



IMPLEMENTATION OF BANDWIDTH BALANCING MECHANISM FOR DQDB NETWORK WITH MODIFIED PROTOCOL

Sayed Abd-El Hadi Nouh

Computer & Systems Department
Faculty of Engineering, Al- Azhar University
Cairo, EGYPT.

ABSTRACT

A potential problem with DQDB network is the achievement of fair sharing of bandwidth in terms of the location of the access node. For the sake of compensating this unfairness, the IEEE 802.6 Standard recommends to utilize a Bandwidth Balancing (BWB) Mechanism.

This paper deals with the implementation of BWB mechanism using the modified DQDB method. We focus on the behavior of this protocol when the network is heavy load. The effect of BWB mechanism on the fairness problems suffered by a DQDB network with two stations for three different scenarios is investigated analytically through using a graph model which is based on the space time diagram. Simulation is used when the number of active stations is larger than two. Our results provide a deep insight into the equilibrium access of data channel by all the active stations. They also show that the station throughputs are independent on the state of the network when overload occurs.

I. INTRODUCTION

The Distributed Queue Dual Bus (DQDB) protocol has been selected by the IEEE 802.6 working group as the preferred access method for Metropolitan Area Network (MAN). It is designed to operate in high speed networks spanning long distances. DQDB Network provides two types of service: the asynchronous transfer service (e.g. data packets) and the isochronous transfer service (e.g. voice and video). Based on a dual bus architecture, the DQDB Network provides for access of asynchronous traffic to the transmission medium by a distributed queue algorithm. The DQDB network consists of two slotted busses implementing unidirectional communications in opposite directions, as shown in Fig. 1. Nodes communicate with each other by transmitting fixed size segments in the proper direction. When a segment is sent in one direction, a corresponding request is sent in the opposite direction.

The request is used to notify nodes with prior access of the slots that a node is waiting for an empty slot. A node grants the request by refraining from transmitting in an empty slot and allowing it to be used by the node that made the request. The main advantages of DQDB are the simplicity of its access mechanism and the fact that it can utilize all of the channel capacity (on either bus) independent of the network size and the channel data rate. The performance obtained with the DQDB access protocol was shown to approach the perfect scheduled performance under favorable conditions [1]. However, many studies [2-4] have reported that the throughput of a node under overload traffic condition depends on its position and the time it first starts transmitting.



IMPLEMENTATION OF BANDWIDTH BALANCING MECHANISM FOR DQDB NETWORK WITH MODIFIED PROTOCOL

Sayed Abd-El Hadi Nouh

ABSTRACT

A potential problem with DQDB network is the achievement of fair sharing of bandwidth in terms of the location of the access node. For the sake of compensating this unfairness, the IEEE 802.6 Standard recommends to utilize a Bandwidth Balancing (BWB) Mechanism.

This paper deals with the implementation of BWB mechanism using the modified DQDB method. We focus on the behavior of this protocol when the network is heavy load. The effect of BWB mechanism on the fairness problems suffered by a DQDB network with two stations for three different scenarios is investigated analytically through using a graph model which is based on the space time diagram. Simulation is used when the number of active stations is larger than two. Our results provide a deep insight into the equilibrium access of data channel by all the active stations. They also show that the station throughputs are independent on the state of the network when overload occurs.

I. INTRODUCTION

The Distributed Queue Dual Bus (DQDB) protocol has been selected by the IEEE 802.6 working group as the preferred access method for Metropolitan Area Network (MAN). It is designed to operate in high speed networks spanning long distances. DQDB Network provides two types of service: the asynchronous transfer service (e.g. data packets) and the isochronous transfer service (e.g. voice and video). Based on a dual bus architecture, the DQDB Network provides for access of asynchronous traffic to the transmission medium by a distributed queue algorithm. The DQDB network consists of two slotted busses implementing unidirectional communications in opposite directions, as shown in Fig. 1. Nodes communicate with each other by transmitting fixed size segments in the proper direction. When a segment is sent in one direction, a corresponding request is sent in the opposite direction.

The request is used to notify nodes with prior access of the slots that a node is waiting for an empty slot. A node grants the request by refraining from transmitting in an empty slot and allowing it to be used by the node that made the request. The main advantages of DQDB are the simplicity of its access mechanism and the fact that it can utilize all of the channel capacity (on either bus) independent of the network size and the channel data rate. The performance obtained with the DQDB access protocol was shown to approach the perfect scheduled performance under favorable conditions [1]. However, many studies [2-4] have reported that the throughput of a node under overload traffic condition depends on its position and the time it first starts transmitting.

