packets in seconds versus average utilization. The 1526 byte packets take 1.2 milliseconds to transmit, the 76 byte packets take approximately 60 microseconds, and the average packet from the packet mixture takes approximately 200 microseconds to transmit. These numbers are reflected in the low load values shown in these figures. The average delay for Ethernet is only slightly higher than the transmission time for the respective packet sizes at low offered load levels. This behavior is matched by both of the CRAs, since they also transmit packets immediately when there is no contention. As the load increases and contention for the channel begins, the two CRAs demonstrate a clear advantage over Ethernet, with SDMA experiencing slightly lower average delay than the Capetanakis protocol. For 1526 byte packets, at a utilization level of 0.85, Ethernet experiences an average delay of 6.8 milliseconds, the Capetanakis protocol experiences average delay of 3.73 milliseconds, and SDMA experiences average delay of 3.65 milliseconds. For the packet mixture, at a utilization level of 0.70, Ethernet's average delay is 5.1 milliseconds, the

Capetanakis protocol's delay is 1.6 milliseconds, and SDMA's delay is 1.4 milliseconds. For 76 byte packets at a utilization level of 0.55, Ethernet's average delay is 1.5 milliseconds, Capetanakis' is 1.0 milliseconds, and SDMA's is 0.7 milliseconds.

The curve for the 76 byte fixed length packets demonstrates an interesting feature of the three protocols. As the offered load increases, at some point all three protocols reach a "plateau" where the delay levels off in spite of the continued traffic increase. For the CRAs, this occurs when all of the stations on the network are continually active, resulting in all stations participating in every collision resolution epoch. In the case of SDMA and the Capetanakis protocol when unique addresses are used, there is an upper bound on the amount of time a resolution epoch can take, based on the number of stations on the network, the maximum packet transmission time, the maximum duration of collisions, and the propagation delay for the network. In the case of Ethernet, the plateau is an artifact of the fact that the protocol gives up after a packet experiences 16 collisions, discarding the packet. So, the leveling off of delay in Ethernet indicates that some packets are being lost, which is not the case for the CRAs.

4.3 Variance of delay

Figures 1(c), 2(c) and 3(c) show the variance of the packet delay versus the network utilization for the three protocols. Lower variance of delay is a well known advantage of CRAs over random backoff collision handling methods, as demonstrated by these figures. At low loads where little contention occurs, there is little difference between the three protocols. As contention begins, the variance of delay for Ethernet increases much more rapidly than for either of the CRAs. Once the traffic becomes heavy the SDMA algorithm resolves collisions among larger numbers of stations more efficiently than Ethernet. When packets are involved in such collisions, the delay they experience is much more regular than Ethernet packets transmitted under similar load conditions due to the unpredictability of Ethernet's Binary Exponential Backoff algorithm. For the 1526 byte packet load, SDMA demonstrates a small advantage over the Capetanakis protocol.

4.4 Collision rate

Figures 1(d), 2(d) and 3(d) show the average collisions per second versus the average utilization. For the CRAs, only "accidental" collisions are counted; the idle marking collisions generated by the algorithms are not included in the count of collisions. For the maximum sized packet load, Ethernet experiences a greater number of collisions per second on average than the CRAs for a given utilization level over the range of loads generated, with SDMA having a slight (but probably insignificant) edge over the Capetanakis protocol. For the two loads generated with smaller packet sizes it is observed that under heavily congested conditions, the collision rate for the two CRAs is actually higher than the rate for Ethernet. This initially seems counterintuitive, given the marked performance advantage the CRAs demonstrate in other performance categories. However, this result is actually a verification of the philosophy adopted by these protocols that more collisions need not be detrimental to performance if they are managed correctly. Under heavily loaded conditions, most packets are experiencing multiple collisions, which results in longer and longer backoff delays for Ethernet, whereas the collisions experienced by the CRAs occur closer together and result in successful transmissions sooner than the Ethernet collisions. For the smaller packet loads, SDMA clearly demonstrates a lower collision rate than Capetanakis for a given utilization level.

