

A Priority-Promoting Strategy to Improve the Fairness in DQDB Network

Tien-Yu Huang¹, Jean-Lien C. Wu²

¹Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan 10764

²Department of Electronic Engineering
National Taiwan Institute of Technology
Taipei, Taiwan 10772

ABSTRACT

In this paper we present a solution to improve the unfairness problem of distributed queue dual bus. This solution requires only a minor modification of the current protocol and wastes no bandwidth to resolve the unfairness problem. Simulation result shows that the proposed strategy has higher throughput and lower message delay than other strategies.

1. INTRODUCTION

The distributed queue dual bus is a communication network designed with a slotted access protocol that is becoming the IEEE 802.6 standard[5][7][9]. As the transmission rate and distance spanned by networks increase, slotted protocols can be much more efficient than token-passing protocols. However, some problems with unfairness in the DQDB networks have been identified[3][4][6]. It has been shown that with the current DQDB protocol, extremely unfair operating conditions can occur during overloads. When two nodes separated by 50km perform file transfers simultaneously, one station can obtain more than a hundred times the throughput of the other station, resulting in unfair delays.

To overcome the unfairness problem, some strategies have been proposed[4][8]. A Bandwidth Balancing (BB) strategy [4] is implemented by adding an extra counter (called a trigger counter) per station. Every time a station transmits a data segment, its trigger counter is increased by one. When the counter reaches a constant β , it is reset to zero and the request counter is incremented by one.

This strategy converges to a fair operating point when several nodes are performing large file transfers. But there are two shortcomings in the Bandwidth Balancing DQDB. First, it wastes some bandwidth to achieve fairness. The larger the waste is, the faster the convergence is. Second, the message delay of BB DQDB is longer than that of current DQDB. The Message Queue with Proportional Assignment (MP) Strategy [8] is a combination of Proportional scheme and (Multiple-Request) FCFS-Message-Based DQDB. In MP strategy, the bandwidths are allocated in proportional to the offered loads, but stations are also allowed to have multiple outstanding requests so as to maintain a current version of the "global message queue". Similarly, it suffers the same defects as Bandwidth Balancing DQDB.

In this paper we propose a Priority-Promoting (PP) strategy to resolve the unfairness problem and it has better performance than the other two strategies[4][8] in throughput and message delay.

2. Priority-Promoting DQDB

DQDB is of a dual-bus topology as shown in Figure 1. Each head of a bus generates continuous slots of fixed duration that travel along their respective buses. Each slot in DQDB contains a busy bit and a request bit. The busy bit indicates whether or not the slot is occupied by a packet, while the request bit is used for sending request for future packet transmission. When a station is idle, it keeps counting the number of requests from its downstream users via the request counter(RQ_CTR). The request counter increases by one when the station receives a

request bit and decreases by one for each empty slot in the forward channel. When the station becomes active, it transfers the content of request counter to count down counter(CD_CTR) and resets the RQ_CTR to 0. The station then sends a request in the reverse channel by setting the first unused request bit of the reverse channel. The count down counter decreases by one for every empty slot in the forward channel. When the count down counter reaches to zero, the station sends out the data segment into the first empty slot of the forward channel. Under DQDB protocol, a station would send a request for a packet transmission only when the corresponding packet reaches the top of the buffer. Since the relative frequency that a station successfully sets a request bit on the reverse bus determines the carried load of the station, that station's throughput can be lower if it is not allowed to signal all of its queued packets soon enough. This is especially true when overload occurs. The later version of DQDB allows stations to generate multiple outstanding requests when a station has multiple packets buffered, and each station is also allowed to have a request pending counter which keeps count of the number of buffered packets at the station whose corresponding requests have not yet been generated.

