Fairness Issues of the DQDB Protocol

Peter Martini

Department of Computer Science (Informatik IV); Technical University of Aachen
Ahornstr. 55; D-5100 Aachen; West Germany

E: +241/80-4522; FAX: +241/80-6295; e-mail: ...unido!rwthinf!martini

Abstract

The Distributed Queue Dual Bus (DQDB) is the draft of a proposed standard for Metropolitan Area Networks (MANs). In this paper we study fairness issues of the current version of this IEEE 802.6 working document which is supposed to be the basis for MAN service offered by public network providers. Performance is studied with respect to both delays and throughput. The paper shows that the throughput provided to specific users may depend on the geographic location with respect to other users. On the other hand, DQDB yields fair medium access for the majority of realistic scenarios.

I. INTRODUCTION

The Distributed Queue Dual Bus (DQDB, cf. [1]) Metropolitan Area Network (MAN) is a new standard proposed by the IEEE 802.6 Working Group on Metropolitan Area Networks (MANs). DQDB (originally known as QPSX for 'Queued Packet and Synchronous Exchange') defines

- a Dual Bus subnetwork with a shared medium physical layer of two unidirectional buses
- an arbitrated media access control sublayer based on Distributed Queueing; and
- a pre-arbitrated media access control sublayer for the support of isochronous services.

The purpose of DQDB is to enable public network providers to offer Metropolitan Area Network service. Therefore, the standard allows for bridges to interconnect two or more Dual Bus subnetworks to form a Metropolitan Area Network operating over areas in excess of 50 kilometers in diameter. Services to be supported include

1. High speed and high quality asynchronous packet communication service, for applications such as Local Area Network interconnection and workstation connection;

2. High speed and high quality isochronous service, for applications such as ISDN access and videoconferencing.

In contrast to the committees IEEE 802.3 (CSMA/CD), IEEE 802.4 (Token Bus) and IEEE 802.5 (Token Ring), the IEEE 802.6 committee was not restricted to a particular network architecture or a particular access method for the shared medium. Instead, it was left up to the committee to find the best choice. The road to consensus was heavily influenced by external events' (as James F. Mollenauer, the 802.6 chairman, put it [2]). However, the current DQDB protocol [1] proposed by Telecom Australia and its subsidiary QPSX Communications is supported by the great majority of the telephone companies in North America and Telecom Australia. Nevertheless, the current draft is not a complete standard and may be changed as a result of further work by IEEE 802.6.

This paper starts with a description of the DQDB network architecture and medium access protocol. After that, we study fairness issues of the DQDB medium access protocol. The paper shows that some users, because of their relative position in the network, may be able to capture a larger share of resources than others and thus enjoy preferential treatment.
II. NETWORK ARCHITECTURE

The DQDB standard allows for MAC level bridges to interconnect two or more Dual Bus subnetworks to form a Metropolitan Area Network where each DQDB subnetwork uses an architecture based on a pair of contra flowing unidirectional buses, known as a Dual Bus pair.

DQDB subnetworks can operate in either of the two topologies shown in figure 1: the open dual bus topology or the looped dual bus topology.

\[ \text{Open Dual Bus Topology} \quad \text{Looped Dual Bus Topology} \]

Fig. 1 Bus Topologies for DQDB

III. MEDIUM ACCESS CONTROL

A. Protocol Data Units

The fundamental DQDB MAC protocol data unit is the slot, which is a formatted set of octets (cf. Fig. 2). This slot structure is independently generated for each unidirectional bus by the slot generator at the head of that bus. It is transferred downstream along that bus, being available for access as it passes each node.
Each slot consists of an Access Control Field and a 'segment', where a segment is nothing but a protocol data unit transferred between peer entities. The Access Control Field (ACF) is used to control the reading of a segment from a slot and the writing of segments into a slot. The first bit of the ACF indicates whether the slot contains data (BUSY = 1) or the slot is available (BUSY = 0). In DQDB, there are two types of slots called 'queued-arbitrated' (Slot-Type = 0, used to transfer asynchronous segments) and 'pre-arbitrated' (Slot-Type = 1, carrying isochronous-sample segments), respectively. Access to these slots is controlled by strategies discussed in section III.B. This section also shows the meaning of the request bits.

