DQDB$^{+/-}$ - A Fair and Waste-Free Media Access Protocol for Dual Bus Metropolitan Networks

Ian F. Akyildiz *   *Jörg Liebeherr* **   *Asser N. Tantawi* ***

* School of Electrical Engineering,
Georgia Institute of Technology,
Atlanta, GA 30332,
U. S. A.

** Computer Science Department,
University of Virginia,
Charlottesville, VA 22903,
U. S. A.

*** IBM Research Division,
Thomas J. Watson Research Center,
Yorktown Heights, NY 10598,
U. S. A

February 26, 1993

Abstract

The Distributed Queue Dual Bus protocol, the media access protocol of the IEEE 802.6 standard for Metropolitan Area Networks, does not fully take advantage of the capabilities of a dual bus architecture. Although a fair bandwidth distribution among the stations is guaranteed when using the so-called bandwidth balancing mechanism, the protocol requires a considerable amount of time to adjust to changes in the network load. Additionally, the bandwidth balancing mechanism leaves a portion of the available bandwidth unused. In this study it is shown that the shortcomings of the IEEE 802.6 standard can be overcome by adopting a different fairness mechanism. A new media access protocol for dual bus networks is presented that achieves a fair distribution of the bandwidth in one round-trip delay. The protocol is based on the unique solution to a so-called fair and waste-free bandwidth allocation. This bandwidth allocation can be implemented in a distributed manner. A comparison of the new protocol with the *DQDB* protocol shows considerable advantages regarding the transmission delay of messages and the time a station needs to obtain a fair portion of the available bandwidth. The advantages of the protocol become more apparent for large networks and high transmission speeds. In addition, the new protocol can provide services which do not distribute the bandwidth equally among the stations, i.e., it can perform non-uniform bandwidth allocations. Examples of several non-uniform bandwidth allocations are presented, such as a per-user bandwidth allocation, a guaranteed bandwidth mechanism, and a (quasi-) priority scheme.

Key Words: Metropolitan Area Network, Dual Bus, DQDB, Bandwidth Guarantee, Media Access.

1 Introduction

The Distributed Queue Dual Bus ($DQDB$) protocol has received considerable attention since it was endorsed by the IEEE 802.6 Committee as the standard for Metropolitan Area Networks. It was shown that without an additional fairness mechanism the $DQDB$ protocol is not able to achieve an equal distribution of the bandwidth among the stations if the network is heavily loaded [9]. Numerous modifications to the protocol have been proposed that ensure a fair distribution of bandwidth under heavy network load. An overview over the various methods is given in [5] and [7]. The most influential extension to the $DQDB$ protocol is the so-called *bandwidth balancing mechanism* [3] that was incorporated into the IEEE 802.6 standard [4] [1]. Bandwidth balancing achieves a fair distribution of the bandwidth by enforcing that each station uses only a fraction of the available bandwidth for transmissions.

Even after adopting the bandwidth balancing mechanism, the current IEEE 802.6 standard [4] leaves the protocol with two major drawbacks. First, the protocol does not allow to fully utilize the bandwidth of the buses. Secondly, the bandwidth balancing scheme converges only slowly to a fair distribution of the bandwidth to the stations.

In this study, we introduce a new protocol for a dual bus network. The new protocol does not have the disadvantages inherent to the IEEE 802.6 standard. For a heavily loaded network with an arbitrary number of transmitting stations a fair distribution of bandwidth is obtained after a time corresponding to one round-trip delay of the bus. Additionally, the protocol allows to utilize the full bandwidth of the buses. The protocol is highly adaptive to changes in the network load. The design of the protocol is based on the concept of a so-called *fair and waste-free bandwidth allocation*. We show that the new protocol is able to provide a number of services which do not distribute the bandwidth equally among the stations.

The remaining sections are organized as follows. In section 2 we derive properties of optimal, i.e., fair and waste-free, bandwidth allocations for dual bus networks. In section 3 we use the results from section 2 to define a new protocol for dual bus networks. We refer to the new protocol as $DQDB^{+/-}$. We compare the performance of $DQDB^{+/-}$ with the IEEE 802.6 standard. In section 4 we show that $DQDB^{+/-}$ can be used to implement a variety of services which do not distribute the bandwidth uniformly among the stations. In section 5 we conclude our results.

2 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 scheme which guarantees a fair distribution of the bandwidth to the stations and utilizes the full bandwidth. The theoretical results from this section are directly applied in the following sections where we propose a multi access protocol for dual bus networks. Because of the symmetry of the dual bus topology we only consider transmissions on one bus. Formally, a bandwidth allocation maps the traffic load from all stations into individual portions of the bandwidth that can be used for transmission.

[1] In the following we will use '$DQDB$ with bandwidth balancing' and 'IEEE 802.6' synonymously.

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 load and the throughput of station i ($1 \leq i \leq n$), respectively, on a single bus. Both load and allocated bandwidth are normalized over the total bandwidth. Let $\underline{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ denote the network load.