The node starts transmitting first can monopolies the channel. The unfairness is caused by the fact that an overload node will continuously transmit segments in one direction and request in the other. Nodes that are located downstream from the overloaded node will find all slots filled with segments and nodes upstream will receive a continues stream of requests. To solve the unfairness problem, an enhancement to DQDB protocol, the Bandwidth Balancing (BWB) Mechanism was proposed in [5] and has been chosen by the IEEE 802.6 Standard Committee to incorporate into the DQDB protocol to remedy the unfairness problem. The BWB mechanism limits the maximum bandwidth available to each node in order to leave a portion of bandwidth unused. This unused bandwidth permits the congestion on the network to relax and allows equal sharing of the available (remainder) bandwidth among the overloaded nodes.

This paper deals with the implementation of BWB mechanism using Modified DQDB method. This method is chosen in spirit to the standard DQDB protocol. The case of two overloaded nodes in the network is developed which completely specifies a graph model, which is based on the space-time diagram. The validation of this model is carried out through using the simulation results. In section II, the original DQDB access mechanism for asynchronous traffic is reviewed. Section III presents the concept of Bandwidth Balancing (BWB) mechanism. In section IV we present the implementation of BWB mechanism using the modified DQDB method. Section V presents the "graph model" which is based on the space-time diagram, for the case of two active (overloaded) nodes in the network, which completely specifies the nodal throughputs. In section VI the simulation results are obtained for various possible scenarios in the case of two and three active nodes in the network. Section VII concludes the paper.

II. DQDB MEDIUM ACCESS PROTOCOL for ASYNCHRONOUS TRAFFIC TRANSFER

DQDB [6] is based on two slotted unidirectional buses which carry traffic in opposite directions. Each node can read and write on both buses. Information is carried on slots of fixed duration generated by the head of each bus. The duration of a slot is equal to the size of a data segment of 44 bytes plus the header of 9 bytes. Each slot contains in its header a busy bit and three request bits, one for each level of priority supported by the DQDB medium access protocol. We are interested in operation at only one level of priority, and hence, will not discuss the behavior of the protocol under multipriority traffic. The busy bit of a slot indicates whether the slot is empty. The request bit is used for making reservations for data segments in opposite direction. Since the operation in both directions is identical, we will consider data transmission in one direction only. One of the buses will be called the data channel (bus A) and the other will be called the reservation (request) channel (bus B), as shown in Fig. 1. Flow of data slots is from upstream nodes to downstream nodes in the data channel. Requests are placed in the reservation channel in the opposite (upstream) direction. Each station has two counters called the Request Counter (REQ-CNT) and the Countdown Counter (CD-CNT). When a station is idle (there is no packet queued for transmission), only the REQ-CNT is operational. While in this state, the REQ-CNT is incremented by one for each request detected in the reservation channel and decremented by one for each empty slot detected in the data channel, as shown in Fig. 2. When a station becomes active (there is a packet queued for transmission), the content of the REQ-CNT is transferred to the Cd-CNT and the REQ-CNT is reset to zero.

Although an arriving packet may contain multiple segments, only one segment at a time is admitted per node into the network. For each segment registered, the station sends out a request in the reservation channel by setting to one the first free request bit received. Independent from this the CD-CNT is decremented by one for each empty slot detected in the data channel. After the CD-CNT reaches zero, the segment is transmitted in the first free slot observed. During the time the CD-CNT is active, the REQ-CNT is incremented for each request slot received from the downstream stations, as

shown in Fig. 3. It must be noted that, since access to both buses is independent, it may happen that a segment is transmitted before the corresponding request could be sent out. In such a case, a new segment is registered immediately, if available, and the REQ-CNT is transferred to the CD-CNT. A third counter counts the outstanding requests while it is allowed to have more than one outstanding request, data segments are still not admitted into the CD-CNT before the previous segment is transmitted. Several analyses for the effect of propagation delay on fairness in DQDB [7-10] demonstrated that in some cases under overload condition the throughput of the different stations is not equal, with unfair advantage to the station that starts overloading the network. Consequently, a fair access mechanism is essentially needed to ensure equal distribution of the available bandwidth among ready stations .