The segment header includes a 16 bit label which is used for segment identification. The meaning of the remaining 16 bits of the segment header varies considerably for queued-arbitrated and pre-arbitrated slots. A discussion of these fields may be omitted since they are not relevant for the rest of the paper. Finally, each slot contains a 'segment payload' which is sixty-four bytes long.

B. Slot Access

It has already been mentioned above that the DQDB medium access protocol uses two types of slots called 'queued-arbitrated' and 'pre-arbitrated', respectively. Details of pre-arbitrated slot access are not relevant for the study presented in this paper. Therefore, we focus on queued-arbitrated slot access.

Queued-arbitrated Slot Access: Queued-arbitrated slots are used to transfer asynchronous segments. Access to this kind of slots is controlled by a function called 'Distributed Queue' which allows the formation and operation of a queue of asynchronous segments which is distributed across the subnetworks. In order to make this distributed queue perform as a centralized queue, each station maintains a dedicated counter for each of the contra flowing unidirectional buses (cf. Fig. 3, Fig. 4).

The busy bit in each slot (see above) indicates whether or not a slot is used. In contrast to this, the request bit is used to send a request that an asynchronous segment occupies a position in the distributed queue. Operation of the counters requires operations on both the busy and request bits. In the following we consider the access to bus A (the operation of queue formation on bus B is a mirror image of this case).

Nodes with no asynchronous segments queued to send on bus A count the number of nodes downstream on bus A which have registered requests at the node for access to bus A. However, the node only counts the requests which have not yet been satisfied. Each such request which is detected on bus B
causes the request counter to be incremented by one. In contrast to this, each empty slot passing on bus A causes the request counter to be decremented by one (provided it is not zero). The reason is that a previously queued segment will gain access to this slot.

Distributed queueing can support multiple levels of priority for access of asynchronous segments to empty, queued-arbitrated slots. This allows segments queued at higher priority to gain access before all segments queued at lower priority. A separate distributed queue, with separate counters, is operated for each level of priority. Each level of priority also requires a separate request bit in the access control field. The DQDB standard defines four levels of priority. The definition of how these levels are to be used is a matter for the operator of the particular DQDB subnetwork.

IV. FAIRNESS IN BANDWIDTH SHARING

In a light-load situation, all users of a communication system may be granted the requested bandwidth. However, in a heavily loaded communication system, requests have to be rejected. Thus, some kind of congestion control procedure has to be invoked to prevent additional traffic from entering the network and/or to distribute the limited bandwidth resources fairly among the users.

The ability to guarantee a given throughput/delay performance to the user is important in private networks. It is a *conditio sine qua non* in public networks where users may demand a constant quality of service (regardless of the network load). A fail-safe solution to unfairness is to offer private, dedicated bandwidth. In DQDB, this is achieved by pre-arbitrated slot access. The drawback of dedicated bandwidth is inefficiency. For that reason, DQDB is an integrated packet and circuit network, where the user can choose between queued-arbitrated service and pre-arbitrated service. The procedure of managing the allocation of dedicated bandwidth has not yet been addressed by the IEEE 802.6 Working Group. In the following, this aspect is left out of consideration. Instead, we focus on queued-arbitrated service.

The DQDB standard claims that 'the distributed queue is a function which allows the formation and operation of a queue of asynchronous segments which is distributed across the subnetwork, yet which gives the performance of a centralized queue. It guarantees minimum access delay and uniform sharing of capacity' (cf. [1]). What does uniform sharing of capacity mean? In a network with *n* stations it may mean that each station is granted access to every *n*th slot. As a matter of
fact, station 1 never transmits data to stations downstream on bus B (cf. Fig. 1). Likewise, station n never transmits data to stations downstream on bus A. Thus, uniform bandwidth sharing of each bus means unfair access to the network as a whole. On the other hand, the topologies presented in Fig. 1 show a logical separation between slot generators and stations transmitting and receiving data. In reality, operation as slot generator is nothing but an additional function of station 1 and station n, respectively. For that reason, station 1 and station n are identical in a looped dual bus topology. This kind of head end station gets fair access to the network if access to each bus is fair.