4.5 Location bias of SDMA

The primary feature of the SDMA protocol is that it takes advantage of the actual positions of stations on the network to achieve an advantage in overall network performance over similar protocols which do not take advantage of such information. It is therefore not surprising that stations at different locations on the network experience different levels of responsiveness. Similar effects have been observed in other protocols for different types of networks that make use of the location of stations as part of their protocol [7]. Simulation results indicate that the SDMA protocol gives a significant performance advantage to those stations which are considered to have low addresses on the network, i.e. those which are closest to the "left hand" end of the cable – addresses are calculated by distance from this end of the cable. Figure 4 shows the average utilization and delay for each station under a fixed load of 1526 byte packets for both SDMA and Ethernet. Results are shown for offered loads of 0.5, 1.0 and 2.0. The bias is demonstrated most strongly under heavy overloads, and is not evident under the more normal 50% load. As Figure 4(c) and Figure 4(d) show, Ethernet also displays a location bias under the same circumstances, though it is stations at either end of the network which suffer. This effect was noted by Gonsalves and Tobagi [5]. Figures 5 and 6 show that the bias intensifies as the average packet length drops.

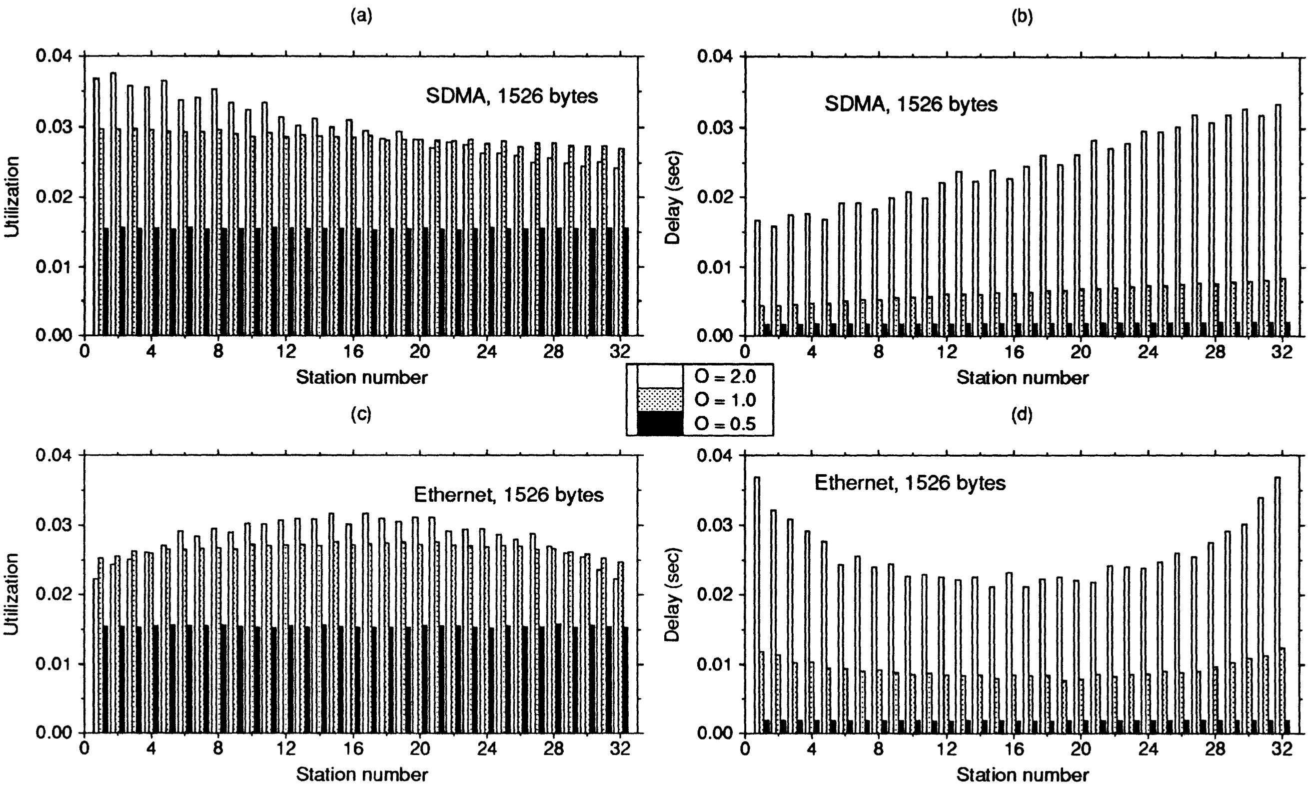

Figure 4: Station utilization and delay for the 1526 byte traffic load. O is offered load.