III. BANDWIDTH BALANCING (BWB) MECHANISM

To solve the unfairness problem, an enhancement to DQDB protocol, the Bandwidth Balancing (BWB) Mechanism was proposed in [1] and has been chosen by the IEEE 802.6 Standard Committee to incorporate into the DQDB protocol to remedy the unfairness problem. This method intentionally wastes a small amount of bus bandwidth called the idle capacity, but fairly divides the remaining bandwidth among the nodes, that is, all the lightly loaded nodes take exactly their needed bandwidth, and all the overloaded nodes take their bandwidth equally. So the throughput achieved by each overloaded node is the same, called the control rate (R).

The control rate R is implicitly communicated through the idle bus capacity; since each node can determine this quantity by observing the passing busy bits from upstream and request bits from downstream, control coordination across the system can be achieved. More specifically, the control rate R is proportional to the idle bus capacity, with a proportionality constant β (called BWB-Mode) > 1 , [1]. Assume that there are N active stations in the network which may require to send segments onto the same data bus. M of them are heavily loaded, and (M - N) are lightly loaded. Then :

$$R = \beta [1 - (S + M R)] \quad (1)$$

where S is the total utilization due to (M - N) lightly loaded nodes. Let $\gamma = R$ denotes the carried load of any heavy loaded node, then :

$$\gamma_n = \alpha [1 - \{ S + (M - 1) R \}] \quad (2)$$

$$\text{where } \alpha = \beta / (1 + \beta)$$

Hence equation 2 can be implemented by restricting each node from taking more than a fraction α of the spare capacity left over by the other nodes. In general, the control rate and the bandwidth wastage (idle bus capacity) are as follows :

$$\begin{aligned} R &= \beta (1 - S) / (1 + \beta M) \\ &= \alpha (1 - S) / [1 + (M - 1) \alpha] \end{aligned} \quad (3)$$

$$\begin{aligned} \text{Bandwidth Wastage} &= R / \beta = (1 - S) / (1 + \beta M) \\ &= (1 - \alpha) (1 - S) / [1 + (M - 1) \alpha] \end{aligned} \quad (4)$$

