Overload Performance of Media Access Protocols for Dual Bus High-Speed Metropolitan Area Networks

Khaled M. F. Elsayed
Department of Communications and Electronics Engineering
Faculty of Engineering, Cairo University, Giza, Egypt 12613
and
The National Telecommunications Institute
Nasr City, Cairo, Egypt
kelsayed@alpha1-eng.cairo.eun.eg

Abstract

In this paper we evaluate some of the media access control (MAC) protocols for high-speed metropolitan area networks based on the dual bus architecture. These MAC protocols are the Distributed Queue Dual Bus (DQDB) family of protocols and the Cyclic Reservation Multiple Access (CRMA). We carry out several simulation experiments to compare the performance of DQDB family of protocols with the CRMA protocol in overload conditions. The results show that DQDB with Access Protection and Priority Control (APPC) provides an efficient and simple method to provide delay and throughput fairness and bandwidth predictability in overload conditions. CRMA provides the fairest access method but it suffers from wasting bandwidth in the process of forming a global queue.

1 Introduction

Local area networks like IEEE 802.3 CSMA/CD, IEEE 802.4 Token Passing Bus and IEEE 802.5 Token Passing Ring operating in the range of 4-16 Mbps are widely used in office and manufacturing applications environment. The success and wide availability of these LANs besides the evolution of Broadband ISDN and the standardization of public fiber optic data networks (e.g. SDH/SONET) operating in the Gbps level necessitated the development of Metropolitan Area Networks to bridge the gap between these networks (e.g. provide an efficient interconnection platform) and to provide integrated services as part of the public network. The basic requirements for a MAN include: providing a shared medium capable of covering an area with a diameter of at least 50 Kilometers, providing transmission bandwidth in the Gbps level, and supporting multimedia traffic services.

To fulfill the above requirements these networks require a media access control (MAC) protocol that can operate efficiently at high speeds and long distances with the following characteristics: providing high throughput and low access delay, flexible in satisfying heterogeneous traffic demands, simple to implement, operate and maintain, and fair with regards to access delay and throughput distribution among access stations. Media access protocols used in local area networks satisfy the above requirement up to their distance coverage and bit rate limitations. The high bit rate and long distances characterizing MANs limit the applicability of LAN MAC protocols. Using LAN MAC protocols in MANs will result in wasting bandwidth and enormous access delays.

In this paper we describe Dual-Bus based MAN architectures and review the details of the DQDB [8] (and its various forms [2, 3]) and the CRMA [7] MAC protocols for MANs. We present a simulation study of the performance of these protocols in overload conditions.

There has been considerable research in the area of performance analysis of MAC Protocols for slotted Dual Bus MANs, of which we report some related work below. Zuckerman and Potter [12] provide a quantitative analysis which shows that a modification in the DQDB draft proposal defined in [4] reduces the unfairness of the upstream stations in favor of the downstream stations. Tran-Gia and Stock [10] presented an approximate performance model based on decomposition of the medium access delay using embedded Markov chains similar to that encountered in the M/G/1 queue. Van As et al. [11] did extensive simulation on DQDB draft proposal [5] under heavy load and overload. Their results showed that this proposal suffered from strong unfairness which increased with increased network bandwidth and size. They also showed that node throughput under heavy load strongly depends on the bus load profile at the instant when heavy loading begins. Moreover, the priority mechanism is not effective when low-priority traffic generating stations lie between the head-end and the higher priority stations. Fdida and Santoso [1] provided similar results using simulation and includes DQDB with bandwidth balancing mechanism. The conclusion is that DQDB with bandwidth balancing offers fair sharing of the medium capacity and predictable performance. However, in [9] it was shown that DQDB with bandwidth balancing fails if more than one priority level exist.

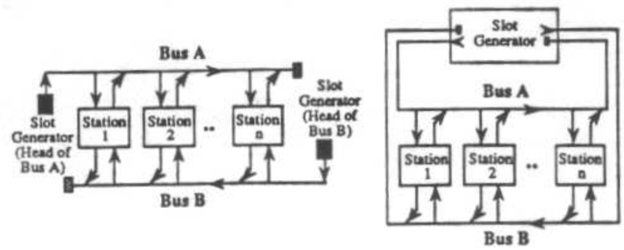

Figure 1: The DQDB Architecture

2 Slotted Bus Access Protocols for High Speed MANs

Bus based MANs can be built in a dual bus or folded configuration as shown in figure 1. The basic components are unidirectional Bus A and Bus B operating in opposite directions. Two head-end stations in the dual bus configuration and one head-end station in the folded bus configuration. Both the forward and backward bus can be used for control signaling and data transfer. A set of access stations are attached to both the forward and backward bus via read/write tabs. A frame is generated each 125 μs. The frame is divided into slots. The head-end A generates fixed size slots periodically. In the dual bus configuration, the end station terminates the forward bus and generate the same slot pattern on the opposite direction.

