A Highly Adaptive Media Access Protocol for Dual Bus Metropolitan Area Networks *

Jörg Liebeherr

Computer Science Division
University of California - Berkeley
Berkeley, CA 94720
U. S. A.

Ian F. Akyildiz

College of Computing
Georgia Institute of Technology
Atlanta, Georgia 30332
U. S. A.

Abstract

The Distributed Queue Dual Bus protocol, the media access protocol of the current IEEE 802.6 standard for Metropolitan Area Networks, has two major drawbacks. First, it is not able to utilize the full bandwidth. Also, it needs considerable time until a fair distribution of bandwidth to the stations is achieved. In this study a protocol for dual bus networks is proposed which does not show the disadvantages inherent to the IEEE 802.6 standard. A fair distribution of bandwidth is obtained after a time equal to one round-trip delay. Additionally, the protocol allows a full utilization of the available bandwidth. The protocol is derived from a so-called 'fair and waste-free bandwidth allocation' scheme. The features of the new protocol can be included into the the existing IEEE 802.6 standard. By comparing the performance of the newly proposed protocol with the IEEE 802.6 standard (including bandwidth balancing) we show that the new protocol has significant advantages over IEEE 802.6.

Key Words: Computer Networks, DQDB, MAN, Communication Protocol.

1 Introduction

The Distributed Queue Dual Bus (DQDB) Media Access Control (MAC) protocol has been adopted as the IEEE 802.6 Metropolitan Area Network standard [7]. Using fiber optic technology a DQDB network allows data transmission at a rate of several hundred Mbps. DQDB has received considerable attention since it is the designated access protocol for the Switched Multi-Megabit Data Service (SMDS) [2].

Early versions of the DQDB protocol were unable to provide a fair distribution of bandwidth to stations under conditions of heavy network load [15]. The problem was overcome by adding the so-called 'bandwidth balancing' mechanism to the protocol [5]. However, the bandwidth balancing mechanism has two known major drawbacks. First, bandwidth balancing forces the network to leave a certain percentage of the available bandwidth unused. In addition, bandwidth balancing needs considerable time until a fair distribution of bandwidth to the stations is achieved.

In this study, we introduce a protocol for dual bus network. The new protocol does not show the disadvantages inherent to the bandwidth balancing mechanism. A fair distribution of bandwidth is obtained after a time corresponding to one round-trip delay. Additionally, the protocol allows to utilize the full bandwidth of the network. The protocol is highly adaptive to changes in the network load.

Several methods to improve the DQDB protocol have been proposed in recent years. Some propositions attempt to enhance the bandwidth balancing mechanism [12, 14]. Other studies substitute bandwidth balancing by an alternative fairness mechanism [3, 4, 6, 8, 9, 12]. An excellent summary of the literature on DQDB is given in [13].

The paper is organized as follows. In section 2 we briefly overview the DQDB (IEEE 802.6) protocol. In section 3 we formally derive new properties of optimal bandwidth allocation schemes for dual bus networks. In section 4 we use the results from section 3 to define a new protocol for dual bus networks. We refer to the new protocol as $DQDB^{+/-}$. In section 5 we compare the performance of $DQDB^{+/-}$ with the IEEE 802.6 standard. In section 6 we conclude our results.

*This work was supported by the IBM T. J. Watson Research Center, Yorktown Heights, under Thesis Agreement No. 20441066.

2 The DQDB (IEEE 802.6) Media Access Protocol

A dual bus network consists of two unidirectional buses with data flow in opposite directions. One bus is denoted by bus A and the other by bus B. A slot generator at the head of each bus emits empty fixed sized slots at a constant rate. Each station is connected to both buses. A station transmits data by filling in an empty slot. Note that due to the topology of the dual bus each node has to make a routing decision whether to use bus A or bus B for transmission according to the physical location of the destination station. Since the architecture of a dual bus network is symmetric we will focus on data transfer on bus A.

The DQDB [1] protocol prevents the stations close to the head of a bus from acquiring all empty slots by implementing a reservation scheme. A station having a segment ready for transmission (on bus A) notifies the stations closer to the the head of bus A by sending a reservation request on bus B.