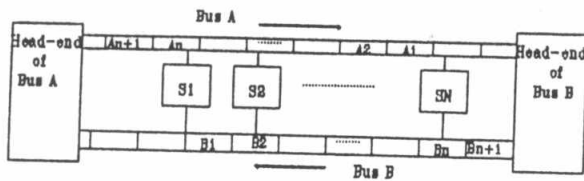


Fig. 1. DQDB network

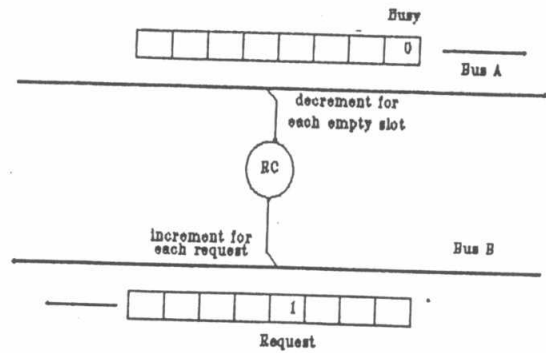


Fig. 2. Request counter operation for bus A; no asynchronous segments waiting

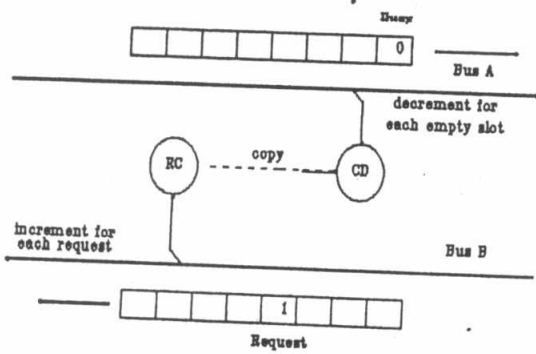


Fig. 3. Counter operation for bus A; asynchronous segments waiting

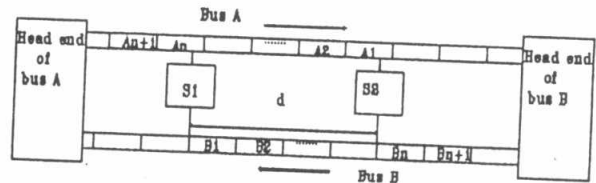


Fig. 4. The two-station case

IV. BWB IMPLEMENTATION USING THE MODIFIED DQDB METHOD

The modified DQDB method uses the same message format as DQDB protocol. In this method of implementation minor change to the DQDB protocol will be made. In DQDB, as explained in section II, a node can transmit a data segment whenever its Countdown Counter (CD-CNT) is zero and the data-bus empty slot is available. We propose instead that a node will be permitted to transmit only a fraction α of that time. This is achieved by artificially incrementing the request counter (REQ-CNT) by one after every β data segments transmitted, forcing the node to send an extra empty slot downstream. For example, if $\alpha=0.9$ (i.e., $\beta=9$), then after every 9 data segments transmitted, the node lets an extra slot pass. The node does not send any extra requests upstream corresponding to the extra empty slots it sends downstream. To implement this scheme, only one more counter, called bandwidth balancing counter (denoted BWB-CNT), is required at each node per bus. The operation of the BWB-CNT is explained as follows :

- (i) If BWB-Mode (β)=0, which indicates BWB disable, the BWB-CNT is idle and the access function is allowed to transmit in the normal way that was explained in section II.
- (ii) If BWB-Mode is not equal zero, which indicated BWB is enabled, the BWB-CNT is incremented by 1 every time the node transmits a segment to the bus as long as the value of the counter is less than β .
- (iii) When BWB-CNT value reaches β , the counter is reset to 0 and the REQ-CNT is incremented by one if no segment is queued. Otherwise, CD- CNT is incremented by one. This makes it happen that the access function to the bus is forced to skip transmitting the segment in the next available empty slot.

V. ANALYSIS OF TWO-STATION BWB-DQDB NETWORK USING THE GRAPH MODEL

In this section we investigate the effect of BWB mechanism, using the modified DQDB protocol , on the performance of heavy load two stations DQDB network, through using the "graph " model. This model is based on the space time diagram. Consider two stations transmitting on bus A and sending requests on bus B, see Fig.4. The station near head of bus A is designated as S_1 and the one further away as S_2 .

Consider the case in which S_1 was overloaded for some time (long enough to fill bus A with busy slots), and then S_2 received a long sequence of segments to transmit. This case is called 'Station 1 starts first'. Then, consider the opposite case in which S_2 has been transmitting continuously on every slot for a long period so that bus B is filled with requests, and only then S_1 starts its transmission. We call this case 'Station 2 starts first'. Finally, consider the case in which both stations become overloaded and start transmission at the same point in time. This is the case 'both stations start at once'. It is assumed that, the BWB-Mode is the same ($\beta=3$) in all the previous three cases.

Case 1: Station 1 Starts First

When S_1 becomes active and finds bus A busy, it immediately queues a request which is then written on bus B in the first arriving request bit. The corresponding space-time diagram is shown in Fig.5. Note that each received request at S_1 is satisfied by letting a free slot go by it, according to the original protocol. Moreover after each three successive transmission by S_1 (S_2), S_1 (S_2) lets a not requested empty slot go by, according to the BWB mechanism, where $\beta=3$. We define the interval between any two successive idle slots (I) as the access period, AP.

