An Approach for Fairness Improvement in DQDB Networks

K. H. Liang and Y. C. Chen*

Department of Computer Science and Information Engineering
National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

Abstract

In this paper we discuss an approach for fairness improvement in Distributed Queue Dual Bus (DQDB) network, which is a draft standard for IEEE 802.6 Metropolitan Area Network (MAN). The DQDB medium access mechanism may be unfair for some realistic situations where both throughput and delay are dependent upon the geographic location of each node in the network. In this work, we analyze the asymmetric access property of DQDB and propose an approach called Reservation Capacity Priority Control (RCPC), which is both more efficient and fair for sending the segments. The RCPC allows the system to operate more uniformly and has a high degree of fairness in bandwidth sharing. The simulation results show that RCPC approach is more fair than Bandwidth Balancing Mechanism under all traffic conditions.

1. Introduction

Distributed Queue Dual Bus (DQDB) is a medium access control protocol proposed as part of IEEE 802.6 standard for Metropolitan Area Networks to interconnect local area networks, mainframes, and other high-speed devices [1,2]. The topology of a DQDB network consists of a pair of unidirectional slotted buses on which the traffic flows in opposite directions. The network stations are distributed along two buses and have the capability to transmit/receive information to/from both buses as shown in Figure 1. The DQDB medium access protocol is based on the QPSX (Queue Packet and Synchronous circuit eXchange) [3,4] protocol. It provides both asynchronous and isochronous services. The former is usually used to carry out the data services such as file transfer and electronic mail; while the latter is intended to support services that require fixed bandwidth and bounded delay, such as digital voice and video. Two access methods are used to support these two services: The Queued–Arbitrated (QA) access method, which has four priority queues for medium access arbitration. It uses fixed length slots for data transfer. Each priority level provides distributed queue access for supporting connectionless MAC service and connection-oriented data service. The Pre-Arbitrated (PA) access method, which uses assigned octet positions in particular slots for the transfer of individual octets of data. This method supports isochronous connection-oriented services. In this work, the Queued-Arbitrated slot access is of particular interest.

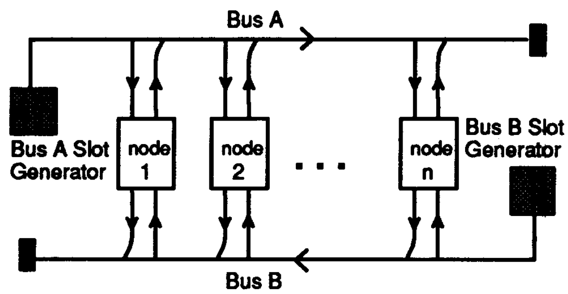


Figure 1. DQDB Topology

The QA access method employes the slot reservation approach to allocate the bandwidth. The method can achieve an aggregate throughput near the bus capacity, independent of the network size and speed. In addition, it offers minimal access delay by that each node can have at most one outstanding slot reservation. However, under the requirement of high bandwidth utilization at high speed and long distance transmission, this method may cause significant throughput unfairness.

The purpose of a computer network is for efficient resource sharing among many users. If a network service relies on extensive resource sharing, the interference and competition among users may occur, and this situation must be avoided. The solution is to design the network protocol properly so

* To whom all the correspondence should be addressed

that the resource sharing can be fair and users are able to get equal satisfaction. In [5,6,7,8,9], simulation analysis shows the unfairness problem in accessing the network with respect to the station positions on the buses. The current draft standard provides an optional Bandwidth Balancing Mechanism (BBM) to modify the access protocol so that the throughput unfairness can be improved. However, the BBM equalizes the throughputs only after a relatively long convergence time, resulting yet another substantial access–delay unfairness. In this work, we analyze the major factors which cause the unfairness of the basic DQDB. Next we attempt to solve the problem using our scheme. We model the network stations as a M/G/1–like queuing network and make an approximate average segment delay analysis for non–isochronous traffic. The simulation results are compared with that of BBM. In the following sections, the fairness analysis of basic DQDB is discussed in section 2, the proposed approach is discussed in section 3, the performance analysis for segment access delay is addressed in section 4, the fairness study of RCPC is addressed in section 5 and section 6 summarizes the work.