Each slot contains two access fields: a busy bit and a request bit. A slot with the busy bit set indicates that the slot contains data. The request bit set indicates a reservation request. If a station writes data into an empty slot it sets the busy bit. A reservation request is submitted by setting the request bit.

Each station keeps a queue of untransmitted segments for data transfer on a particular bus. Only the segment at the head of the queue is allowed to submit a request on the bus. Note that setting the request bit may be delayed since a station has to wait for a slot on bus B which has the request bit not set. A station determines its turn to transmit a segment with two counters: the request counter (RQ) and the countdown counter (CD). If a station is idle, i.e., the station does not have any segments to transmit, it increments RQ for each passing slot on bus B with the request bit set and decrements RQ for each empty slot the station detects on bus A. On arrival of a segment to a station the station becomes active. RQ is copied to CD and then set to zero. Then, RQ resumes increasing its value for each slot on bus B having the request bit set. CD is decremented for each empty slot passing the station on bus A. If CD reaches zero the station takes the next empty slot for transmission.

The DQDB protocol is not able to achieve an equal distribution of the bandwidth among the stations if the network is heavily loaded [15]. The so-called bandwidth balancing mechanism [5] achieves a fair distribution of the bandwidth and was added to the IEEE 802.6 standard. The bandwidth balancing mechanism enforces that each station uses only a fraction of the available bandwidth for transmissions. This is achieved by incrementing the request counter each time after BWB_MOD transmissions of a segment. A DQDB network with bandwidth balancing has two major drawbacks. First, it cannot utilize the entire bandwidth. Secondly, the bandwidth balancing mechanism needs considerable time until a fair distribution of the bandwidth to the stations is reached.

[1] We will use 'DQDB' and 'IEEE 802.6' synonymously.

3 Fair and Waste-Free Bandwidth Allocations

In this section we derive some properties of a bandwidth allocation scheme for dual bus networks. We show the existence of a unique bandwidth allocation scheme which guarantees fairness and utilizes the full bandwidth. We present a distributed version of this scheme. The theoretical results from this section are directly applied in the following sections where we propose a multi access protocol for dual bus networks.

Definition 1 Let $\mathbf{N} = \{1, 2, \ldots, n\}$ be a set of stations. Let λ_i ($\lambda_i \geq 0$) and γ_i ($0 \leq \gamma_i \leq 1$) denote the arrival rate and the throughput of station i ($1 \leq i \leq n$), respectively.

1. A bandwidth allocation is a relation $R = \{(\lambda_i, \gamma_i); 1 \leq i \leq n\}$ such that:
$$\gamma_i \leq \lambda_i \qquad \text{and} \qquad 0 \leq \sum_{i=1}^{n} \gamma_i \leq 1$$
2. A *fair* bandwidth allocation satisfies:
   (a) $(\forall i, j)(1 \leq i, j \leq n) : (\lambda_i < \lambda_j \rightarrow \gamma_i \leq \gamma_j)$, and
   (b) $(\forall i, j)(1 \leq i, j \leq n) : (\lambda_i = \lambda_j \rightarrow \gamma_i = \gamma_j)$
3. A *strongly fair* bandwidth allocation satisfies: $(\exists \alpha^* > 0)(\forall i)(1 \leq i \leq n) : ((\lambda_i \leq \alpha^* \rightarrow \gamma_i = \lambda_i) \wedge (\lambda_i > \alpha^* \rightarrow \gamma_i = \alpha^*))$.
4. A *waste-free* bandwidth allocation satisfies:
   (a) If $\sum_{i=1}^{n} \lambda_i < 1$, then $\sum_{i=1}^{n} \gamma_i = \sum_{i=1}^{n} \lambda_i$, and
   (b) If $\sum_{i=1}^{n} \lambda_i \geq 1$, then $\sum_{i=1}^{n} \gamma_i = 1$.