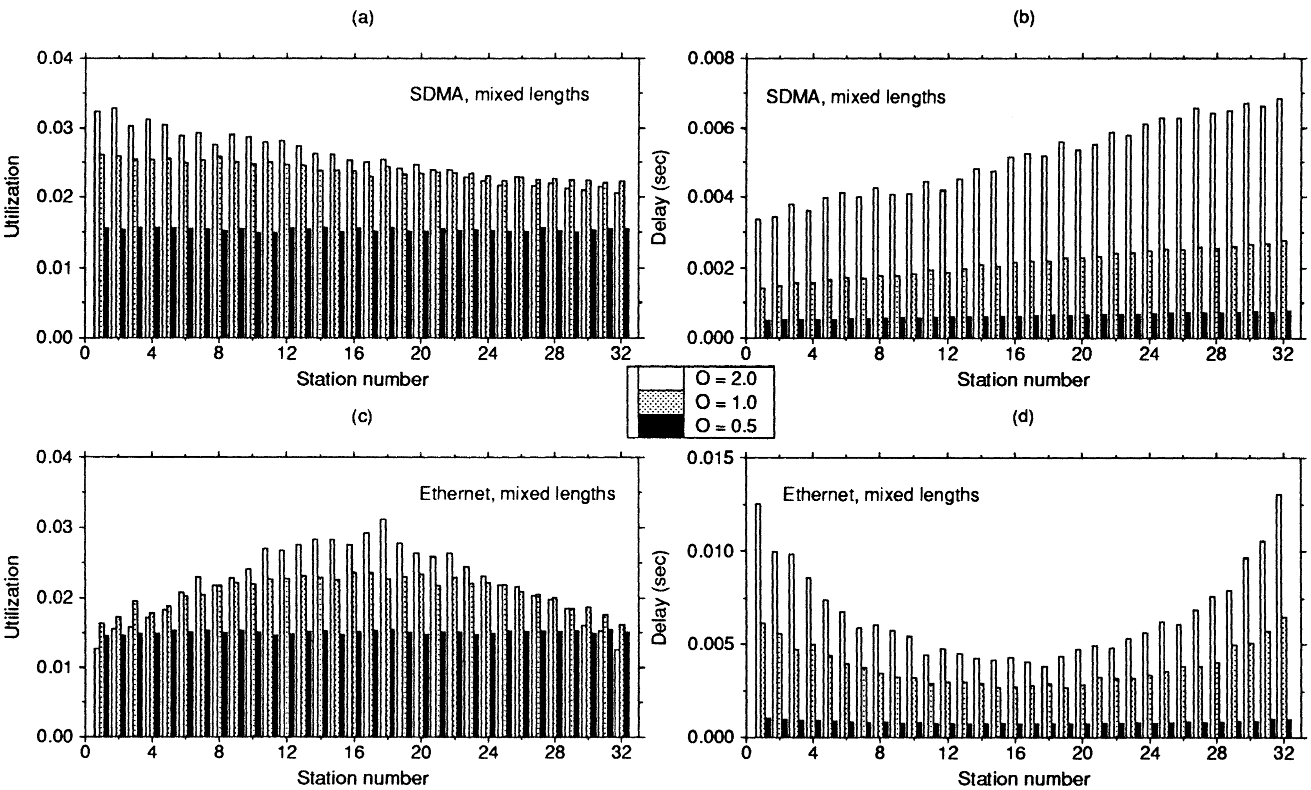

Figure 5: Station utilization and delay for the packet mixture. O is offered load.