2.1 The DQDB MAC Protocol

Distributed Queue Dual Bus (DQDB) is being developed as IEEE 802.6 standard. The DQDB uses a dual slotted bus architecture. The bandwidth is divided into fixed length slots of 53 bytes which can be used to carry both isochronous and asynchronous traffic. The stations access their share of the bandwidth using a distributed queueing algorithm [4, 5, 6, 8]. The slot has a header of 5 bytes and payload of 48 bytes. The Access Control Field Byte within the slot header contains the following items: Type: distinguishes between isochronous and asynchronous types of traffic, Busy: indicates whether a slot is empty or occupied, REQUEST: used to inform upstream stations that an additional segment has been added to the distributed queue. There are 4 Request Bits for each level of priority.

DQDB MAC protocol Operation: Each station maintains a Countdown (CD), a Request (RC) and Pending Request (REQ) counter for each bus and priority level. A station can be either in IDLE state or COUNTDOWN state:

IDLE State: The station has an empty queue. The protocol updates the station counters as follows:

- A slot arriving on Bus B with the REQUEST bit set $\Longrightarrow RC = RC + 1$.
- A slot arriving on Bus A with BUSY bit cleared $\Longrightarrow RC = min(0, RC - 1)$

COUNTDOWN State: The station has segments queued for transmission.

- The CD is decremented for each free slot passing on A and RC counter is incremented for each slot with request bit set on Bus B.
- If a slot with request bit cleared arrives on bus B and $REQ > 0$, the station set the REQUEST bit and decrements REQ.

State Transition:

- From IDLE to COUNTDOWN: A segment to be transmitted arrives at the station. The contents of the counters is changed as follows: $CD \leftarrow RC$ and $RC \leftarrow 0$, REQ is incremented.
- From COUNTDOWN to IDLE: When the CD becomes zero, the station starts hunting for the next free slot and transmits a segment as soon as such a slot is available. The station state then becomes IDLE or COUNTDOWN depending on whether the local queue is empty or not.

The above described scheme suffered from unfairness and unpredictability of station throughput and access delay in overload and heavy load conditions. The next two subsections describe mechanisms that was suggested to overcome this shortage.

2.1.1 DQDB with Bandwidth Balancing Mechanism

The proposal in [3] describes a bandwidth balancing mechanism for maintaining a fair allocation of network resources among stations before congestion on any station may occur. Each station maintains another counter called the Collision Avoidance Counter CA-CNTR. The CA-CNTR is used to count the number of segments transmitted by each station. When the counter reaches its maximum value (maximum recommended is 7 or 8), one of the following actions should be taken: If a station has no segment for transmission then increment the RC counter and reset the CA-CNTR; If a station has segments queued then increment CD counter and reset the CA-CNTR. This effectively limits the maximum number of segments a station can transmit.

2.1.2 Access Protection and Priority Control (APPC) Mechanism

In order to counteract the unfairness which occurs in the distributed queueing and have a mechanism that controls amount of bandwidth allocated to each station, Filipiak [2] proposed a set of access protection and priority control as follows.

Upper and Lower Protection Limits: Denote the upper protection limit in the ith station by $\hat{P}$ and the lower protection limit by $\check{P}$. The protection limits are applied when transferring the the contents of the RC counter to the CD counter or when the Countdown process is completed as follows:

- When a packet arrives and $RC > 0$

  Upper Protection : $CD = min(RC, \hat{P})$ and $RC = max(0, RC - \hat{P})$

  Lower Protection : $CD = max(RC, \check{P})$

- When a segment is successfully transmitted set the CD to $\check{P}$ when the lower protection mechanism is applied.

As pointed out in [2], the upper protection mechanism eliminates the stubborn unfairness (all stations trying to capture the total bandwidth at the same time), while the lower protection mechanism relieves the transient unfairness (one node trying to capture all bandwidth for a limited time) at the expense of wasting bandwidth.