The *fairness* condition guarantees that a station does not obtain more bandwidth than a station with a higher arrival rate, and stations with the same arrival rate obtain the same throughput. *Strong fairness* additionally guarantees that stations with an arrival rate less than a threshold value obtain all the bandwidth they need. Note that the condition for *strong fairness* implies *fairness.*

Remark: The bandwidth allocation in IEEE 802.6 without bandwidth balancing is *waste-free* but not *fair.* IEEE 802.6 with bandwidth balancing is *strongly fair* but not *waste-free.* A bandwidth allocation which assigns the bandwidth proportional to the arrival rates [12] is *fair* but not *strongly fair.*

Ideally, a bandwidth allocation is both *strongly fair* and *waste-free.* The following theorem is very useful in finding a *fair* and *waste-free* bandwidth allocation.

Theorem 1 *For each set of arrival rates there exists exactly one strongly fair and waste-free bandwidth allocation which is determined by the solution of the following system of equations:*

$$\gamma_i = \min \{\lambda_i\ ,\ \frac{1 - \sum_{j \in \mathbf{U}} \lambda_j}{|\mathbf{O}|}\} \tag{1}$$

with

$$\mathbf{U} = \{j \mid \lambda_j = \gamma_j\} \tag{2}$$

$$\mathbf{O} = \{j \mid \lambda_j > \gamma_j\} \tag{3}$$

where $\mathbf{U}$ denotes the set of station indices which satisfy their bandwidth demands, and $\mathbf{O}$ denotes the set of station indices where the arrival rate of segments exceeds the allocated bandwidth. A complete proof of the theorem is given in [10].

In the following theorem we show that the *strongly fair* and *waste-free* bandwidth allocation can be obtained in a distributed manner. The correct allocation can be obtained even if each station has only limited information about parameters of other stations. This allows us to present a distributed access protocol which implements the *strongly fair* and *waste-free* bandwidth allocation.

Definition 2 *For each station i $(1 \leq i \leq n)$ define Λ_i, Γ_i and $\mathbf{O}_i$ by:*

$$\Lambda_i = \sum_{i<j\leq n \wedge j \in \mathbf{U}} \lambda_j \tag{4}$$

$$\Gamma_i = \sum_{1\leq j<i} \gamma_j \tag{5}$$

$$\mathbf{O}_i = \{j \mid j \in \mathbf{O} \wedge i < j \leq n\} \tag{6}$$

Theorem 2 *Given a strongly fair and waste-free bandwidth allocation. Then, $i \in \mathbf{O}$ if and only if*

$$\gamma_i = \frac{1 - \Gamma_i - \Lambda_i}{|\mathbf{O}_i| + 1} \tag{7}$$

The proof can be found in [10].

Corollary 1 *The strongly fair and waste-free bandwidth allocation can be obtained from equations (3) - (7).*

In the following sections we will use the results of this section to present a protocol which achieves a *strongly fair* and *waste-free* bandwidth allocation. The protocol will be derived from the distributed version of the *strongly fair* and *waste-free* bandwidth allocation implied by Theorem 2.

4 DQDB$^{+/-}$: A Fair and Waste-Free Access Protocol

Based on the results of section 3 we present a new media access protocol for dual bus networks. The protocol, referred to as *DQDB*$^{+/-}$, has the same hardware requirements as the DQDB (IEEE 802.6) access protocol. Therefore, the features of the *DQDB*$^{+/-}$ protocol can be included into the current IEEE 802.6 standard [7]. We will show that the new scheme has several advantages over DQDB (with bandwidth balancing), such as:

- better adaptation towards changes of the network load,
- full utilization of the bus,
- better performance at high transmission speeds and/or physically long buses.

Here we consider only uni-priority traffic[2]. We discuss the design concepts of *DQDB*$^{+/-}$ and present an implementation of the scheme.

When discussing the access scheme, we consider only data transmission on bus A since channel access for data transmission on bus B is symmetric. We use the station index to denote the relative physical distance to the slot generator of bus A. So, station 1 denotes the station closest to the slot generator of bus A, station 2 the second closest station, etc.. The stations with greater index than station i are referred to as the downstream stations of station i, stations with smaller index are referred to as the upstream stations.

[2]Multiple priorities are treated in [11]

4.1 Design Concepts of DQDB$^{+/-}$

As mentioned in section 2, the DQDB protocol sends one reservation request for each segment to be transmitted. As a result, high contention for data transmission is reflected in high contention for sending reservation requests.