2. Fairness Analysis of Basic DQDB

From [5,6,7,8,9] we can find that basic DQDB is unfair. The problem is caused by the slot reservation mechanism, which is affected by the time delay needed to pass the status information between nodes. This situation results in different views of the distributed queue by different nodes. Due to some characteristics of the network such as the propagation delay, the network utilization, and the message type, a certain severity problems may occur. We will analyze the unfairness according to network traffic conditions. Two types of traffic conditions are considered: normal conditions and asymptotic conditions. Normal conditions occur when the offered load is lower (underload) or slightly higher (overload) than the medium capacity. Asymptotic conditions occur when each node in the network is trying to seize the whole medium capacity. Although both synchronous and asynchronous traffic may exist in DQDB network and there are 4 priorities classes [1,2] allowed, the unfairness problem occur only in asynchronous traffic and especially at the same priority level. In order to clearly discuss our study, we assume that only asynchronous traffic at the same priority class exists in DQDB network.

2.1 Fairness in Normal Conditions

To define the fairness quantitatively, we use the MAC (Medium Access Control) access delay as the parameter. The MAC access delay is defined as the time interval between the insertion of a segment into the distributed queue of an Access Unit (AU) and the beginning of its successful transmission. The interference needed to be considered only exists between users on different nodes. Therefore, a fair DQDB network usually means that all nodes must experience the same MAC access delay. We adopt the metrics defined in [8] to evaluate the fairness. The **variance** U of the MAC access delay is defined as:

$$U = \frac{1}{\overline{MAC}^2} \sum_{i=1}^{k} (MAC_i - \overline{MAC})^2$$

where MAC_i = Average MAC delay of Node i

$$\overline{MAC} = \frac{1}{k} \sum_{i=1}^{k} MAC_i$$

and the **difference** D between the maximum and minimum MAC delay is defined as :

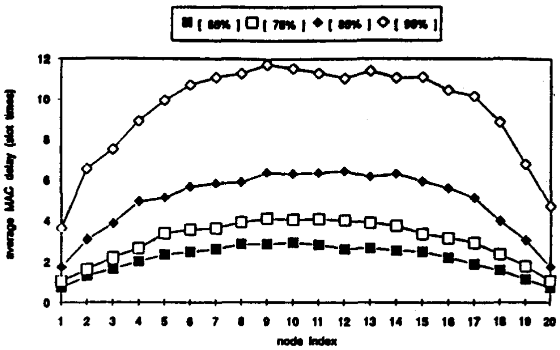
$$D = MAC_{max} - MAC_{min}$$

an ideal fairness network has $U = 0$ and $D = 0$.

Simulation results [10] show that when offered load is increased, the average MAC delay experienced by each node will increase. Also for multiple priority levels, the higher priority traffic is preemptive with respect to the lower priority traffic, therefore the former has smaller access delay than the latter. Figure 2 shows the average MAC delay at each each node for different levels of offered load, from the skew curves, we can find that the access delay experienced by each node varies with their position on both buses. As a consequence, we know that the unfairness occurs and the DQDB has fairness problem in normal conditions.

2.2 Fairness in Asymptotic Conditions

An asymptotic condition is the situation which causes the maximum interference between nodes. Since all active nodes compete for seizing as much medium bandwidth as possible. We use the throughput (i.e. the average number of packets transmitted in a unit of time) as the performance



[65%] : U = 2.11, D = 2.20
[75%] : U = 2.13, D = 3.01
[85%] : U = 1.90, D = 4.67
[95%] : U = 1.24, D = 8.13

Figure 2. Average MAC Delay vs. Node Positions at Different Levels of Offered Load

index which characterizes the DQDB asymptotic behavior. To make the analysis easy, we assume that only two nodes would be active in the network and no processing delay within the nodes. We also assume that once a node gets available slot to transmit a data segment, it is allowed to insert next data segment into the distributed queue and send the corresponding request.