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 *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 $\sum_{i=1}^{n} \gamma_i = 1$.
3. A bandwidth allocation is *fair* if for any two stations i and j ($1 \leq i, j \leq n$) the following conditions hold:
   (a) if $\lambda_i < \lambda_j$ then $\gamma_i \leq \gamma_j$, and
   (b) if $\lambda_i = \lambda_j$ then $\gamma_i = \gamma_j$.
4. A bandwidth allocation is *strongly fair* if there exists a value $\alpha^* > 0$ such that for each station i ($1 \leq i \leq n$) the following conditions hold:
   (a) if $\lambda_i \leq \alpha^*$ then $\gamma_i = \lambda_i$, and
   (b) if $\lambda_i > \alpha^*$ then $\gamma_i = \alpha^*$.

A waste-free bandwidth allocation can utilize the entire bandwidth for transmission if the load is sufficiently high. The fairness condition guarantees that a station does not have a higher throughput than a station with a higher load, and stations with the same load obtain the same throughput. Strong fairness additionally guarantees that for a given network load $\underline{\lambda}$ each station cannot achieve a higher throughput than a given threshold value α^*. Stations with a load less than the threshold value obtain all the bandwidth they need. The value of α^* is dependent on the offered load from all stations. Since α^* denotes the maximum throughput of a station for a given load vector, we will refer to the value of α^* as a *share* or a *quota*. Note that the condition for strong fairness implies fairness.

Remark: The bandwidth allocation of the DQDB protocol without bandwidth balancing is waste-free but not fair. DQDB with bandwidth balancing is strongly fair but not waste-free. A bandwidth allocation which assigns the bandwidth proportional to the arrival rates [6] is fair but not strongly fair.

Ideally, a bandwidth allocation should be both strongly fair and waste-free. In the following we show the existence and uniqueness of a strongly fair and waste-free bandwidth allocation. The bandwidth allocation requires the partition of all stations into so-called *underloaded* and *overloaded* stations. Underloaded stations can transmit the entire load, and all overloaded stations obtain the same portion of the bandwidth that is not used by underloaded stations.

Theorem 1 *There exists exactly one strongly fair and waste-free bandwidth allocation which is determined by the solution of*

$$\gamma_i = \mathbf{min}\ \{\lambda_i\ ,\ \alpha^*\} \tag{1}$$

where

$$\alpha^* = \begin{cases} \dfrac{1 - \sum_{j \in \mathbf{U}} \lambda_j}{|\mathbf{O}|} & \textit{if } \mathbf{O} \neq \emptyset \\ \infty & \textit{otherwise} \end{cases} \tag{2}$$

and

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

$$\mathbf{O} = \{j \mid \lambda_j > \gamma_j\} \tag{4}$$

$\mathbf{U}$ and $\mathbf{O}$ denote the index set of underloaded stations and overloaded stations, respectively. The proof of Theorem 1 is given in Appendix A. It follows from Theorem 1 that any bandwidth allocation that is not able to distinguish between underloaded and overloaded stations cannot achieve both strong fairness and waste-freedom.

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:*

$$\Lambda_i = \sum_{i < j \leq n \wedge j \in \mathbf{U}} \lambda_j \tag{5}$$

$$\Gamma_i = \sum_{1 \leq j < i} \gamma_j \tag{6}$$

$$\mathbf{O}_i = \{j \mid j \in \mathbf{O} \wedge i < j \leq n\} \tag{7}$$

Λ_i denotes the accumulated load of all underloaded stations with higher index than station i, Γ_i denotes the accumulated throughput of the stations with lower index than station i, and $\mathbf{O}_i$ denotes the set of overloaded stations with higher index than station i.

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

$$\lambda_i > \frac{1 - \Gamma_i - \Lambda_i}{|\mathbf{O}_i| \ + \ 1} \tag{8}$$

We prove Theorem 2 in Appendix B. From Theorem 2 we can conclude that the strongly fair and waste-free bandwidth allocation can be obtained if each station i has knowledge of parameters $(\lambda_i, \Lambda_i, \Gamma_i, \mathbf{O}_i)$. 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.

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

The new protocol, referred to as $DQDB^{+/-}$, has the same hardware requirements as the DQDB (IEEE 802.6) access protocol. Therefore, the IEEE 802.6 standard [4] can be extended to include the features of the $DQDB^{+/-}$ protocol. 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.

We only consider uni-priority traffic. We discuss the design concepts of $DQDB^{+/-}$ and present an implementation of the scheme. Then we compare our protocol with the IEEE 802.6 standard.

We distinguish the buses by referring to them as bus A and bus B. We only consider data transmission on bus A since channel access for data transmission is symmetric for both buses. We will use the station index to denote the relative physical distance to the slot generator of bus A. So, station 1 will denote 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.

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

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 obtains the same bandwidth as all other overloaded stations.

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 [4]. 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 stations. 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 requests, 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 the necessary and sufficient condition for a station i to be overloaded. Since we use the station index to denote the relative position of a station on bus A the values of equation (8) are given by:

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

Next we describe how to implement the $DQDB^{+/-}$ protocol.

3.2 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 additional counters [2]. The slot header in $DQDB^{+/-}$ therefore contains a busy bit, a request bit, a plus bit, and a minus bit. The busy bit is set by a station when inserting data into the slot. The other bits are used for sending reservation requests.