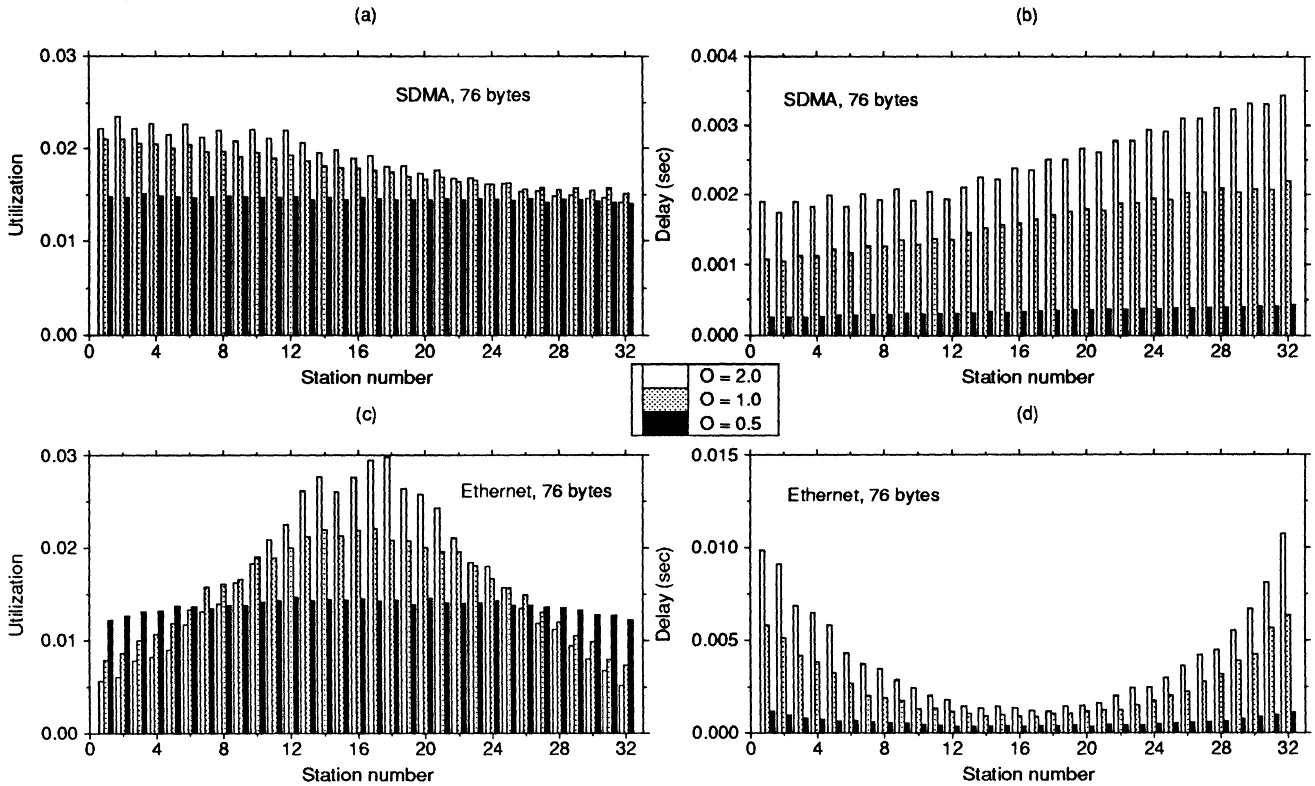

Figure 6: Station utilization and delay for the 76 bytes traffic load. O is offered load.

For the mixture of packet lengths, the bias does not appear significant at the 50% offered load level, but is strongly evident when the offered load is equal to the capacity of the network. For the 76 byte packet load, the bias is evident for both protocols even at the 50% offered load level.

To compare the level of bias in the two protocols, Table 2 presents the average, standard deviation, and "unfairness" of the per station utilization for the various packets lengths and traffic levels considered in this study. The unfairness of a protocol is simply the average difference between an individual station's utilization and the average station utilization given as a percentage of the average utilization. For long packets, the SDMA protocol is less fair than Ethernet except at the lowest load level when both protocols appear to provide reasonably fair access to all stations. However, as the average packet size decreases, Ethernet becomes significantly less fair than SDMA at all offered load levels. For the worst case of 76 byte packets and offered load of 200% of capacity, the standard deviation of average station utilization is approximately two and a half times larger for Ethernet. Even in the more reasonable scenario of mixed packet lengths and offered load equal to network capacity, the Ethernet standard deviation is approximately twice that of SDMA.

5 Conclusions

We have described a new Collision Resolution Algorithm, the Space Division Multiple Access protocol. The protocol's performance was investigated through simulation and compared with the simulated performance of Ethernet and an asynchronous version of the Capetanakis Tree Protocol under identical network conditions. SDMA was found to achieve a higher average utilization and lower average delay under heavy load than either of the other protocols. The protocol also achieved a significant improvement in the variance of delay versus Ethernet. SDMA's performance advantage over Ethernet holds in essentially any situation where contention occurs. The effect of the protocol on the performance of individual stations was also investigated. While there is a location bias in the SDMA protocol, it was found to be less biased by location than the Ethernet protocol run under the same network layout and traffic conditions. We conclude that the location bias, being no worse than that present in a widely deployed protocol, should not be a serious impediment to the use of SDMA.