A. LIGHT-TO-MODERATE LOAD

Up to now, there are only few papers available dealing with the performance of DQDB. Of course, the main reason is that DQDB is a rather new protocol. In [6] Newman and Hullet presented a rather crude model of DQDB (originally known as 'QPSX' for 'Queued Packet and Synchronous Exchange'). The performance comparison included in [6] shows smaller packet delays for DQDB when compared to Token Passing or CSMA/CD. Similar results have been published in [7]. However, in [7] the authors do not provide any indication how the curves have been obtained. Since the authors discuss the superiority of a product developed by themselves, (justifiable?) doubts should be allowed.

Fig. 5 shows the average packet delay (waiting time for medium access + transmission time) for 25 km medium length, 25 nodes (spaced equidistantly along the medium), 100 Mbit/s total data rate, i.e. 50 Mbit/s on each bus, and uniform distribution of destination addresses. The load generated by each station has been varied from 0.1 Mbit/s to 2 Mbit/s where we assumed a model of interactive traffic (details included in [9]) which also included packet segmentation.

Obviously, results obtained for station 13 (located in the middle) and results obtained for station 1 (located next to the slot generator, cf. Fig. 1) differ significantly (intervals of confidence have been 2 % to 3 % of the mean values). Similar results have also been published in [8] for a totally different load model and 100 km medium length.

To get a more detailed insight into the correlation of geographic location and packet delay, Fig. 5 also shows the results obtained for each station where we assumed 0.1 Mbit/s and 1 Mbit/s generated load per station, respectively.

The reason for the different results presented in Fig. 5 is as follows: For light load, arrivals and 'service' by the medium may be modeled as shown in Fig. 6 where \( \lambda_i \) denotes the arrival rate at station i and \( \mu_A \) and \( \mu_B \) are the service rates (number of queued-arbitrated slots generated by the slot generator per time unit) of bus A and bus B, respectively.
Segments arriving at station $i$ are either destined to a station downstream on bus $A$ (arrival rate $\lambda_{i,A}$) or to a station downstream on bus $B$ (arrival rate $\lambda_{i,B}$).

![System Model for Light Load](image)

Assuming $\lambda_1 = \ldots = \lambda_n = \lambda$ (symmetric load), uniform distribution of destination addresses and numbering of stations as shown in Fig. 1, the arrival rates $\lambda_{i,A}$ and $\lambda_{i,B}$ may be calculated as

$$\lambda_{i,A} = \frac{n - i}{n - 1} \lambda \quad \lambda_{i,B} = \frac{i - 1}{n - 1} \lambda.

The queueing system shown in Fig. 6 is well-known from many applications where high server capacity is achieved by placing several low capacity servers in parallel and where the customer stream is split into components according to some rules. Each part is served by only one server. If queues form in such a system, the waiting time depends on how the queues are served. At one extreme, customers may change from one queue to another. In this case, $m$ servers of service rate $\mu_1$ behave like a single server with service rate $\mu = \mu_1 + \ldots + \mu_m$. All servers are kept busy unless all queues vanish.

In DQDB, the situation is different. Here, the 'customers' desire service from a specific server. Thus, in Fig. 6 customers (segments) may have to wait for service by Bus $A$ while the server representing Bus $B$ is idle. For light load, the probability that both buses (independent servers) are simultaneously busy is very small. Therefore, the successor of a customer already waiting for service of bus $A$ has a good chance to find bus $B$ idle. Stations located in the middle benefit greatly from this effect because their load is balanced between the servers. In contrast to this, all packets generated by stations located next to the slot generator are destined to the same bus.

In section IV.B, we show that the DQDB protocol may be unfair in the sense that saturated senders located at the head of the bus more often get free slots than any downstream saturated sender trying to transmit to this bus. This phenomenon is due to propagation delay causing differences in the time required for reservations. However, for lightly loaded systems and uniform distribution of destination addresses this effect is more than compensated by the fact that stations located in the middle only need half as much free slots on each of the two buses.