An underloaded station sends a reservation request by setting a request bit in a slot on bus B. One request bit is set for each segment to be transmitted. If an underloaded module becomes overloaded, it sets a plus bit in a slot on bus B. After setting the plus bit, no more reservation requests are transmitted. If an overloaded station becomes underloaded, it sets a minus bit, and resumes setting request bits, one for each segment.

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

[2] 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 [4].

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 RQ is set to zero. The value of ORQ remains unchanged. 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. For each empty slot on bus A, CD is decremented by one. If CD is zero the station decrements OCD 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 RQ is set to zero.

Each station can determine whether it is overloaded or underloaded. The rates needed to calculate equation (8) are obtained from the values of counters. Most of the required information is stored in counters RQ, CD, RQ, and CD. Three additional counters are needed. SEG_CTR contains the total number of segments queued for transmission, $SLOT\_CTR$ is incremented for each arriving slot on bus B, and BSY_CTR is incremented for each busy slot read on bus A. A station evaluates its state each time after an interval of `basis` slots have passed by on bus B ($SLOT\_CTR = \texttt{basis}$). Then it calculates:

$$quota = \frac{\texttt{basis} - BSY\_CTR - RQ - CD}{ORQ + 1}, \tag{9}$$

and sets counters $SLOT\_CTR$ and BSY_CTR 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 $SEG\_CTR > quota$, the station is overloaded, otherwise the station is underloaded. If a state change has occurred, the station takes the appropriate action as described above.
A complete description of the $DQDB^{+/-}$ protocol is given in [5].

Remark: The value for parameter `basis` can be chosen from a wide range of values without having an effect on the performance of the protocol. 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 can 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 (8), 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 (8) will be less accurate.

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

In order to evaluate the performance of $DQDB^{+/-}$ we compare our protocol with the IEEE 802.6 standard, i.e., with DQDB including bandwidth balancing, by simulating a dual bus network [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.

3.3.1 Transient Behavior During File Transfers

For illustrative purposes, we study a network with three active stations. The distance between adjacent stations is assumed to be 25 slots [3]. 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 1 shows the results for the $DQDB^{+/-}$ protocol with `basis` = 100. For comparison, we show in Figure 2 a simulation of the same scenario in a IEEE 802.6 network with a bandwidth balancing modulo of `bwb_mod` = 8. [4]

It can be seen 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 also that the drawbacks of IEEE 802.6 result in longer transmission delays of the stations, as can be seen in Figures 1 and 2.

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. In Figures 3 and 4 we present the results of $DQDB^{+/-}$ (with `basis` = 400) and IEEE 802.6 (with `bwb_mod` = 8).

3.3.2 Steady State Behavior for Fixed Workload

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 in a message,
ρ	:	total traffic load of a bus,
Δ	:	slot distance between stations.

Traffic is measured on both buses. The traffic load, ρ, is the same for both buses A and B. We assume that traffic between any two stations is symmetric. 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 simulate the dual bus network with the parameter sets shown in Table 1.

[3] With a slot size of 53 bytes [4] one slot corresponds to a length of 1896 m at transmission rate of 44.7 Mb/s, to 546 m at 155.5 Mb/s and to 137 m at 622 Mb/s.

[4] `bwb_mod` = 8 is the default value in the IEEE 802.6 standard [4].

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. We will only provide mean delay measures for network loadings of $\rho < 0.90$. For $\rho > 0.90$ the network will be overloaded most of the time [1]. Note that in overloaded networks some station will eventually accumulate an infinite queue, and thus, will experience an infinite message delay. When simulating the IEEE 802.6 protocol we assume a bandwidth balancing modulo `bwb_mod` = 8. For simulation of $DQDB^{+/-}$ we set `basis` equal to the round-trip slot delay (see section 3.2), i.e., `basis` = 36 if $\Delta = 2$, and `basis` = 180 if $\Delta = 10$, respectively.

In Figures 5, 6, 7 and 8 we depict the mean message delays of each station for Experiments I, II, III and IV. It shows that $DQDB^{+/-}$ yields lower mean message delays than IEEE 802.6. If the message length is long (Experiments II and IV), the difference between $DQDB^{+/-}$ and IEEE 802.6 becomes more obvious. Note that in $DQDB^{+/-}$ an arrival of a long message will cause a previously idle station to become overloaded. Since an overloaded station in $DQDB^{+/-}$ obtains a fair share of the network load in less time than in IEEE 802.6, the message will be transmitted in a shorter time.

4 DQDB$^{+/-}$ for Non-Uniform Bandwidth Allocation

In this section we present an extension to the $DQDB^{+/-}$ protocol that can provide *non-uniform bandwidth allocations*, i.e., bandwidth allocations which do not distribute the bandwidth equally among the stations. Non-uniform bandwidth allocations are required to provide the following services [8]:

- Some stations might want to acquire more than one share of the network bandwidth. Note that obtaining more bandwidth results in shorter transmission delays at a station.
- A *per-user* bandwidth allocation distributes the available bandwidth equally among the users of the network. Each station obtains as many shares of the bandwidth as it has active users.
- Some applications require a bandwidth guarantees for transmission. In most cases the application will not fully use the guaranteed bandwidth all the time. It is desirable to let other stations allocate the unused bandwidth and have the guaranteed bandwidth ready when it is needed. For instance, the required bit rate for compressed video transmissions is dependent on the movement of the transmitted picture. Although the maximum bit rate must be available when needed, it is not required if pictures containing little movement are transmitted.