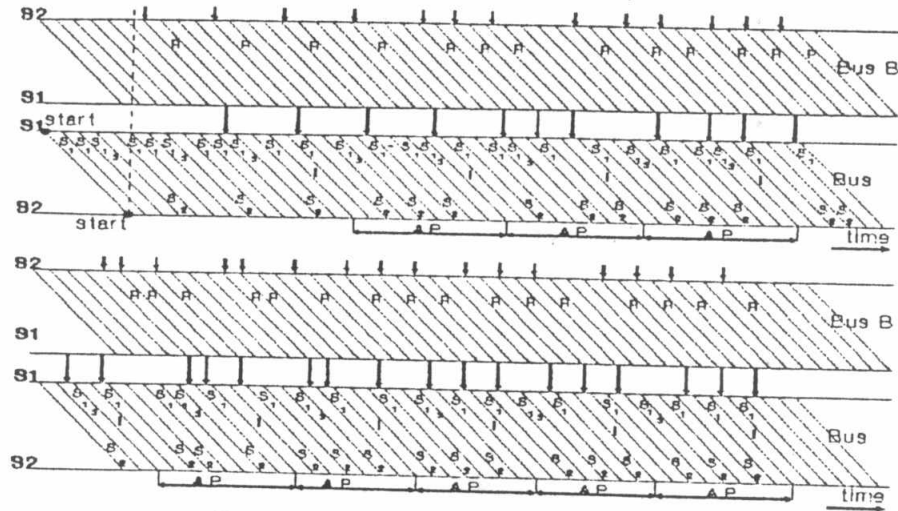


Fig.5. DQDB_BWB, Station 1 starts first

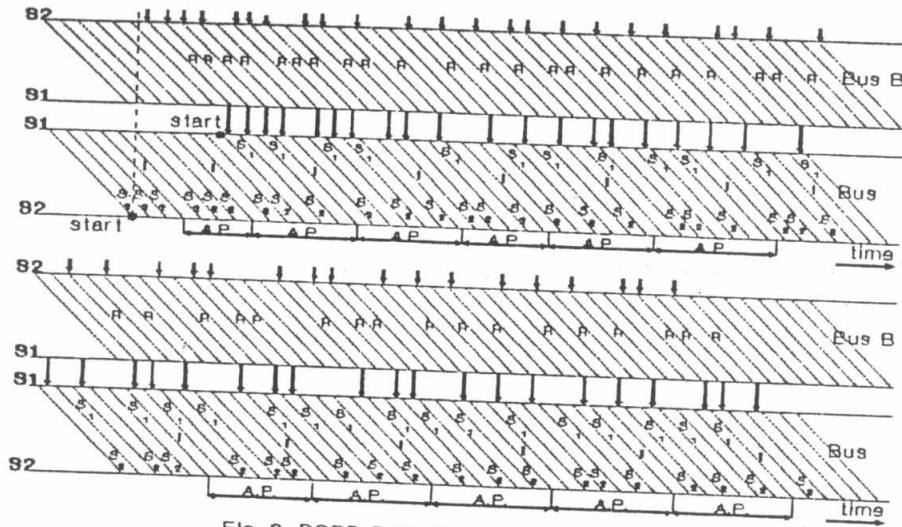


Fig. 6 DQDB_BWB, Station 2 starts first

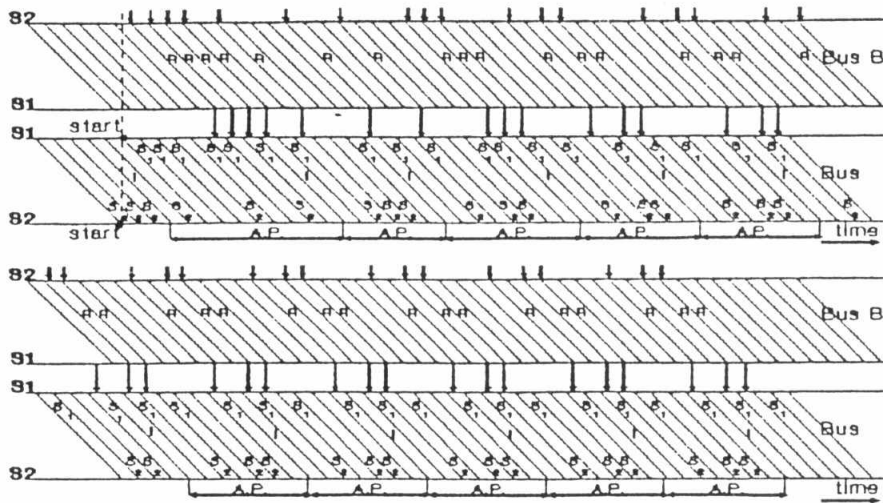
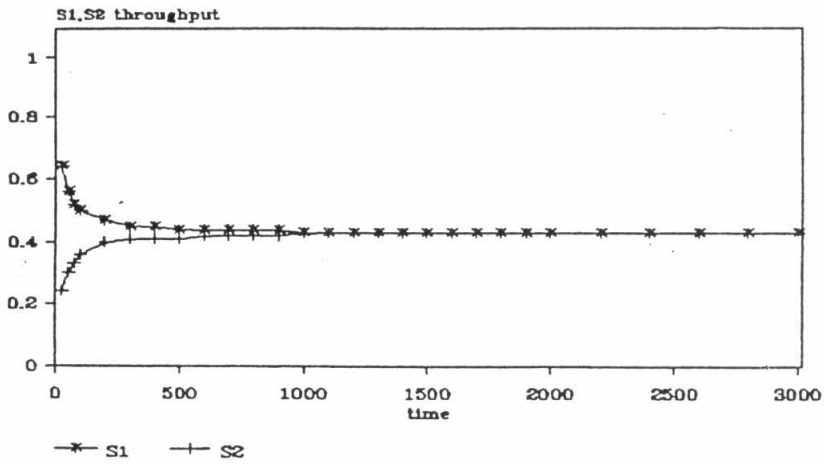


Fig.7. DQDB_BWB, both stations start at the same time



unit time in slot

Fig.8. S1 started 1st..S2 2nd.