2.2 Cyclic Reservation Multiple Access (CRMA) Protocol

CRMA is an access scheme for slotted dual bus networks that achieves high performance even at high speeds and throughput efficiency [7]. Moreover, by using a back pressure mechanism, it can place an upper bound on the worst case access delay for a station.

The header of the CRMA slots contain a reserve command partitioned into three parts such that it can carry the 8-bit cycle number and the 16-bit cycle length. The reserve subcommands and other commands are identified unique opcodes. All commands use a 2-bit priority field. The CRMA access protocol works as follows:

- Head end periodically issues RESERVE commands with cycle length set to zero and new cycle number.

- When RESERVE command passes a station on the outbound bus the station can reserve slots by incrementing the cycle length by the number of slots needed. The station places the cycle number and number of reservations made in a Local Reservation Queue.

- When the reserve command bounces back to the head end, a reservation containing cycle number and cycle length is placed the Global Reservation Queue at the head end.

- The head end serves each reservation by issuing a START command containing the cycle number followed by a number of free slots equal to the number of reservation made by the stations.

- When a START command reaches at a certain station it checks to see if it had previously made a reservation at this cycle. If so it waits for the next available slot and can transmit up to the number of slots it had reserved.

The access protocol is thus highly centralized and is based upon forming a global queue by exploring station states ahead of allowing them to transmit. The frequency of issuing RESERVE commands is crucial to the successful operation of the network. The period between consecutive RESERVE commands should be made less than transmission time of a maximum length packet. However, if reserve commands are issued at high frequency then bandwidth may be wasted. We should also note that CRMA allows for contiguous segment transmission. So, if we have a packet that is divided into many segments all segments will be transmitted consecutively in a single cycle. DQDB family of protocols does not assure this.

The station delay is a function of its position on the network. It is clear that in situations where the network is heavily loaded the upstream stations will experience the worst delay. The delay can be reduced by using inbound bus for RESERVE and the outbound bus for START.

3 Performance of Slotted Bus Access Control Protocols

We built a discrete-event simulation model (using C) for DQDB, DQDB with bandwidth balancing, DQDB with APPC and CRMA. The objective of the simulation is to study the performance of these protocols in overload. The model is composed as follows. The traffic characteristics do not change during the simulation. Packets arrive to stations in accordance with a Poisson process. All stations have infinite buffer capacity. The propagation speed is fixed at 200000 Km/s. The slot size on the bus is 53 bytes and the bus bit rate is 155.5 Mbps (366745.28 slots/s), We assumed that stations process incoming slots in negligible time. The first station assumes the role of the head end A and the last station assumes the rule of the other head end B. All the experiments were done for access on the forward bus and neglecting slots allocated to isochronous traffic. We considered three scenarios for the experiments:

1. All stations request the full network bandwidth.

2. Each station requested bandwidth is smaller than network capacity. But the sum of all stations requested bandwidth is greater than total network capacity.

3. A station in the middle of the bus requests bandwidth significantly higher than other stations and sum of all requested bandwidth is larger than network capacity.

We performed these scenarios for two configuration of the network:

1. A network of 5 stations equally spaced and bus length = 120 Km.
2. A network of 15 stations equally spaced and bus length = 140 Km.

The results obtained for each station include: 1) node throughput in Mslots/s (and confidence interval), 2) access delay in msec (and confidence interval), and 3) mean queue length (and confidence interval) (not shown). In the results we show only throughput and access delay since the delay can be used to estimate the queue length. Each simulation is run for a period of 30000 slots (82.1 msec) and is repeated 10 times, starting with an empty queue at each station. The confidence intervals obtained were very narrow and therefore, we are not showing the confidence intervals in the figures.

Results for the Five Stations Bus
Figures 2(a) and 2(b) shows the results for scenario 1. In this case, CRMA and DQDB-APPC provide similar throughput and access delay for all the stations independent of their position on the bus while DQDB and DQDB-BWB give preference to the upstream stations. Figures 2(c) and 2(d) shows the results for scenario 2. Again, we notice similar response as in scenario 1 but the three upstream stations get similar share of the bandwidth and access delay. In figures 2(e) and 2(f) we show the results for scenario 3. In this case, DQDB-APPC fails to provide fair access to the bus. Station 3 which is requesting nine times more bandwidth than the other station is given less bandwidth and suffers from excessive delays. CRMA still manages to provide fair access for all stations. Similar behavior to scenarios 1 and 2 is observed for DQDB and DQDB-BWB with station 3 getting more bandwidth than the rest of the stations.