Two cases may be used to demonstrate the asymmetric access property of DQDB: (1) An upstream node is initially active, then a downstream node becomes active. (2) A downstream node is initially active, then an upstream node becomes active. Both two cases can be generalized to the situation where more than two nodes are active simultaneously. From Figure 3, we are able to compute the time interval between two successive transmissions from node 2 :

$$X_{k+1} - X_k = \alpha + \tau + \beta + \tau + T$$

where X_k is the time at which node 2 transmits its K^{th} segment, α and β are the phase delay, τ is the propagation delay and T is the slot time. The fraction of the bandwidth B_2 used by node 2 is

$$B_2 = T / (\alpha + \beta + 2\tau + T)$$

and node 1 gets the rest of the bandwidth

$$B_1 = 1 - T / (\alpha + \beta + 2\tau + T)$$

Figure 4 shows that the larger the internodal length, the more the upstream node monopolize the bus. In the second case, the time interval between two consecutive transmissions from node 1 is

$$Y_{k+1} - Y_k = (1 + N_r) * T$$

where N_r is the number of requests received by node 1 during time interval (Y_{k-1}, Y_k). The simulation results also show that the larger the interval length, the more bandwidth the node 2 gets.

In [11], it is noted that if two nodes are very close together, or if they start transmission at exactly the same time, then each node gets half the bandwidth. However if the internodal length L is very large and node 2 starts much later, then the throughput rate of node 2 is only about $1/2L$. Node 1 has the same effect, it gets a roughly throughput rate $1/\sqrt{2L}$ when L is large and the node starts transmission much later.

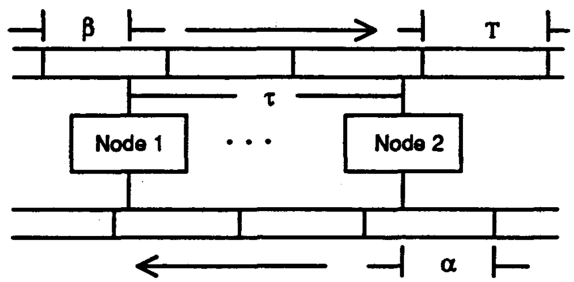


Figure 3. Two Active Node Configuration

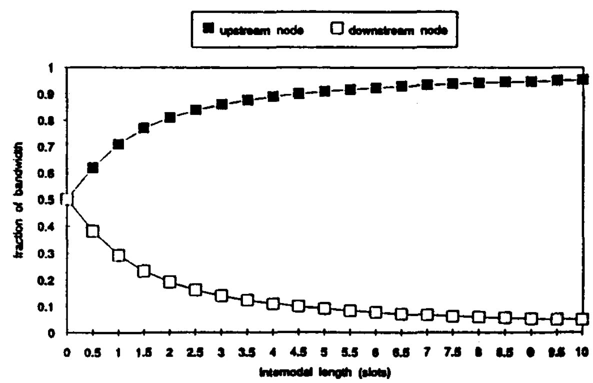


Figure 4. Fraction of Bandwidth vs. Internodal Length

3. The RCPC Approach

Our proposed RCPC mechanism is to add one more bit to the Access Control Field (ACF) and one counter to each bus for each priority level. Although the mechanism needs extra overhead, it always results in fair bandwidth allocation.

3.1 The Concept of the Approach

The ideal bandwidth fairness is that no bandwidth should be wasted, and nodes with sufficient traffic should all obtain the same throughput, called the controlled rate. In RCPC, downstream nodes use two bits in the slot ACF on request bus, say Bus y (y = B or A) to inform upstream nodes of the number of active downstream nodes. If a node becomes active and it knows that there are n nodes downstream which need to access a bus x (x = A or B respectively), it will defer to the n downstream nodes by allowing n empty QA slots to pass downstream on Bus x before it tries to send a segment. On the other hand, if a node is not active, it still has to detect all the information on Bus y to know the status information about the downstream nodes. In RCPC, the access scheme is modified to behave as a round–robin–like queuing system so that the overall distributed queue mechanism looks like a centralized queue with a round robin queueing discipline, therefore it is fair.

Figure 5 and 6 show the new Access Control Field and the REQUEST field & REQUEST–END field respectively. The REQUEST field indicates that a node has just received data from its user and it is prepared to request for transmission. The REQUEST–END field indicates that the current segment is the last one and the node will soon become inactive.

BUSY	SL_TYPE	RESERVED	PSR	REQUEST	REQUEST-END
(1 bit)	(1 bit)	(1 bit)	(1 bit)	(1 bit)	(1 bit)

Figure 5. New Access Control Protocol

REQ_3	REQ_2	REQ_1	REQ_0	END_3	END_2	END_1	END_0

Figure 6. Request Field & Request–End Field

3.2 Distributed Queue Operations

In our modified protocol, the Distributed Queue operates the request, countdown and preempt counters for each combination of priority level I (I = 0, 1,2 3) and Bus x (x = A or B). These counters are used to control the write access of Queued Arbitrated (QA) segments to empty QA slots on the corresponding bus.