In the following we show that the uni-priority $DQDB^{+/-}$ protocol can provide services with a non-uniform bandwidth allocation if stations are allowed to transmit more than one plus bit. Then we present examples of different services that apply non-uniform bandwidth allocations. We refer to [5] for a detailed discussion of implementation issues.

4.1 Design Concepts of DQDB$^{+/-}$ for Non-Uniform Bandwidth Allocation

The $DQDB^{+/-}$ protocol enforces that each overloaded station obtains a fair share of the network bandwidth by sending a plus bit to the upstream stations. Consequently, if a station is permitted to transmit more than one plus bit, it obtains one share for each transmitted plus bit.

Non-uniform bandwidth allocations can be achieved by assigning two parameters P_i^{max} and λ_i^G to each station i, where:

1. P_i^{max} denotes the maximum number of plus bits a station is allowed to transmit. The default value is $P_i^{max} = 1$. The value of P_i^{max} does not need to be fixed. For example, P_i^{max} can be set to the number of users transmitting from station i.

2. λ_i^G denotes the bandwidth guarantee to station i. The default value is $\lambda_i^G = 0$. The sum of guaranteed bandwidth to all stations should not exceed the available bandwidth, i.e., $\sum_{i=1}^{n} \lambda_i^G \leq 1$ [5]. A station with a bandwidth guarantee is allowed to transmit as many plus bits as needed to achieve a throughput equal to the guaranteed bandwidth, i.e., $\gamma_i = \lambda_i^G$.

A station with permission to transmit multiple plus bits should reserve at most as much bandwidth as it is actually going to use for transmission. If station i reserves more bandwidth than it uses, the upstream stations ($j < i$) perceive the network as heavily loaded, whereas the downstream stations ($j > i$) see a lightly loaded network. Therefore, if a station i has a current bandwidth demand that is not an integral multiple of the quota (denoted by ξ_i), it transmits a number of plus bits (P_i) corresponding to the highest integer less than the number of needed quotas. The remaining bandwidth, denoted by ϕ_i, is reserved by setting request bits.

The correct reservation strategy is obtained by first calculating the throughput guarantee of the station. Since the bandwidth guarantee should not exceed the actual arrival rate of segments at a station, the guaranteed throughput of station i, denoted by β_i, is given by:

$$\beta_i = \mathbf{min} \{\lambda_i, \mathbf{max} \{\lambda_i^G, P_i^{max} \cdot \xi_i\}\} \tag{10}$$

From β_i the station calculates the number of needed quotas. If ξ_i denotes the current value of a quota at station i, the station has a need for β_i/ξ_i quotas. If $\beta_i/\xi_i < 1$ the station does not send any plus bits. Instead, it sends request bits at rate λ_i. Otherwise, the station sets P_i plus bits on bus B with:

$$P_i = \lfloor \frac{\beta_i}{\xi_i} \rfloor \tag{11}$$

Note that during initialization of the network, no station has transmitted a plus bit, i.e., $P_i = 0$. If $P_i < P_i^{max}$ the station additionally sets reservation requests at a rate φ_i with:

$$\varphi_i = \frac{\beta_i}{\xi_i} - \xi_i \cdot \lfloor \frac{\beta_i}{\xi_i} \rfloor . \tag{12}$$

[5] This requires either an agent that grants guarantees to the stations, or a negotiation between stations that require bandwidth guarantees.

Then a waste-free bandwidth allocation is obtained if the quota ξ_i is calculated by:

$$\xi_i = \frac{1 - \Gamma_i - \Lambda_i - \phi_i}{|\mathbf{O}_i| \; + \; \mathbf{max}\{1, P_i\}} \tag{13}$$

where

$$\phi_i = \begin{cases} \varphi_i & \text{if } 1 \leq P_i < P_i^{max} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

If $\lambda_i^G = 0$ and $P_i^{max} = 1$ for all stations ($1 \leq i \leq n$), the protocol reduces to the $DQDB^{+/-}$ protocol described in section 3.

4.2 Examples of Non-Uniform Bandwidth Allocations

Non-uniform bandwidth allocation allow to provide a variety of services. Here we present three applications of non-uniform bandwidth allocations:

1. a per-user bandwidth allocation where the allocated bandwidth is proportional to the number of users at that station,

2. a guaranteed bandwidth scheme where some stations have a guaranteed minimal throughput,

3. a quasi-priority scheme where high priority stations obtain more bandwidth than low priority stations; but high priority stations are not unaffected by low priority traffic.

4.2.1 Per-User Bandwidth Allocation

Consider a network with the same parameters as in section 3.3.1. Three stations with a slot distance of 25 slots between adjacent stations start file transfers at different times. Assume station 1 has one active user, station 2 two users and station 3 three users. We obtain a per-user bandwidth allocation by setting $P_1^{max} = 1$, $P_2^{max} = 2$, and $P_3^{max} = 3$.