Results for the Fifteen Stations Bus
Figures 3(a) and 3(b) shows the results for scenario 1. Similar behavior as in the five stations case is observed. For scenario 2, the behavior was similar to the five stations case also and is not shown. In figures 3(c) and 3(d) we get some interesting results. We note that CRMA provides more bandwidth to station 8 than the rest of the stations. The bandwidth is divided among the access nodes proportional to their requested bandwidth except for station 14 whose share of the throughput is significantly lower than the other stations with similar bandwidth requirements. It is not clear why CRMA behaves like this in this scenario. DQDB-APPC provides equal share of the bandwidth to the stations while station 8 experiences more delay. However, the unfairness is not as strong as in the five stations case. In the DQDB and DQDB-BWB cases, throughput decreases as a function of station number until we reach station 8 where the throughput increases abruptly and then the rest of the stations get equal bandwidth (less than that of station 8). Delay distribution also depends on the station location on the bus with the lowest access delay near the head end.

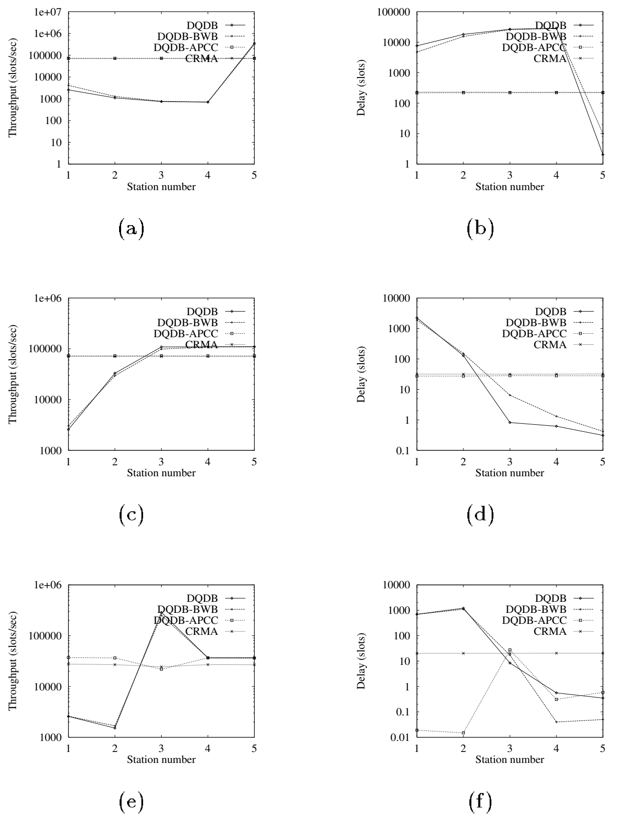

Figure 2: (a),(b) Throughput and access delay of stations under different MAC protocols with each station requesting full bandwidth. (c),(d) Throughput and access delay of stations under different MAC protocols with each station requesting 0.3 full bandwidth. (e), (f) Throughput and access delay of stations under different MAC protocols with each station requesting 0.1 full bandwidth except station 3 which requests 0.9 of full bandwidth

From the above results, we can see that CRMA is the best protocol since it used a global queue for requests. However, it wastes bandwidth compared to DQDB-APPC since some of the available bandwidth is consumed in the process of forming the global queue. DQDB-APPC fails in the case when one station has much more traffic demands than other stations. However, in this situation, the station requesting more bandwidth is allowed more bandwidth but the access delay is higher than the other stations.