Of course, system performance is nothing but an answer to a dedicated load. Therefore, results would differ from the curves shown in Fig. 5 if different arrival processes would be assumed. However, for light load DQDB has to be expected to prove as a fair medium access protocol (all transfer times are very small). For this scenario, differences in transfer times may be ignored since the customer of a communication system generally does not become aware of delays in the order of milliseconds. If small delays are essential, better quality of service may be achieved by operating with priorities. In this case, unfairness is an intrinsic feature of priority scheduling.

**B. SATURATED SENDERS**

Any public communication system has to keep users from hogging all the bandwidth for themselves. In this section we study the fairness of DQDB with respect to this point by assuming saturated senders. Since the allocation of pre-arbitrated slots is not discussed in this paper we restrict our analysis to asynchronous traffic. In particular, we assume that the slot generator at the head of each bus does not generate any pre-arbitrated slots at all. The impact of this assumption is discussed in section V.

A network with saturated senders may be regarded as worst case of heavily loaded networks. If DQDB would not provide reservation mechanisms the two stations next to the slot generator would fill all slots with data, i.e. downstream stations would not have a chance to transmit data of their own. However, access to empty, queued-arbitrated slots is controlled.
by a distributed queue (cf. section III.). It has already been mentioned above that the DQDB standard claims that this distributed queue gives the performance of a centralized queue. The numerical evaluation of the model presented in the next section shows that reality may be different.

1) Network Model

In DQDB, access to empty, queued-arbitrated slots is controlled by 'request counters' and 'countdown counters'. Stations requesting access to bus A make upstream stations let pass empty slots by transmitting requests on bus B. As a matter of fact, stations may not transmit a new request before getting access to an empty slot. Due to propagation delays it takes some time to make all stations know that a specific station requires medium access. During this time, upstream stations act as if the downstream station had not requested medium access at all. Likewise, it takes some time for a slot left empty by all upstream stations to reach the station requesting access.

It is difficult to find a queueing system which is an exact model of DQDB. It would be much more difficult to analyze this model in an appropriate way. However, for our fairness analysis we may apply a rather crude model of bus access. Our model of access to bus A is shown in Fig. 7.

![Fig. 7 Model of Access to Bus A](image)

For very heavy load, each service by the central server (i.e. each slot transmission to a specific bus) is followed by a reservation. In Fig. 7, \( t_{r,i} \) denotes the average reservation time of station i, i.e. the time station i needs to make all stations know that it requests service by a specific bus. It should be noted that station n does not transmit to bus A. Likewise, station 1 does not transmit to bus B. Thus, a model of access to bus B may be obtained by replacing \( t_{r,1} \) by \( t_{r,n} \) and \( \lambda_1 \) by \( \lambda_n \).

For our analysis, the most important part of the reservation time is the propagation delay to and from the slot generator. Additionally, we have to take into consideration that medium access for both reservation and data transmission is based on a slot structure. Since slots for bus A and bus B are generated mutually independent, additional delays of one slot length (half a slot length for both reservation and transmission) have to be expected for each access. In the following, this delay is added to the propagation delay. It should be noted that the station next to the slot generator may use all slots which have not been reserved by different stations. Therefore, the average reservation time is given by

**Bus A:**

\[
\begin{align*}
\text{For } & i < n, \\
\end{align*}
\]

\[
t_{r,i} = 2 \frac{L_{A,i}}{V} + t_s
\]

**Bus B:**

\[
\begin{align*}
\text{For } & i < n, \\
\end{align*}
\]

\[
t_{r,i} = 2 \frac{L_{B,i}}{V} + t_s
\]

where \( t_s \) denotes the service time (slot transmission time) and \( L_{A,i} \) and \( L_{B,i} \) denote the distance between station i and the station next to the slot generator of bus A and bus B, respectively; the signal velocity is denoted by \( V \).