Figure 9 shows the results for the $DQDB^{+/-}$ protocol with `basis` = 100. $DQDB^{+/-}$ quickly adapts to the load changes in the network and the theoretical values for the ratio of allocated bandwidth are strictly enforced. Note that the entire bandwidth is utilized. If station 1 and station 2 are active, they share the bandwidth at a ratio 1 : 2. If all stations are active, the ratio of allocated bandwidth between stations 1, 2 and 3 is exactly 1 : 2 : 3.

4.2.2 Guaranteed Bandwidth Allocation

We assume the same network with three active stations as before. Station 2 has a guaranteed bandwidth of $\lambda_2^G = 0.7$. The other stations do not have bandwidth guarantees, i.e., $\lambda_1^G = \lambda_3^G = 0$.

Note that a station with a bandwidth guarantee that becomes active while the network is heavily loaded may initially calculate a need for an infinite number of quotas. As a consequence, the station sets a high number of plus bits and, eventually, will acquire the entire bandwidth. When the station realizes that is reserved too much bandwidth it compensates by setting minus bits on bus B. We avoid this temporary throttling of the network by limiting the number of plus bits a station is allowed to set at a time.

In Figure 10 we see that as soon as station 2 becomes active ($t = 2500$), it quickly obtains the guaranteed bandwidth and leaves station 1 the remaining portion. Eventually, both stations calculate the same value for a quota, i.e, $\xi_1 = \xi_2 = 0.3$. Station 2 obtains the guaranteed bandwidth by setting two plus bits ($P_i = 2$) and sending reservation requests at a rate of $\phi_i = 0.1$. When station 3 starts transmission ($t = 5000$), the throughput of station 2 drops because of the reservations from station 3. Station 2 adopts by recalculating the quotas. After a short transient phase, all stations calculate the quota $\xi_1 = \xi_2 = \xi_3 = 0.15$. Stations 1 and 3 obtain one quota; station 2 obtains $P_2 = 4$ quotas and sends reservation requests at a rate $\phi_i = 0.1$.

As soon as station 2 finishes its file transfer ($t = 8316$), station 1 and 3 share the available bandwidth. Because of the propagation delay of the minus bits sent by station 2 after it finishes transmission, station 3 is able for a short period of time to pick up some of the bandwidth that was reserved by station 2.

4.2.3 A Quasi-Priority Scheme

$DQDB^{+/-}$ allows the implementation of a simple priority scheme. Stations are partitioned into disjoint classes and all stations of a class c are assigned the same value for the maximum number of plus transmitted bits, i.e., $P_i^{max} = p_c$ for all $i \in c$. If the difference between values of p_c for different classes is large, we obtain a priority scheme. A small value for p_c defines a low priority class, and a large value for p_c defines a high priority class. Low priority traffic will not be completely preempted by high priority traffic (therefore: '*quasi*'-priorities). However, values for p_c can be chosen such that the remaining low priority traffic in the presence of high priority traffic becomes arbitrarily small. Note that unlike a strict priority scheme, which preempts low priority traffic completely, the quasi-priority scheme can be implemented without additional hardware needs.

In Figure 11 we present the file transfer example for two priority classes. Station 1 belongs to the low priority class with $P_i^{max} = 1$, stations 2 and 3 belong to the high priority class with $P_i^{max} = 12$. When station 2 becomes active, it does not completely preempt the traffic from station 1. However, the throughput of station 1 decreases to 1/13 of the total bandwidth, and station 2 obtains the remaining bandwidth. After station 3 becomes active, stations 2 and 3 obtain the same bandwidth, and station 1 remains with 1/25 of the total bandwidth. When station 3 becomes idle, station 1 again obtains a fraction of 1/13 of the available bandwidth.

5 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 derived from our mathematical framework given in section 2. There we showed the uniqueness of a *strongly fair* and *waste-free* bandwidth allocation, i.e., a bandwidth allocation scheme limits the maximum throughput of each station and utilizes the entire bandwidth and. We described the $DQDB^{+/-}$ protocol in section 3.

We showed that at the cost of two additional bits in the slot header and few additional counters $DQDB^{+/-}$ 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 transmission delays of messages are reduced when 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 $DQDB^{+/-}$. In section 4 we extended $DQDB^{+/-}$ and could provide non-uniform bandwidth allocations, i.e., schemes that do not distribute the bandwidth equally among the stations. We showed examples of a per-user bandwidth allocation, a guaranteed bandwidth mechanism, and a (quasi-) priority scheme.

Acknowledgments

We would like to thank Dr. Chatschik Bisdikian for making available his simulator for dual bus networks.

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] M. Conti, E. Gregori, and L. Lenzini. A Methodological Approach to an Extensive Analysis of DQDB Performance and Fairness. *IEEE Journal on Selected Areas in Communications*, 9(1):76 – 87, January 1991.

[3] E. L. Hahne, A. K. Choudhry, and N. F. Maxemchuk. DQDB Networks with and without Bandwidth Balancing. *IEEE Transactions on Communications*, 40(7):1192 – 1204, July 1992.