DQDB$^{+/-}$ departs from this scheme by following an access scheme which is adopted from the ***strongly fair*** and ***waste-free*** bandwidth allocation of section 3. In *DQDB*$^{+/-}$, stations are partitioned into two sets: the set of overloaded stations and the set of underloaded stations, respectively. An underloaded station obtains all the bandwidth it needs for transmission. An overloaded station cannot satisfy its bandwidth requirements, and it obtains the same bandwidth as all other overloaded stations. We denote the bandwidth which can be allocated by an overloaded station as a ***share*** or a ***quota***.

Both underloaded and overloaded stations use bus B to send reservation requests to upstream stations. However, only underloaded stations send a reservation request for each segment following the same protocol as DQDB. If an underloaded station becomes overloaded, it stops sending reservation requests, and sends a signal on bus B to notify the upstream stations that it is overloaded. Once the signal is set, no more reservations are sent. If an overloaded station becomes underloaded, it sends an opposite signal on bus B to indicate that it is underloaded. Then the station resumes sending reservation requests, one for each segment.

Before a station is allowed to transmit a segment it has to consider all reservations from downstream nodes. For each reservation request and for each overloaded station downstream the station has to leave an empty slot on bus A.

The advantages of this reservation scheme over IEEE 802.6 are twofold. First, there is little or no contention for sending reservation, since overloaded stations do not transmit reservation requests. Secondly, since a received overload signal acts like a permanent reservation request, a station is able to obtain a quota of the bandwidth in one round-trip delay.

From Theorem 2 we know that a necessary and sufficient condition for a station i to be overloaded is given equation (7). Note that we use the station index to denote the relative position of a station on bus A. Therefore, the values of equation (7) are given by:

Γ_i: rate of busy slots seen by station i,
Λ_i: rate of reservation requests received by station i,
$|O_i|$: number of overloaded stations downstream on bus A,
λ_i: arrival rate of segments to station i.

4.2 Implementation of DQDB$^{+/-}$

The overhead of implementing *DQDB*$^{+/-}$ compared to an implementation of IEEE 802.6 consists in two additional bits in the slot header, referred to as the ***plus bit*** and ***minus bit***, and in a few additional counters[3]. The slot header in *DQDB*$^{+/-}$ therefore contains a busy bit, a request bit, a plus bit, and a minus bit. The busy bit and the request bit are used as in DQDB. The busy bit is set by a station when inserting data into the slot. The request bit is set to transmit a reservation request for a single segment. The plus bit is set by a station to indicate that it is overloaded, and the minus bit is set by a station to indicate that it is underloaded.

Each station determines its turn to transmit a segment with four counters, the request counter (*RQ*), the countdown counter (*CD*), the overload request counter (*ORQ*) and the overload countdown counter (*OCD*). *RQ* and *CD* have the same functions as in the DQDB protocol. An idle station, i.e., a station that does not have a segment queued for transmission, increments *RQ* for each passing slot on bus B with the request bit set. *ORQ* is incremented for each passing slot on bus B with the plus bit set, and decremented by one for each slot with the minus bit set. For each empty slot passing by on bus A the station decrements *RQ* by one as long as *RQ* is greater than zero.

If a segment arrives at an idle station, the contents of *RQ* and *ORQ* are copied to *CD* and *OCD*, and both *RQ* and *ORQ* are set to zero. Now, *RQ* is incremented for each set request read on bus B. *ORQ* is incremented for each set plus bit and decremented for each set minus bit. If a minus bit is read on bus B, and *ORQ* is zero, *OCD* is decremented by one. For each empty slot on bus A, *CD* is decremented by one. If *CD* is zero the station decrements *OCD* by one, and increments *ORQ* by one. If an empty slot is read and both *CD* and *OCD* are zero, the empty slot is used for transmission of the segment. If the station has more segments waiting for transmission, *RQ* and *ORQ* are copied to *CD* and *OCD*, and then set to zero.

A station can be in one of four states, UL, UL^*, OL and OL^*. A station is underloaded in states UL and UL^*, and it is overloaded in states OL and OL^*.