In our proposed Priority-Promoting DQDB, a station is allowed to generate the requests for all of its queued packets as soon as possible. When a station accepts a message of μ requests from downstream, it will send all of the μ requests as quickly as it can. Thus, the upstream station can really understand the traffic load of downstream stations. This scheme is implemented by using a Request-Pending Counters(RP_CTR) at each station. The counter is incremented by μ when a message of μ packets is accepted by the station. Each time a request is sent out, its counter is decremented by one. A station persists on sending requests as long as the contents of its Request-Pending Counter is nonzero. This strategy is followed in [5]. In our strategy, every station has a request counter and a passing counter but no count down counter. The request counter counts unserved requests from downstream nodes as in DQDB. When a station have any buffered packets to

be transmitted, it waits for an empty slot in the forward channel, then checks if its request count is zero. If its request counter is not zero, this means that there are buffered packets in the downstream station waiting for transmission, thus, it passes the empty slot to downstream stations, decreases its request counter by one, and increases its passing counter by one. The passing counter is used to indicate how many empty slots the active station has continuously passed to downstream stations. When the passing counter reaches to a constant value α , the station will promote its own priority to the highest. Thus it will send out its buffered packet to the first empty slot ignoring if its request counter is zero. The passing counter is reset to zero whenever the station sends out a packet successfully. The value α dominates the minimum available throughput of a station. Consider that there are N stations connected by a DQDB network, and each station performs large file transfer, thus each station tries to control all of the bus capacity. Because the system is overloaded and multiple requests are permitted, all the request bits of the reverse channel are always set. Thus, the request counters of upstream stations will never reach zero. If the control parameter of Station 1 is set to α_1 , it will pass

α_1 empty slots to downstream stations then use one slot. Thus, its throughput is $1/(1+\alpha_1)$.

Similarly, the throughput of Station 2 is equal to $a_1/((1+a_1)(1+a_2))$. It is easy to infer that the maximum throughput γ_n of Station n is

$$r_n = \begin{cases} \left(\prod_{i=1}^{n-1} \frac{\alpha_i}{1+\alpha_i} \right) \frac{1}{1+\alpha_n} & \text{for } n < N \\ 1 - \gamma_1 - \gamma_2 - \dots - \gamma_{n-1} & \text{for } n = N \end{cases}$$

If we set α_n to be the number of downstream stations, the throughput of each of the N stations is equal to $1/N$. If the traffic load of Station n is ρ_n which doesn't exceed $1/N$. The throughput S_n of Station n will be ρ_n . The flowchart of this Priority-Promoting strategy is shown as Figure 2.

3. Performance Comparisons and Conclusion

In this section, we compare the performance of Priority-Promoting (PP) DQDB to other modified DQDB protocol. In the simulations, each station is provided with multiple message buffers. The message arrivals at Station i follow an independent Poisson process with rate λ_i msg/slot, and message lengths of Station i are geometrically distributed with mean μ_i pkt/msg. For Bandwidth Balancing DQDB, the value of β is chosen to be 9. In Table 1, we show the mean message delay under different schemes with 5 active stations, and bus utilization equal to 0.9. The distances between stations are 100, 50 and 1 slots, respectively. Furthermore, we consider one of the five stations to be heavily-loaded with 9/10 of the total traffic, while the remaining load is equally shared by the other four stations. The location of the heavily-loaded station is moved from the most upstream position on bus to the most downstream position. Under BB strategy, the mean delay at the heavily-loaded station is much higher compared to the delay at the lightly-loaded station and the upstream stations have smaller delay compared to downstream stations. However, in MP strategy, the delay at the heavily-loaded station is slightly lower compared to that of the lightly-loaded stations and the upstream stations have smaller delay compared to downstream stations too. Whereas in PP strategy, the heavily-loaded stations have higher delay than lightly-loaded stations do, and the upstream stations also have smaller delay than downstream stations. From table 1, the delay is about 1000 slots in PP scheme, 2000 slots in MP scheme, 5000 slots in BB scheme and 1000 slots in current DQDB scheme. Thus, the delay time of PP scheme is much better than both BB scheme and MP scheme, and slightly better than current DQDB scheme. Table 2 shows the distribution of bandwidth among three active stations, when the upstream and downstream stations are involved in long file transfers and the middle station is a low-rate user, with Poisson arrivals. We find that the unfairness problem disappears. The low-rate user will not be