[4] 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.

[5] 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.

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

[7] B. Mukherjee and C. Bisdikian. A Journey Through the DQDB Literature. Technical Report RC 17016, IBM Research Division, 1991.

[8] 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.

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

A Proof of Theorem 1

Let L^* denote the system of equations (1), (2), (3) and (4). We will prove the theorem after presenting the following lemmas.

Lemma 1 *L^* has a unique solution.*

Proof:
Without loss of generality we may assume that the station indices are ordered according to the arrival rates, i.e., $i < j$ if $\lambda_i \leq \lambda_j$. Note that in this case $i \in \mathbf{U}$ implies that if $j < i$ then $j \in \mathbf{U}$.

1. *Existence of solution:*
   A solution of L^* is obtained by defining the set $\tilde{\mathbf{U}}$ by:

$$\tilde{\mathbf{U}} = \{j \mid \lambda_j \leq \frac{1 - \sum_{i=1}^{j} \lambda_i}{n-j}\}, \tag{15}$$

   It can be easily verified that equations (1) and (2) are satisfied with $\mathbf{U} = \tilde{\mathbf{U}}$ and $\mathbf{O} = \mathbf{N} \setminus \tilde{\mathbf{U}}$. Note that the definition of $\tilde{\mathbf{U}}$ is constructive.

2. *Uniqueness of solution:*
   We will show that $\mathbf{U} = \tilde{\mathbf{U}}$. Let $\tilde{k}$ be the highest index in $\tilde{\mathbf{U}}$. Assume a strongly fair and waste-free bandwidth allocation with k the highest index in $\mathbf{U}$. Obviously, it holds that $k \leq \tilde{k}$. On the other hand, since $k+1 \in \mathbf{O}$:

$$\lambda_{k+1} > \frac{1 - \sum_{i=1}^{k} \lambda_i}{n-k} \tag{16}$$

   This is equivalent to:

$$\lambda_{k+1} > \frac{1 - \sum_{i=1}^{k+1} \lambda_i}{n-k-1} \tag{17}$$

   From equation (15), we obtain $k+1 > \tilde{k}$, and as a consequence, $k = \tilde{k}$. □

Lemma 2 *L^* defines a strongly fair and waste-free bandwidth allocation.*

Proof:
Clearly, L^* defines a bandwidth allocation. The condition for strong fairness is satisfied by definition of α^* in equation (2). To show that L^* defines a waste-free bandwidth allocation we have to distinguish two cases.

1. If $\sum_{i=1}^{n} \lambda_i \leq 1$, the unique solution for L^* is given by:

$$\mathbf{U} = \mathbf{N} \quad (18)$$

$$\mathbf{O} = \emptyset \quad (19)$$

   Then, $\gamma_i = \mathbf{min}\ \{\lambda_i\ , \infty\} = \lambda_i$, and $\sum_{i=1}^{n} \gamma_i = \sum_{i=1}^{n} \lambda_i$

2. If $\sum_{i=1}^{n} \lambda_i > 1$, the sum of the throughputs γ_i yields:

$$\sum_{i=1}^{n} \gamma_i = \sum_{i=1}^{n} \mathbf{min}\ \{\lambda_i\ ,\ \frac{1 - \sum_{j \in \mathbf{U}} \lambda_j}{|\mathbf{O}|}\} \quad (20)$$

$$= \sum_{j \in \mathbf{U}} \lambda_j + |\mathbf{O}| \cdot \frac{1 - \sum_{j \in \mathbf{U}} \lambda_j}{|\mathbf{O}|} \quad (21)$$

$$= 1 \quad (22)$$

□

Lemma 3 *Every strongly fair and waste-free bandwidth allocation is a solution to* L^**.*

Proof:
$\mathbf{U}$ and $\mathbf{O}$ are given by:

$$\mathbf{U} = \{j \mid \lambda_j \leq \alpha^*\} \quad (23)$$

$$\mathbf{O} = \{j \mid \lambda_j > \alpha^*\} \quad (24)$$

Applying the condition for strong fairness, we obtain equations (3) and (4). In order to show that equation (1) holds, we have to distinguish two cases:

1. If $\sum_{i=1}^{n} \lambda_i \leq 1$, then

$$\alpha^* > \mathbf{max}\ \{\lambda_i \mid 1 \leq i \leq n\} \quad (25)$$

   since the allocation is waste-free. In this case, $\mathbf{O} = \emptyset$ and with equation (2), $\alpha^* = \infty$. Therefore, equation (1) is satisfied with $\gamma_i = \lambda_i$ $(1 \leq i \leq n)$.