[3]The plus and minus bit can be accommodated in the two unused bits of the access control field in an IEEE 802.6 slot header [7]

	State	Type of Reservation
underloaded	UL	send reservation requests
	UL^*	set minus bit, send reservation requests
overloaded	OL	delete request queue, no reservation requests
	OL^*	set plus bit no reservation requests

Table 1: State Description of $DQDB^{+/-}$.

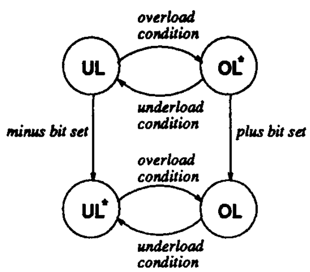

Figure 1: State Transitions in $DQDB^{+/-}$.

Each station in state UL sends one reservation request for the segment on top of the queue of untransmitted segments. This is done by setting the request bit in a slot on bus B. Setting the request bit may be delayed since a station cannot set the request bit if it is already set. Therefore, each station maintains a queue REQ_QUE of untransmitted request bits.

If a station in state UL becomes overloaded, it enters state OL^* and attempts to set the plus bit in a slot on bus B. No more reservation requests are transmitted. If the station becomes underloaded before the plus bit is set, it returns to state UL, and resumes sending reservations. Otherwise, the station enters state OL and sets REQ_QUE to zero. If a station in state OL becomes underloaded, it enters state UL^* and resumes sending reservation requests. In state UL^* a station attempts to set a minus bit on bus B. If the station becomes overloaded before the minus bit is set, it re-enters state OL. Otherwise, it enters state UL.

The states with the respective strategies for reserving slots are summarized in Table 1. Figure 1 depicts the state transition diagram. Each station determines by itself whether it is overloaded or underloaded. The rates needed to calculate equation (7) are obtained from the values of counters. Most of the required information is already stored in counters *RQ*, *CD*, *RQ*, and *CD*. Three additional counters are needed. *NoSeg* contains the total number of segments queued for transmission, *SlotCtr* is incremented for each arriving slot on bus B, and *Bsy* is incremented for each busy slot read on bus A. A station determines its state each time *Basis* slots have passed by on bus B ($SlotCtr = Basis$). Then it calculates:

$$Quota = \frac{Basis - Bsy - RQ - CD}{ORQ + OCD + 1}, \quad (8)$$

and sets counters *SlotCtr* and *Bsy* to zero. *Quota* provides the local value of a share of the network load, that is, the maximum number of slots a station can transmit during a period of *Basis* slots. If $NoSeg > Quota$, the station is overloaded, otherwise the station is underloaded. If a state change has occurred, the station takes the appropriate action as described above.

Remark: The performance of the protocol is quite insensitive towards the choice of *Basis*. Since the propagation of information in a dual bus architecture is limited by the round-trip delay, we set *Basis* to the sum of the slot lengths of bus A and bus B. Then, *Quota* denotes the maximum number of segments each station is allowed to transmit in a round-trip delay. If *Basis* is chosen large each station i will calculate *Quota* closer to the right-hand side of equation (7), but it will react slower to changes in the network load. Small values for *Basis* increase the reactivity of a station towards load changes, but the calculation of *Quota* as an estimate of equation (7) will be less accurate.

5 Comparison of DQDB$^{+/-}$ with IEEE 802.6

To evaluate the performance of $DQDB^{+/-}$ we compare our protocol with the DQDB (IEEE 802.6) protocol (including bandwidth balancing). The performance comparison is done by simulation. We use a simulator for dual bus networks which is presented in [1]. Two types of simulations are presented. Simulations of short periods allow to study how the protocols adapt to changes in the network load. Long term simulations provide mean performance measures of a network under fixed traffic assumptions.