The steady state throughput is achieved after a certain start-up period which can be divided into two phases. The first phase of that start-up period is the one in which the throughput of S_1 is much larger than that achieved by S_2 and no idle slots. The second phase consists of a number of access periods having variable lengths. In every access period the throughput of S_1 is smaller than that in the first phase, but still larger than the throughput of S_2 .

In the steady state, the access periods have a fixed and equal durations each of length $(2\beta+1)$ slots. In our case this length equal to 7 slots, where $\beta=3$, see Fig.5. The steady state throughput of Station 1 and 2 is the same and equal to β segments per one access period and the bandwidth wastage is one slot per one access period. For $\beta=3$, $\gamma_1 = \gamma_2 = 0.43$ and the bandwidth wastage =0.14. These results are consentaneous to that obtained from equations 3 and 4, where $S=0$ & $M=2$ and $\beta=3$.

Case 2: Station 2 Starts First

In this case, when S_1 becomes overloaded, it finds the request channel busy. This case is opposite to the previous one, in the start-up period the throughput of S_2 is higher than that of S_1 . As in the case 1, the steady state throughput of both stations is the same and equal to β segments per one access period of length $(2\beta+1)$ slots, see Fig. 6.

Case 3: Both Stations Start at Once

This case represents a situation in which both stations transmitting at once and find both buses empty. The steady state throughput of both stations is the same as in case 1 and 2, see Fig. 7.

Simulation Results :

Simulation results of BWB-DQDB network, using the modified method, were obtained, based on the 150 Mb/s line capacity for each bus, slot size equal to 53 bytes, propagation speed 200 000 Km/s, the distance between the two stations is 4 Km and BWB-Mod = 3.

The results for the above mentioned three cases (S_1 starts first, S_2 starts first and the both start together) are shown in Figs. 8, 9 and 10 respectively. These results are consentaneous to the graph analysis.

VI. SIMULATION OF THREE ACTIVE STATIONS

In this section we use simulation to examine the performance of the BWB-DQDB protocol, using the modified method, when there are three active stations.

Fig. 11 depicts simultaneous file transfers by three stations, with 5 slots (= 4km) between successive stations and with $\beta=9$. The plot shows the nodal throughput measured over 10 000 slots. The system starts in a totally idle state; the most upstream station (S_1) comes up first and almost immediately is able to achieve throughput of $\beta/(1+\beta)$. The most downstream station (S_3) turns on next and contends with the upstream station for a fair share of the bandwidth (each with $\beta/(1+2\beta)$). The middle station turns on next and the system again adjusts so that all three stations achieve equal throughput of $\beta/(1+3\beta)$. The most downstream station and then the middle station complete their file transfers, and in each case the system adjusts rapidly and redistributes the available

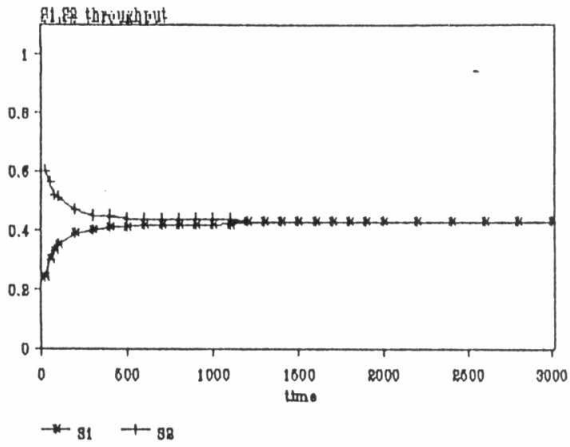


Fig.9. S2 started 1st., S1 2nd.