2. For $\sum_{i=1}^{n} \lambda_i > 1$ let us assume that $|\mathbf{O}| = m$. Note equation (1) is implied by the bandwidth allocation being strongly fair. Since the allocation is additionally waste-free, we have:

$$1 = \sum_{i=1}^{n} \gamma_i \quad (26)$$

$$= \mathbf{min}\ \{\lambda_i\ ,\ \alpha^*\} \quad (27)$$

$$= \sum_{\lambda_i \leq \alpha^*} \lambda_i\ +\ m \cdot \alpha^* \quad (28)$$

which gives

$$\alpha^* = \frac{1 - \sum_{\lambda_i \leq \alpha^*} \lambda_i}{|m|} \tag{29}$$

$$= \frac{1 - \sum_{j \in \mathbf{U}} \lambda_j}{|\mathbf{O}|} \tag{30}$$

□

Now we are ready to prove the theorem. Lemma 2 and Lemma 3 prove the equivalence of strongly fair and waste-free bandwidth allocations and solutions of L^*. Therefore, the uniqueness of a solution for L^*, shown in Lemma 1, implies that solving L^* provides the unique strongly fair and waste-free bandwidth allocation. □

B Proof of Theorem 2

Proof:
Given a strongly fair and waste-free bandwidth allocation. Assume $i \in \mathbf{O}$. Then, $\lambda_i > \gamma_i$ (equation (3)), and we obtain from equation (1):

$$\gamma_i \cdot |\mathbf{O}| = 1 - \sum_{j \in \mathbf{U}} \lambda_j \tag{31}$$

Using equation (7) this can be rewritten as:

$$\gamma_i \cdot (|\mathbf{O}_i| + 1) \;+\; \gamma_i \cdot (|\mathbf{O}| - |\mathbf{O}_i| - 1) = 1 - \sum_{j \in \mathbf{U}} \lambda_j \tag{32}$$

Because of equations (1), (2) and (4), $\gamma_j = \gamma_i$ for all $j \in \mathbf{O}$, and we obtain:

$$\gamma_i \cdot (|\mathbf{O}_i| + 1) = 1 - \sum_{j \in \mathbf{U}} \lambda_j - \sum_{j \in \mathbf{O} \wedge 1 \leq j < i} \gamma_j \tag{33}$$

Since $\gamma_j = \lambda_j$ for all $j \in \mathbf{U}$ (equation (3)) we obtain:

$$\gamma_i \cdot (|\mathbf{O}_i| + 1) = 1 - \sum_{j \in \mathbf{U} \wedge i < j \leq n} \lambda_j - \sum_{1 \leq j < i} \gamma_j \tag{34}$$

Equation (34) can be rewritten as:

$$\gamma_i = \frac{1 - \sum_{j \in \mathbf{U} \wedge i < j \leq n} \lambda_j - \sum_{1 \leq j < i} \gamma_j}{|\mathbf{O}_i| + 1} \tag{35}$$

From the definitions of Λ_i and Γ_i (equations (5) and (6)), and from $\lambda_i > \gamma_i$ (equation (4)) we obtain equation (8).

Now let $i \in \mathbf{U}$. If $|\mathbf{O}_i| = 0$, then $j \in \mathbf{U}$ for $j > i$, and we obtain:

$$\Lambda_i = \sum_{i<j\leq n} \lambda_j = \sum_{i<j\leq n} \gamma_j \tag{36}$$

Equation (8) then reads:

$$\lambda_i > 1 - \sum_{1\leq j<i} \gamma_j - \sum_{i<j\leq n} \gamma_j \tag{37}$$

Since $\lambda_i = \gamma_i$ (equation (3)) we obtain the following contradiction to Definition 1:

$$\sum_{1\leq j<n} \gamma_j > 1 \tag{38}$$

If $|\mathbf{O}_i| > 0$, we select k such that:

$$k = \min_j \{j \mid j \in \mathbf{O_i}\} \tag{39}$$

From $k \in \mathbf{O}$, we obtain with equation (35):

$$\gamma_k = \frac{1 - \Gamma_k - \Lambda_k}{|\mathbf{O}_k| \; + \; 1} \tag{40}$$

Since $j \in \mathbf{U}$ for $i < j < k$ we obtain:

$$|\mathbf{O}_i| \;\; = \;\; |\mathbf{O}_k| + 1 \tag{41}$$

$$\Gamma_i \;\; = \;\; \Gamma_k - \sum_{i\leq j<k} \gamma_j \tag{42}$$

$$\Lambda_i \;\; = \;\; \Lambda_k + \sum_{i<j<k} \gamma_j \tag{43}$$

Therefore:

$$\gamma_k = \frac{1 - \Gamma_i - \Lambda_i - \gamma_i}{|\mathbf{O}_i|} \tag{44}$$

This can be re-written as :

$$\gamma_k \cdot |\mathbf{O}_i| + \gamma_i = 1 - \Gamma_i - \Lambda_i \tag{45}$$

Since $\lambda_i = \gamma_i$ and $\gamma_i < \gamma_k$ equation (44) yields:

$$\lambda_i < \frac{1 - \Gamma_i - \Lambda_i}{|\mathbf{O}_i| \; + \; 1} \tag{46}$$

□

C Recovery from Transmission Errors

For a reliable operation of the $DQDB^{+/-}$ protocol it is important that plus bits and minus bits be received correctly. If the bit error rate of the buses is not negligible we have to consider cases where plus bits and minus are lost due to transmission errors. If a plus bit is lost the station that transmitted the plus bit will not receive any bandwidth. If a station does not receive a minus bits it will continue to reserve bandwidth for an overloaded station. In addition, lost plus and minus bits will result in inconsistent computations for the value of *quota* in equation (9).