5.1 Transient Behavior During File Transfers

For illustrative purposes we study a network with three active stations. The distance between adjacent

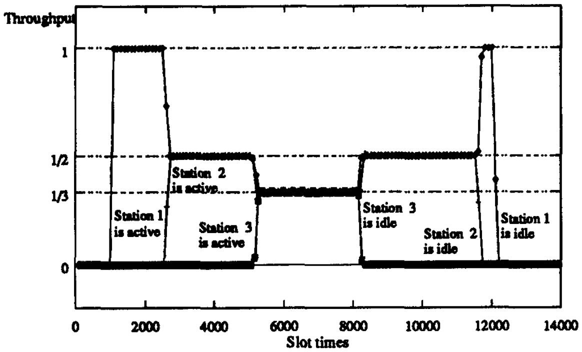

Figure 2: File Transfer with $DQDB^{+/-}$ ($Basis$ = 100; Round-trip delay = 100 slots)

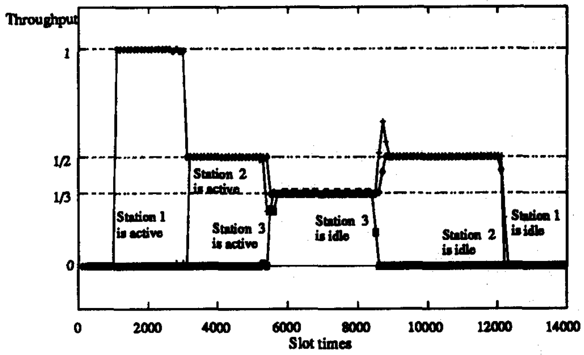

Figure 4: File Transfer with $DQDB^{+/-}$ ($Basis$ = 400; Round-trip delay = 400 slots)

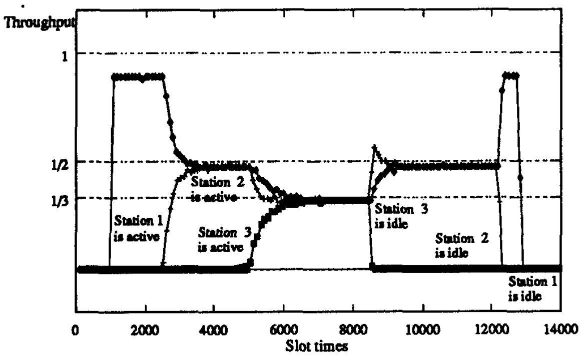

Figure 3: File Transfer with IEEE 802.6 (BWB_MOD = 8; Round-trip delay = 100 slots)

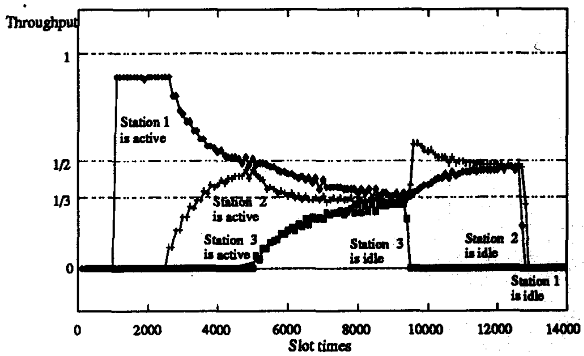

Figure 5: File Transfer with IEEE 802.6 (BWB_MOD = 8; Round-trip delay = 400 slots)

stations is assumed to be 25 slots. The total round-trip delay of the bus is given by 100 slots. All station start file transfers at different times. At $t = 0$ the network is empty, at $t = 1000$ station 1 starts to transmit 6000 segments, at $t = 2500$ station 2 starts to transmit 4000 segments, and at $t = 5000$ station 3 starts to transmit 1000 segments. We measure the throughput of a station, i.e., the number of transmitted segments, once per round-trip (every 100 slots) for a period of 14000 slots. Figure 2 shows the results for the $DQDB^{+/-}$ protocol with $Basis = 100$. For comparison, we show in Figure 3 a simulation of the same scenario in a IEEE 802.6 network with a bandwidth balancing modulo of $BWB\_MOD = 8$ [4]. We see that $DQDB^{+/-}$ immediately adapts to changes of the network load. Moreover, the full bandwidth is utilized. IEEE 802.6 does not only waste a certain percentage of the bandwidth, but also takes considerable time until each station obtains the same share of the bandwidth. Note that the drawbacks of IEEE 802.6 result in longer transmission delays of the stations.