Of course, the throughput \( \lambda_i \) of station i is not determined by reservation times only. The other components are waiting time \( t_{w,i} \) for medium access (due to server congestion) and service time \( t_s \) (average slot transmission time). Obviously, the service time \( t_s \) is identical for all stations. In contrast to this, waiting times \( t_{w,i} \) may be different.
Since central service is provided by a deterministic server and (for saturated senders) reservation is deterministic, too, waiting times may vary decisively. With parameters and starting condition appropriately chosen, scenarios may easily be constructed where some stations are always served immediately while other stations always have to wait the maximum time possible. Of course, the maximum waiting time is \((n - 2) \cdot t_s\) since 'customers' are served according to a FIFO strategy and for each bus there is exactly one station (at the downstream end) which does not require access at all. This bound is sharp, i.e. it is in fact achievable (cf. e.g. [9]). On the other hand, this bound is sharp for specific stations (microscopically) only. In contrast to this, for reservation times larger than service times \(t_s\) the average waiting time taking into consideration all stations (macroscopic average) must be smaller than \((n - 2) t_s\).

For DQDB with saturated senders, it may be shown that busy stations only let pass slots which will be filled with data by downstream stations. Thus, for saturated senders the overall throughput (slots per second) is given by \(\lambda_S = 1/t_s\). The throughput for each station is given by \(\lambda_i = 1/(t_{r,i} + t_{w,i} + t_s)\).

This yields the following constraints for waiting times:

\[
\frac{1}{t_s} = \sum_{i=1}^{n} \frac{1}{t_{r,i} + t_{w,i} + t_s} \quad \text{for Bus A}
\]

\[
\frac{1}{t_s} = \sum_{i=2}^{n} \frac{1}{t_{r,i} + t_{w,i} + t_s} \quad \text{for Bus B}
\]

In the following we study the model shown in Fig. 8 for \(t_{w,1} = \ldots = t_{w,n} = t_w\). Since \(t_s\) and \(t_{r,i}\) are system parameters determined by the implementation, \(t_w\) may be calculated (e.g. iteratively) from the equations given above. This solution provides the station-specific throughput \(\lambda_i\) for the case that waiting times for the central server are identical for all stations. Of course, this solution is exact if reservation times may be ignored when compared to the time spent in the central server (FIFO-service). It is a good approximation if waiting times and/or service times are large when compared to reservation times. This is the case if there are many stations attached to the network and/or distances are small. Otherwise, it may be a good approximation or it may be not. On the other hand, we are interested in system characteristics due to reservation mechanisms. For that reason, we choose this solution for our numerical evaluation presented in section IV.B.2. Nevertheless, it should be kept in mind that there is no guarantee for the correctness of this solution. The solution is very close to simulation results which have been obtained by simulating the DQDB protocol for saturated senders (cf. [10]). These results have not been 'tuned' to fit with the analysis. The results are also very close to the results obtained by Wong (cf. [11]). But this only shows that the model shown in Fig. 7 is not wrong.

2) Numerical Evaluation

For our numerical evaluation we have to take into consideration that DQDB has been designed for Metropolitan Area Networks (MANs) where a MAN is composed of DQDB subnetworks interconnected by bridges. In order to keep the number of hops small, the interconnection of 'low speed' subnetworks (covering small areas) by 'high speed' backbone networks (covering large areas) is a reasonable approach. This approach also reduces routing complexity.

In this section we discuss fairness issues for both subnetworks and backbone networks. Thus, we have to study the impact of data rate and geographical distance. The data rate determines the service time \(t_s\). In contrast to this, the geographical distance determines the reservation time \(t_{r,i}\). Furthermore, we must take into consideration the impact of the number of stations. Of course, the throughput in bit/s is additionally affected by the packet length distribution. Segmentation of information packets may yield many segment payload fields only filled in part. In the following, this aspect is left out of consideration. Instead, we restrict our analysis to the number of slots which may be transmitted per second. In order to allow a comparison between our analysis and simulation results presented in [10] we assume a data rate of 50 Mbit/s per bus. Furthermore, we assume that stations are spaced equidistantly on the bus with station 1 and station \(n\) operating as slot generators. Fig. 9 and Fig. 10 show the impact of medium length on the throughput achieved by 5 saturated senders. The symbols show simulation results, the lines connect values obtained analytically by determination of waiting time \(t_w\). For 100 km medium length, station 1 may use bus A almost exclusively. This effect is decisively reduced for smaller distances. It should be noted that station 2 to 4 obtain almost equal shares of the remaining bandwidth. Of course, station 5 does not transmit to bus A.
With only 4 stations transmitting to one bus, waiting times for the central service (cf. Fig. 7) are small. For that reason, throughput is strongly affected by reservation delays. The situation is different for more stations requiring access to the common bus. This is shown in Fig. 11 where we show the throughput normalized to the throughput in the case of fair bandwidth sharing, i.e. normalized to \( (2 \cdot 50 \text{ Mbit/s}) / n \). The figure compares the throughput of a station located in the middle to the throughput of the station next to the slot generator and the throughput of a station operating as slot generator. Again, we assume uniform distribution of stations along the bus.