The reliability of transmitting plus and minus bits can be arbitrarily increased by forcing stations to set multiple plus and minus bits. If a station changes its state from underloaded to overload it sets plus bits in a fixed number of K slots ($K > 1$). Likewise, if an overloaded station becomes underloaded it sets minus bits in K slots. Note that due to propagation delays a station may not be able to set the K plus bits (or K minus bits) in consecutive slots. Therefore, the station will attempt to set the plus bits (or minus bits) in the first available K slots.

Each station increments its ORQ counter whenever K slots with the plus bit set have been read. Likewise, a station decrements the ORQ counter whenever it reads K slots with the minus bit set.

In the following we discuss the protocol for detection of and recovery from transmission errors of plus and minus bits on bus B. Due to the symmetric nature of the dual bus architecture, the error protocol for transmission errors on bus A is identical.

The first station on bus A is responsible for detecting transmission errors of plus or minus bits on bus B. The station assumes that a transmission error has occurred if the following conditions hold:

1. The station reads a number of consecutive slots with the plus bit or minus bit set, followed by a slot with neither the plus bit nor the minus bit set.

2. The number of consecutive plus bits or minus bits that have been read is not an integral multiple of K.

If the conditions hold, the station sets the request bit, the plus bit *and* the minus bit in K consecutive slots on bus A, and then enters the *error recovery state for bus B.* A station in the *error recovery state for bus B* disregards all slots on bus B with either the plus or the minus bit set. In addition, the station is not allowed to set plus or minus bits on bus B.

All other stations enter the *error recovery state for bus B* when they receive K slots on bus A with the request bit, the plus bit and the minus bit set.

When the last station on bus A reads K' slots ($K' \leq K$) with the plus, minus and request bit set, followed by a slot that has not all three bits set, it sets plus and minus bits in K' consecutive slots on bus B. If the station is overloaded, it sets plus bits in the following K slots.

Now, each station that reads slots on bus B with both the plus and minus bit set, followed by a slot which does not have both the plus and minus bit set, will reset its ORQ counter to zero. Then, the station leaves the error recovery state, i.e., it will now increment the ORQ counter if K

plus bits are read on bus B, and decrement the ORQ counter if K minus bits are read on bus B. If the station is *overloaded* it will attempt to set K plus bits in the next slots on bus B.

The first station on bus A counts the number of slots with both plus and minus bits set. If the number of those slots is less than K the station assumes that a transmission error has occurred during the error recovery procedure. In this case, the station will initiate a new error recovery procedure. If K consecutive slots are read with both plus and minus bits set, the station assumes that the error recovery procedure was successful.

In Figures 12 to 16 we illustrate the steps of the error recovery procedure. We show a dual bus network with five stations. We assume that K is set to $K = 2$. [6]

[6] In Figures 12 to 16, slots are represented by three fields of the slot header, representing the plus bit, the minus bit, and the request bit. An empty field indicates that the appropriate bit is not set. A transmission error is indicated by a shaded area. Diagonal lines indicate that a station is in the *error recovery state*.

Figure 1: File Transfer with $DQDB^{+/-}$ (`basis` = 100).

Figure 2: File Transfer with IEEE 802.6 (`bwb_mod` = 8).

(Round-trip delay = 100 slots)

Figure 3: File Transfer with $DQDB^{+/-}$ (`basis` = 400).

Figure 4: File Transfer with IEEE 802.6 (`bwb_mod` = 8).

(Round-trip delay = 400 slots)

Figure 5: Mean Message Delay (Experiment I).

Figure 6: Mean Message Delay (Experiment II).

Figure 7: Mean Message Delay (Experiment III).

Figure 8: Mean Message Delay (Experiment IV).

Figure 9: $DQDB^{+/-}$ with *Per-User Bandwidth Allocation*.

Figure 10: $DQDB^{+/-}$ with *Guaranteed Bandwidth*.

Figure 11: $DQDB^{+/-}$ with *Quasi-Priorities*.

Figure 12: Station 1 detects a transmission error of a plus bit on bus B. It sets the plus, minus and request bit in $K = 2$ consecutive slots on bus A.

Figure 13: Station 3 enters the *error recovery state* when it reads $K = 2$ slots with the plus, minus and request bit set on bus A. In this state, station 3 ignores all slots on bus B with either the plus bit or the minus bits set. Station 3 itself will not set any plus or minus bits on bus B.

Figure 14: After receiving the shown slots on bus A, the last station on bus A (station 5) sets both the plus and minus bit in 2 consecutive slots on bus B. If station 5 is overloaded it sets $K = 2$ plus bits in the following slots.

Figure 15: Station 3 leaves the *error recovery state* if it reads 2 slots with both plus and minus bit set on bus B. If station 3 is overloaded it attempts to set $K = 2$ plus bits in the following slots.

Figure 16: The recovery procedure is completed when station 1 reads $K = 2$ slots on bus B with both the plus and minus bit set.

Table 1: Simulation Parameters.