The advantages of $DQDB^{+/-}$ over IEEE 802.6 become more apparent when the slot distance between the stations is increased. Increasing the slot distance corresponds to increasing the physical distance between stations, or equivalently, increasing the transmission speed of the network. We present the same simulation scenario for a network with a slot distance of 100 slots between two adjacent stations. Figures 4 and 5 present the results of $DQDB^{+/-}$ (with $Basis$ = 400) and IEEE 802.6 (with $BWB\_MOD = 8$).

[4] $BWB\_MOD = 8$ is the default value in the IEEE 802.6 standard [7].

Experi- -ment	ρ	m	Δ
I	0.65, 0.75, 0.85	100 slots	2 slots
II	0.65, 0.75, 0.85	100 slots	10 slots

Table 2: Simulation Parameters.

5.2 Steady State Behavior for Fixed Workloads

We simulate a dual bus network with 10 stations for a duration of 5 million slots. The parameters that define the workload of the network remain unchanged for the entire simulation. We present several experiments where in each experiment the following parameters are varied:

m : number of segments per message,
ρ : total traffic load of a bus,
Δ : slot distance between stations.

We measure the traffic on both buses. The traffic load, ρ, is the same for both bus A and bus B. We assume that traffic between each two stations is the same for all stations. Then, the traffic load of a station on a particular bus is proportional to the number of downstream stations on that bus. Note that the most downstream station on bus A (B) does not generate any traffic for bus A (B). The time between arrivals of messages to a station are exponentially distributed. We present results of simulation runs with the parameters shown in Table 2. We focus on presenting results for the mean delay of a message, that is, the time from the arrival of a message to a station until the last segment of the message is transmitted.
The delay is given in slot time units. For IEEE 802.6 we assume a bandwidth balancing modulo $BWB\_MOD = 8$. For $DQDB^{+/-}$ we set *Basis* equal to the round-trip slot delay (see section 4.2), i.e., $Basis = 36$ if $\Delta = 2$, and $Basis = 180$ if $\Delta = 10$, respectively.

In Figures 6 and 7 we depict the mean message delays for Experiments I and II, respectively. The traffic load of each bus is varied from $\rho = 0.65$ to $\rho = 0.85$. It shows that $DQDB^{+/-}$ achieves significantly lower mean message delays for all traffic loads.

6 Conclusions

We presented a new protocol for dual bus networks, referred to as $DQDB^{+/-}$, which does not show the disadvantages of the IEEE 802.6 protocol. The new protocol is based on theoretical considerations in section 3. There we showed the uniqueness of a fair bandwidth allocation scheme that utilizes the entire bandwidth. We described the $DQDB^{+/-}$ protocol in section 4. We showed that at the cost of two additional bits in the slot header, $DQDB^{+/-}$ is able to achieve a fair distribution of the bandwidth in one round-trip delay. The behavior of the protocol was compared to the IEEE 802.6 protocol with bandwidth balancing. We demonstrated that $DQDB^{+/-}$ is superior to IEEE 802.6 in many aspects. It converges faster to a fair distribution of the bandwidth to the stations and is able to utilize the full bandwidth. We showed that the transmission delays can be reduced by using $DQDB^{+/-}$. The advantages of $DQDB^{+/-}$ over IEEE 802.6 are even more apparent for large networks and high transmission speeds. The current IEEE 802.6 standard can be upgraded to include the features of $DQDB^{+/-}$.

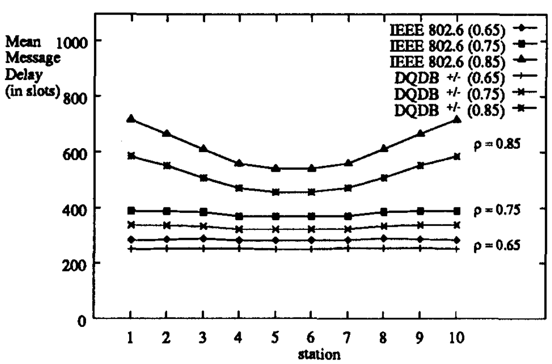

Figure 6: Mean Message Delay (Experiment I).