Protocol	Packet Length	Total Offered Load	Average Station Utilization	Standard Deviation	Average Unfairness
Ethernet	1526	0.5	1.55%	0.0136	0.71%
		1.0	2.67%	0.0743	2.21%
		2.0	2.84%	0.2733	7.81%
SDMA	1526	0.5	1.55%	0.0099	0.48%
		1.0	2.86%	0.0790	2.31%
		2.0	3.02%	0.4109	11.58%
Ethernet	mixed	0.5	1.50%	0.0266	1.44%
		1.0	2.10%	0.2237	9.03%
		2.0	2.24%	0.5112	19.03%
SDMA	mixed	0.5	1.53%	0.0245	1.40%
		1.0	2.39%	0.1250	4.60%
		2.0	2.58%	0.3630	12.01%
Ethernet	76	0.5	1.38%	0.0677	3.87%
		1.0	1.58%	0.4848	26.45%
		2.0	1.68%	0.8008	41.54%
SDMA	76	0.5	1.46%	0.0237	1.28%
		1.0	1.79%	0.1883	9.14%
		2.0	1.89%	0.2971	13.88%

Table 2: Summary of location bias for various packet lengths and network utilization levels.

Acknowledgements

Thanks to Brian Whetten and the anonymous referees for comments on this work.

References

[1] B. L. Barnett, III. An Ethernet Performance Simulator for Undergraduate Networking. In *Proceedings of the ACM SIGCSE Technical Symposium*, pages 145 – 150, Feb. 1993.

[2] J. Capetanakis. Generalized TDMA: The Multi-Accessing Tree Protocol. *IEEE Transactions on Communications*, 27(10):1476 – 1484, Oct. 1979.

[3] J. Capetanakis. Tree algorithms for packet broadcast channels. *IEEE Transactions on Information Theory*, IT-25(5):505 – 515, Sept. 1979.

[4] R. G. Gallager. Conflict resolution in random access broadcast networks. In *Proceedings of the AFOSR Workshop on Communication Theory Applications*, pages 74 – 76, Sept. 1978.

[5] T. A. Gonsalves and F. A. Tobagi. On the Performance Effects of Station Locations and Access Parameters in Ethernet Networks. *IEEE Transactions on Communications*, 36(4):441 – 449, Apr. 1988.

[6] Institute of Electrical and Electronics Engineers. *IEEE Standards for Local and Metropolitan Area Networks: Distributed Queue Dual Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN)*, 1990. IEEE Std 802.6-1990.

[7] H. Kaur and G. Campbell. DQDB – an access delay analysis. In *INFOCOM '90*, pages 630 – 635, 1990.

[8] R. M. Metcalfe and D. Boggs. Ethernet: Distributed packet switching for local networks. *Communications of the ACM*, 19(7), July 1976.

[9] M. Molle. Asynchronous multiple access tree algorithms. In *Proceedings of SIGCOMM 83 Symposium on Communications Architectures and Protocols*, pages 214 – 218. ACM, March 1983.

[10] M. K. Molloy. Collision resolution in an unslotted environment. *Computer Networks*, 9(3):209 – 214, Mar. 1985.

[11] S. S. Panwar, D. Towsley, and Y. Armoni. Collision resolution algorithms for a time-constrained multiaccess channel. *IEEE Transactions on Communications*, 41(7):1023 – 1026, July 1993.

[12] M. Rios and N. D. Georganas. A hybrid multiple-access protocol for data and voice-packet over local area networks. *IEEE Transactions on Computers*, C-34(1):90 – 94, Jan. 1985.

[13] F. A. Tobagi, F. Borgonovo, and L. Fratta. Expressnet: A high-performance integrated services local area network. *IEEE Journal on Selected Areas in Communications*, SAC-1:898 – 912, Nov. 1983.

[14] D. Towsley and G. Venkatesh. Window random access protocols for local computer networks. *IEEE Transactions on Computers*, C-31(8):715 – 722, Aug. 1982.

[15] B. S. Tsybakov and V. A. Mikhailov. Random multiple packet access: Part-and-try algorithm. *Problemy Peredachi Inform.*, 16:65 – 79, Oct. - Dec. 1980.

[16] Y.-T. Wu and J.-F. Chang. Collision resolution for variable-length messages. *IEEE Transactions on Communications*, 41(9):1281 – 1283, Sept. 1993.