The Distributed Queue State Machine (DQSM) has eight instances at each node, just like the original DQDB. The DQSM for access at priority level I (I = 0, 1, 2, 3) to bus x (x = A or B) can be in one of two states: idle state or countdown state. In both states, the DQSM should observe Bus y (y = B or A respectively) for REQ_I and END_I (I = 0, 1, 2, 3) bits in the ACF, which are set by nodes downstream on Bus y. The DQSM should also observe Bus x for empty QA slots, which are indicated by both the BUSY bit and the SL_TYPE bit of the ACF being set to zero. In [10], more detailed state machine operations are discussed.

The REQUEST_END Queue Machine (REQM) is to control the sending of request–end mark. Each DQSM on Bus x (x = A or B) is uniquely associated with REQM on Bus y (y = B or A respectively). Therefore each instance of the REQM queues request–end messages to be sent on Bus y for switching the DQSM from Countdown State to Idle State. The operation of REQM on END_I_Q_y counter is same as that RQM operates an REQ_I_Q_y counter in original DQDB. Whenever the END_I_Q_y counter is greater than zero, the REQM shall attempt to set the END_I bit on Bus y to one in the ACF of the first slot in which the END_I bit is set to zero, or to clear the REQ_I bit on Bus y to zero in the ACF of the first slot in which both REQ_I and END_I are equal to 1. When the attempt to send a request–end mark to upstream nodes is completed, the value of END_I_Q_y is decrement by one. If DQSM becomes active for Bus x again, it does not require that the REQM has sent request–end mark on Bus y. That is, the operation of writing request–end marks and sending QA segments are independent.

3.3 Protocol Operations

We assume single priority here to simplify the discussion. Suppose that there are n active nodes initially, and the node with largest subscript refers to the closest active node to the head of Bus x (x = A or B), while $node_1$ will refer to the farthest active node from the head of Bus x.

When $node_k$ becomes active it sends a request mark on request bus to join the active group and starts sending its segments immediately. The request mark takes $t_{k,k+1}$ time to reach $node_{k+1}$. During the period $t_{k,k+1}$, $node_1$ through $node_k$ will share a fraction (k – 1) / n of the total bandwidth. As $node_{k+1}$ receives the request mark, it allows an additional empty QA slot to pass on Bus x (x = A or B). Meanwhile, from $node_1$ through $node_{k+1}$, including the new active node, will share a fraction k/n of the total bandwidth which was available to them before $node_k$ turns to be active. After $t_{k+1,k+2}$, the request mark reaches $node_{k+2}$ and $node_1$ through $node_{k+2}$ will share a fraction (k + 1) / n of the total bandwidth at that time. This process continues until the request mark reaches $node_n$. Finally, all the n+1 nodes have equal sharing of the total bandwidth.

When $node_k$ has transmitted the last segment in its local buffer on Bus x (x = A or B), it sends a request–end mark on request bus y (y = B or A respectively). The request–end mark take $t_{k,k+1}$ to reach $node_{k+1}$. During the period $t_{k,k+1}$, $node_1$ through $node_{k-1}$ share a fraction 1/n of the bandwidth originally allocated to $node_k$, in addition to their own normal share of the bandwidth. During the next period $t_{k+1,k+2}$, $node_1$ through $node_{k-1}$ and $node_{k+1}$ will share the bandwidth originally allocated to $node_k$. Finally, the request–end mark reach $node_n$ and the remaining n – 1 active nodes have equal shares of the network bandwidth.