affected by the overloaded user. When the traffic load of low-rate user is 0.2 which is under 1/3, its throughput is 0.2. The remaining bus capacity ($1/3 - 0.2$) is shared by the downstream user whose throughput is 0.466667. When the traffic load of low-rate user is over 1/3, its throughput will be limited to 1/3. We also find that the total throughput in PP DQDB is equal to 1 which represents that the PP DQDB does not waste bandwidth to achieve fairness.

The current DQDB protocol can be extremely unfair during overload condition. Although some other strategies can resolve this problem successfully, they all waste some bandwidth to resolve the problem, and the speed of convergence depends on the cost it spends. This paper proposes another method which can successfully resolve the unfairness problem in overload condition and has a superior performance than other schemes.

REFERENCE

- [1] C.C. Bisdikian, "Waiting Time Analysis in a Single Buffer DQDB(802.6) Network", IEEE Journal on Selected Area in Commun., Vol. 8, No. 8, pp. 1565-1573, October 1990.
- [2] M. Conti, E. Gregori, and L. Lenzi, "DQDB Under Heavy Load: Performance Evaluation and Fairness Analysis", Proc. IEEE INFOCOM '90, San Francisco, CA, pp. 313-320, June 1990
- [3] J. Filipiak, "Access Protection for Fairness in a DQDB MAN", Proc., IEEE Intl Conf. Commun. '89, Boston, MA, pp. 635-639, June 1989
- [4] E.L. Hahne, A. Choudhury, and N.F. Maxemchuk, "Improving the Fairness of DQDB Networks", Proc., IEEE INFOCOM '90, San Francisco, CA, pp. 175-184, June 1990
- [5] IEEE Standard: "Distributed Queue Dual Bus (DQDB) Metropolitan Area Network (MAN)", p802.6, 1990
- [6] H. Kaur, and G. Campbell, "DQDB—An Access Delay Analysis", Proc. IEEE INFOCOM '90, San Francisco, CA, pp. 630-635, June 1990

- [7] J.F. Mollenauer, "Standards for Metropolitan Area Networks", IEEE Commun. Mag., Vol. 26, pp. 15-19, Apr. 1988
- [8] B. Mukherjee, and S. Banerjee, "Alternate Strategies for Improving the Fairness In and An Analytical

- Model of DQDB Networks", Proc. IEEE INFOCOM '91, Florida, pp. 879-888, Apr. 1991
- [9] R.M. Newman, Z.L. Budrikis, and J.L. Hullet, "The QPSX MAN", IEEE Commun. Mag., Vol. 26, pp. 20-28, Apr. 1988