For 100 km bus length and a small number of stations attached to the network, throughput is heavily affected by the station’s geographic location. If only two stations are attached to each bus, both stations operate as slot generators. In this case, the protocol is fair since each slot generator may capture the whole bandwidth of one bus. With 3 stations attached to each bus, DQDB yields unfair bandwidth sharing. The situation gets even worse if more stations are added. The reason is as follows: Throughput of the slot generators decreases slightly with more stations attached to the network. On the other hand, stations attached far away from the slot generators do not really have a chance to capture a fair part of the capacity. Since the throughput shown in figure 11 is normalized to the throughput for fair bandwidth sharing (this throughput decreases for increasing \( n \)), the normalized throughput increases decisively.
Fig. 11 shows that the normalized throughput of the slot generator is upper bounded. This is due to the fact that waiting times for central service increase with an increasing number of stations. This effect slows down stations enjoying preferential treatment because of shorter reservation times. Finally, it yields fair bandwidth sharing of each bus which means unfair treatment for the slot generators which transmit to one bus only. Therefore, the normalized throughput of slot generators approaches 0.5 for $n \to \infty$.

From Fig. 11, it comes clear that stations located next to slot generators achieve more throughput than stations located in the middle. The normalized throughput of other stations is in-between (similar to the curve for 10 km medium in Fig. 10). These values are not shown in the figure.

With the number of stations increasing, all stations except the slot generator reach an equal share of the capacity. This is reflected in Fig. 11 by both the curve for the station located next to the slot generator and the curve for a station located in the middle getting close to 1 for an increasing number of stations. It has already been mentioned above that what seems to be an unfair treatment of slot generators is a fair treatment in the case of a looped dual bus topology where station 1 and station $n$ are identical.

The extension of the region of unfair treatment depends heavily on the medium length. This comes clear from Fig. 12 which shows the normalized throughput vs. the number of stations for 10 km bus length. Now, DQDB has to be called fair if at least 20 stations are attached.

V. CONCLUSION AND FURTHER WORK

In this paper we have studied fairness issues of the DQDB medium access control protocol. For light-to-moderate load, DQDB is somewhat unfair. However, even for stations handicapped by geographical distance, packet transfer times are very small. A comparison included in [12] shows that transfer times for the most handicapped station are decisively smaller than transfer times in FDDI-II (cf. [12]) which may be regarded as the most promising alternative when it comes to networking in a metropolitan area.

The fairness discussion for saturated senders has shown that DQDB may be called fair for the majority of realistic scenarios. On the other hand, DQDB is unfair in subnetworks interconnecting few stations and covering large distances. This has been shown by studying results obtained for 5 stations and 100 km bus length. Of course, this choice of parameters may be called unrealistic. On the other hand, this scenario shows the limits for backbone networks interconnecting few stations scattered over large areas. For this special application, DQDB is questionable.

Allocation of dedicated bandwidth (pre-arbitrated service) may be an appropriate solution. This approach will not result in much inefficiency if the number of stations requiring dedicated bandwidth is small. This strategy is currently studied by simulation at our institute. Our study also discusses unfairness due to more or less hops on the path between source and destination. This aspect is important since a metropolitan area network is to be formed by bridges interconnecting DQDB subnetworks.

Our simulation model may also be used for discussing the impact of pre-arbitrated service on fairness issues of queued-arbitrated service. Since pre-arbitrated slots carry requests for queued-arbitrated slots, a large amount of isochronous traffic reduces unfairness due to geographical disadvantages.
Another aspect which has to be studied is the impact of destination address distribution. In extreme cases, the distribution of destination addresses may force a station to direct all transmissions to one bus. This holds both for stations located in the middle of the bus and for stations located near slot generators.

References