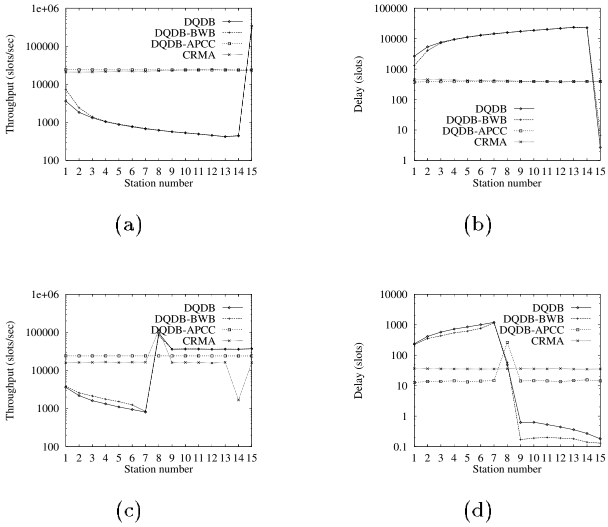

Figure 3: (a), (b) Throughput and access delay of stations under different MAC protocols with each station requesting full bandwidth. (c), (d) Throughput and access delay of station under different MAC protocols with each station requesting 0.1 of full bandwidth, except station 8 which requests 0.7 of total bandwidth

4 Conclusions

The simulation model shows that a simple distributed mechanism like DQDB-APPC is comparable in performance to a more complex protocol like CRMA. DQDB-APPC can provide fair sharing of the medium capacity and predictable performance under heavy load and overload in an independent manner of the initial conditions of the network. We conjecture that the APPC mechanism can be used in multi-priority networks successfully. However, efficient implementation of DQDB-APPC should adjust protection limits adaptively. CRMA provides the best overall performance in exchange of wasting a portion of the bandwidth in forming global queue by using RESERVE and START command. There are many ways in which the work presented here can be extended. The arrival process can be modeled by a distribution or stochastic process which can describe the bursty nature of the traffic. Also a limited buffer size can be imposed on the stations so we can measure the probability of loss of a segment and/or packet. Testing whether using inbound bus in CRMA for reservation will enhance the downstream stations delay with respect to upstream stations. In order for DQDB-APPC to be cost effective, it should be capable of adaptive adjustment of limits, bounds and thresholds to changing load conditions and position of active stations. Control stations can be added to the network to perform maintenance, control, and testing. Alternatively, user stations can serve as control stations if needed. Traffic management would consist of measuring channel utilization, collecting information from other stations and recalculating flow control thresholds.

References

[1] S. Fdida and H. Santoso, Perfomance issues of the DQDB protocol, in *Proc. 4th Int'l Conf. on Data Com. Systems and their Performance*, Barcelona, Spain, June 1990.

[2] J. Filipiak, Access protection for fairness in a distributed queue dual bus Metropolitan Area Network, in *Proc. ICC '89 Conf.*, Boston, MA, June 1989, pp. 635-639.

[3] E. L. Hahne, N. F. Maxemchuk, and A. K. Chaudhury, Improving DQDB fairness, in *Proc. INFOCOM 90*, San Fransisco, CA, June 1990, pp. 175-184.

[4] IEEE 802.6/D6, Draft Proposal Standard Distributed Queue Dual Bus (DQDB) Metropolitan Area Network (MAN), November 1988.

[5] IEEE 802.6/D7, Draft Proposal Standard Distributed Queue Dual Bus (DQDB) Metropolitan Area Network (MAN), May 1989.

[6] IEEE 802.6/D15, Draft Proposal Standard Distributed Queue Dual Bus (DQDB) Metropolitan Area Network (MAN), October 1990.

[7] M. Nassehi, CRMA: An access scheme for high-speed LANs and MANs, in *Proc. ICC 90*, pp. 346.5.1-345.5.6

[8] R. Newman, Z. Budrikis, and J. Hullet, The QPSX MAN, *IEEE Communication Magazine*, vol. 26, April 1988, pp. 20-28.

[9] M. Spratt, A problem in the mutli-priority implementation of the bandwidth balancing mechanism, *Contribution to IEEE 802.6 Working Group IEEE P802.6-89/61*, 1989.

[10] P. Tran-Gia and I. Stock, Approximate performance of the DQDB access protocol, in *Proc. 7th ITC Sem.*, Adelaide, Australia, Sept. 1989, paper 16.1.

[11] H. R. van As, J. W. Wong, and P. Zafirpolo, Fairness, priority, and predictability of the DQDB MAC protocol under heavy load, in *Proc. Int. Zurich Sem.*, Zurich, Switzerland, Mar. 1990, pp. 410-417.

[12] M. Zuckerman and P. Potter, The DQDB protocol and its performance under overload conditions, in *Proc. 7th ITC Sem.*, Adelaide, Australia, Sept. 1989, paper 16.4.