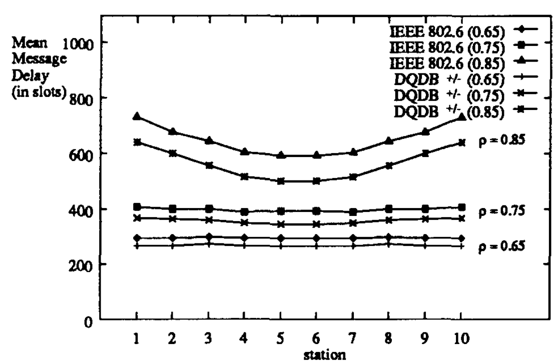

Figure 7: Mean Message Delay (Experiment II).

Acknowledgement

We thank Chatschik Bisdikian for providing us his simulator for dual bus networks. We are also grateful to Asser N. Tantawi and Ahmed Tantawy for their comments and suggestions.

References

[1] C. Bisdikian. DQDB Delay Analysis: The BWB Mechanism Case. In *Fourth IEEE Workshop on MANs*, pages 5.1.1 – 5.1.28, November 1990.

[2] F. R. Dix, M. Kelly, and R. W. Klessig. Access to a Public Switched Multi-Megabit Data Service Offering. *Computer Communications Review*, 20(3):46 – 61, July 1990.

[3] J. Filipiak. Access Protection for Fairness in a DQDB MAN. In *Proc. ICC 89*, pages 635 – 639, Boston, MA, June 1989.

[4] Y. Gong and M. Paterakis. Design and Performance Analysis of an Accurate and Fast Dynamic Protocol Achieving User Fairness in High Speed Dual Bus Networks. In *Proc. INFOCOM '91*, pages 479 – 488, Bal Harbour, FL., April 1991.

[5] E. L. Hahne, A. K. Choudhury, and N. F. Maxemchuk. Improving the Fairness of Distributed Queue-Dual-Bus Networks. In *Proc. INFOCOM '90*, pages 175 – 184, San Francisco, CA., June 1990.

[6] E. Y. Huang and L. F. Merakos. On the Access Fairness of the DQDB MAN. In *Proc. IEEE 1990 Phoenix Conference*, pages 556 – 559, Phoenix, AZ, March 1990.

[7] IEEE. Std 802.6-1990, IEEE Standards for Local and Metropolitan Area Networks: Distributed Queue Dual Bus (DQDB) of a Metropolitan Area Network (MAN), July 1991.

[8] K.M. Khalil and M.E. Koblentz. A Fair Distributed Queue Dual Bus Access Method. In *Proc. 14th Conf. on Local Computer Networks*, pages 180 – 188, Minneapolis, MN, October 1989.

[9] Y.-S. Leu and D. H. C. Du. Cycle Compensation Protocol: A Completely Fair Protocol for the Unidirectional Twin-Bus Architecture. In *Proc. 15th Conf. on Local Computer Networks*, Minneapolis, MN, October 1990.

[10] J. Liebeherr, I.F. Akyildiz, and A.N. Tantawi. Distributed Fair and Waste-Free Media Access Protocols for Dual Bus Metropolitan Area Networks. Technical Report RC 17489, IBM Research Division, December 1991.

[11] J. Liebeherr, I.F. Akyildiz, and A.N. Tantawi. An Effective Scheme for Pre-Emptive Priorities in Dual Bus Metropolitan Area Networks, August 1992. to appear: Proc. ACM Sigcomm '92.

[12] B. Mukherjee and A. Banerjee. Alternative Strategies for Improving the Fairness in and an Analytical Model of DQDB Networks. In *Proc. INFOCOM '91*, pages 879 – 888, Bal Harbour, FL., April 1991.

[13] B. Mukherjee and C. Bisdikian. A Journey Through the DQDB Literature. to appear: Performance Evaluation Journal.

[14] M. P. Spratt. Allocation of Bandwidth in IEEE802.6 using Non-Unity Ratio Bandwidth Balancing. In *Proc. ICC 1991*, pages 729–735, Denver, CO, June 1991.

[15] J. W. Wong. Throughput of DQDB Networks under Heavy Load. In *EFOC/LAN*, Amsterdam, Netherlands, June 1989.