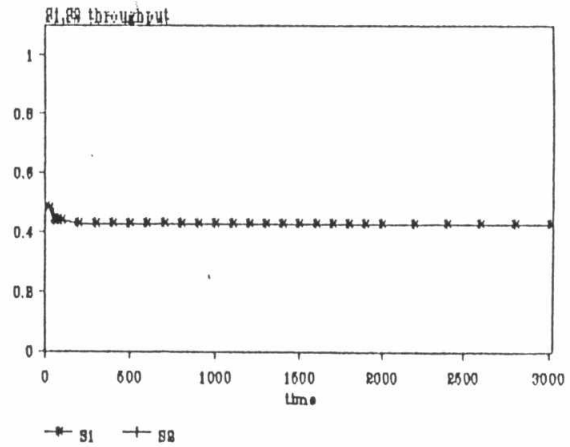


Fig.10. S1 & S2 start together

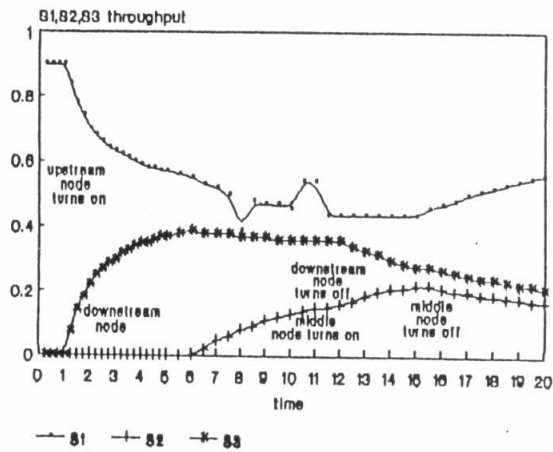


Fig.11. Throughputs for three stations with B=9
time in thousand

bandwidth equally. Note that the amount of wasted bandwidth decreases as the number of active stations increases.

VII. CONCLUSION

In this paper, a simple implementation of the BWB mechanism (the modified protocol) has been introduced that requires a very limited modification of the DQDB standard. A graph model based on the space-time diagram is developed. This model provides a deep insight into the behavior of BWB-DQDB protocol. The dynamic behavior of the system is presented through using simulation of three active nodes. The results show that the station throughputs are independent on the state of the network when overload occurs, but they only depend on the number of active stations and the BWB-Mode.

REFERENCES

- [1] R.L Newman and J.L.Hullet, "Distibuted queuing: a fast and efficient packet access protocol for QPSX", Proc. 8th ICCS, Munich 1986.
- [2] J.W. Wong, "Throughput of DQDB networks under heavy load", in EFOC/ LAN-89, Amsterdam, The Netherlands, PP. 14-16, 1989.
- [3] M. Conti, E. Gregori, and L. enzini, " DQDB media access control protocol. Performance evaluation and unfairness analysis", in 3rd IEEE workshop on MAN's, San Deigo, Calif., PP. 375-408, March 1989.
- [4] M. Zukerman and P. Potter, " The DQDB protocol and its performance under overload traffic conditions", in ITC Specialist Seminar, Paper No.164, Adelaide,Australia, 1989.
- [5] E.L. Hahne, A.K. Choudhury, and N. F.Maxemchuk, " Improving the fairness of distributed-queue-dual-bus networks", Proc. IEEE INFOCOM'90, San Francisco, June 1990, PP. 313-320.
- [6] IEEE 802.6 Committee, IEEE 802.6 standard: "Distributed Queue Dual Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN)", Institute of Electrical and Electronics Engineers, New York, Dec. 1990.
- [7] H. R.Van As, J.W. Wong, P. Zafiropulo," Fairness, Priority and Predictability of the DQDB MAC protocol under Heavy Load", Proc. Internal Zurich Seminar on Digital Comm., Zurich, Switzerland,Mar.1990, PP. 410-417.
- [8] M. Zukerman, P.G. Potter, " The Effect of Eliminating the stand by state on DQDB Performance under overload", International Journal of, Digital and Analog Cabled Systems, vol.2, 1989, PP. 179-186.
- [9] V. Phung and R. Breault, " on the unpredictable behavior of DQDB ", IEEE Contribution No. 802.6-90117, Irvine, March 1990.
- [10]P. Tran-gia and T. Stock," Approximate performance analysis of the dqdb access protocol", in ITC Specialist Seminar, Paper No. 16.1 (Adelaide, Australia), 1989.