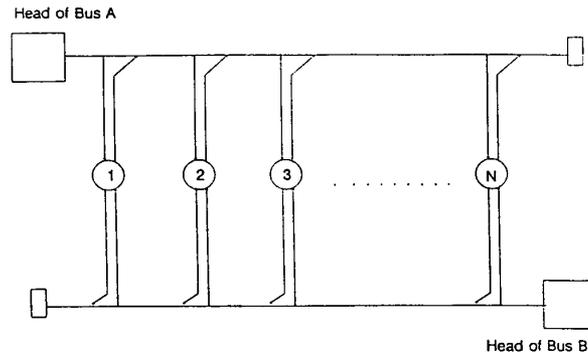


Figure 1 DQDB Network

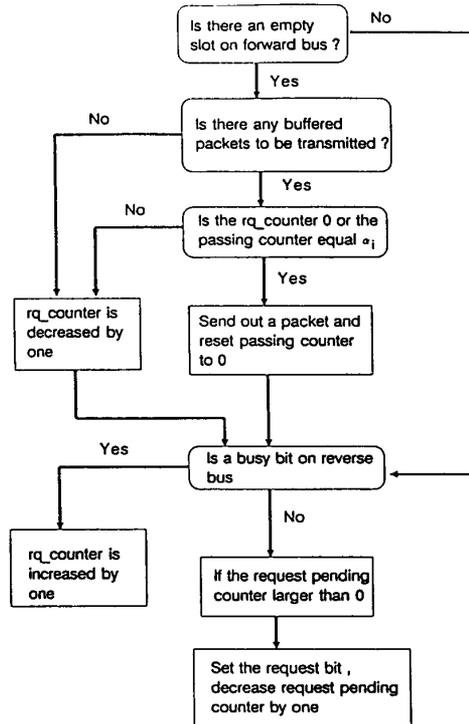


Figure 2 The flowchart of Priority-Promoting DQDB

STATION	DQDB SCHEME			BB SCHEME			MP SCHEME			PP SCHEME		
	INTERSTATION DISTANCE			INTERSTATION DISTANCE			INTERSTATION DISTANCE			INTERSTATION DISTANCE		
	100	50	1	100	50	1	100	50	1	100	50	1
1*	850	824	1074	5748	5010	5800	2175	2121	2010	1004	1002	995
2*	3509	3077	310	582	461	246	2501	2438	2267	420	386	428
3*	3933	3527	491	703	582	255	3001	2823	2788	583	490	536
4*	4110	3841	864	854	662	258	3222	3149	2987	725	582	574
5*	4745	4441	864	913	783	262	3216	3186	2988	885	713	621
OVERALL	1173	1115	1025	5251	4566	5241	2256	2198	2085	968	956	850
1*	1276	859	263	583	431	237	550	457	398	481	480	481
2*	861	853	1099	5743	4746	5517	2130	1938	1799	996	904	900
3*	4182	3845	336	582	461	244	3021	2820	2788	396	378	361
4*	4414	4091	479	705	571	238	3411	3020	3132	560	511	479
5*	4878	4523	824	838	662	250	2943	3028	2900	720	615	523
OVERALL	1143	1104	1032	5232	4317	4882	2166	1977	1850	950	844	838
1*	1791	1436	350	867	535	251	531	482	444	468	471	472
2*	1380	1034	298	543	431	229	504	478	451	401	401	401
3*	855	860	1083	6006	5132	5752	2234	2148	2081	990	988	986
4*	4708	4305	328	579	455	242	3108	2973	3019	381	401	398
5*	5081	4718	477	718	580	250	2985	3103	3187	583	471	341
OVERALL	1082	1060	1011	5488	4658	5198	2188	2108	2058	837	834	825
1*	2828	2184	367	781	607	253	531	478	488	458	457	483
2*	2211	1718	340	887	540	243	544	500	501	394	386	384
3*	1536	1082	298	548	427	238	547	488	518	300	298	298
4*	901	918	1110	6195	4888	5848	2310	2252	2267	988	986	983
5*	5191	4582	304	582	471	238	2580	2667	2435	382	310	227
OVERALL	1103	1088	1032	5636	4453	5375	2184	2130	2138	927	924	919
1*	3334	2548	391	852	648	250	504	508	483	451	448	453
2*	3314	2427	367	770	617	248	504	483	478	388	388	388
3*	2308	1771	342	686	533	238	518	483	480	298	298	217
4*	1848	1318	311	534	444	242	510	483	500	218	217	217
5*	930	988	1102	8087	4838	5853	2088	2227	2281	988	987	982
OVERALL	1102	1072	1027	5530	4488	5381	1838	2053	2101	924	922	918

Table 1 Effect of interstation distance on message delays in a five-station network (mean message length = 100 slots)

✕ Message arrival rate = 0.0081

† Message arrival rate = 0.000225

Offered load at the middle node	Throughputs			
	Upstream	Middle	downstream	Total
0.2	0.333333	0.2	0.466667	1
0.3	0.333333	0.3	0.366667	1
0.4	0.333333	0.333333	0.333334	1

Table 2 Throughput for Heterogeneous users