4. Performance Analysis for Segment Access Delay

To model a DQDB, we concentrate our attention on only one bus and ignore the synchronous traffic (PA slot). Assume that segments arrive for transmission at each transmitting node according to independent Poisson processes. Let λ be overall arrival rate to the system, since the segment transmission time is deterministic and segment transmissions can start only at times m, 2m, 3m, . . . , i.e., at the beginning of a slot of m time units, we can model the idealized DQDB as M/G/1 queue with vacations system. The server is considered as a deterministic slotted server. When there are no segments in the distributed queue at the beginning of a slot, the server takes a vacation for one slot, or m time units. Thus if we have the average service time $\overline{X} = m$, the second moment of service time $\overline{X^2} = m^2$, the first moment of vacation interval $\overline{V} = m$, and the second moment of vacation interval $\overline{V^2} = m^2$, from [10] we can have

$$W = \frac{\lambda m^2}{2(1-\lambda m)} + \frac{m}{2}$$

for preemptive priority queuing, we have

$$W_k = \frac{m^2 \sum_{i=1}^{n} \lambda_i + m(1 - m\lambda_1 - m\lambda_2 \ldots - m\lambda_n)}{2(1 - m\lambda_1 \ldots - m\lambda_{k-1})(1 - m\lambda_1 \ldots - m\lambda_k)}$$

The modeling and performance analysis of the DQDB network is known to be a hard problem. The reason is due to the high degree of interactions among a lot of processes. This situation makes an accurate analysis of the network almost impossible. The idealized model discussed here allows us to analyze a complex system. From the last two equations, we are able to get the MAC access delay which is defined in section 2.

5. A Fairness Study of RCPC

In this section, we use simulation results and fairness metrics mentioned in section 2 to make the comparisons between BBM mechanism and RCPC mechanism in both normal and asymptotic conditions.

5.1 The Simulation Model

In our simulation model, no node has any Pre–Arbitrated traffic. Each node generates segments for transmission according to a Poisson distribution and the segments are independent. The amount of traffic generated by each node on a particular bus is proportional to the number of downstream nodes on that bus. A new segment arriving at a node will be lost when the number of segments awaiting transmission (i.e. the segments waiting in the local node queue plus the one already inserted in the distributed queue) is equal to a given threshold. (In the following, this threshold will be referred to as the buffer size.) The parameters shown in Table 1 are used in our discussion.

5.2 A Comparison to BBM

BBM is an optional scheme which idles a fraction of medium capacity to get fair bandwidth sharing among nodes. It can only operates at a single priority level. The time that DQDB network converges to fair access operation depends upon the fraction of slots that the network is willing to waste. The larger the waste, the faster will be the convergence speed. The BBM mechanism involves a bandwidth modulus parameter which arbitrates the fraction of bandwidth to be wasted and determine the medium utilization in the steady state.

Figure 7and 8 show the average MAC delay versus nodes for BBM and RCPC mechanisms in overload conditions. For underload conditions, the result is similar [10]. Figure 9 and

* Internodal length = Length of bus / N
* Capacity of each bus C = 150 Mbps
* Slot size S = 53 octets
* Slot duration = 2.83 μsec
* Length of each bus = 57 slots
* Bus latency (t) = (2.83 * 57) μsec
* Total number of nodes N = 20
* Buffer size B = 10 segments

Table 1. Network Parameters

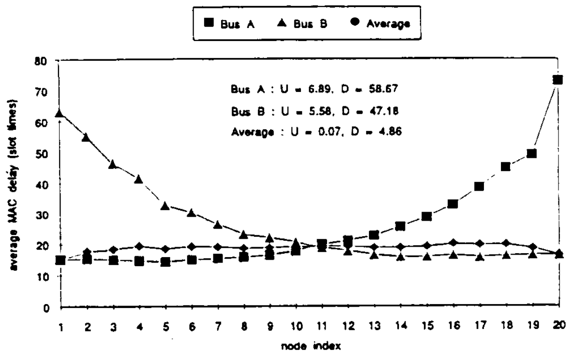


Figure 7. Average MAC Delay vs. Node Position with Offered Load = 105% (BBM)

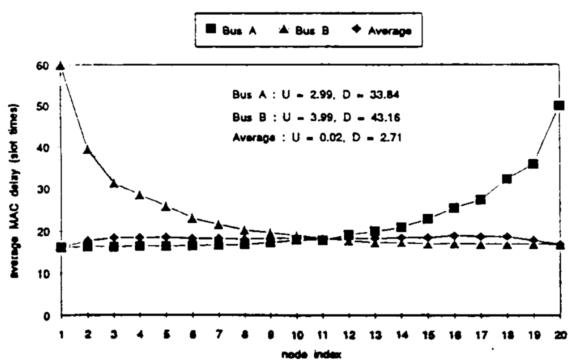


Figure 8. Average MAC Delay vs. Node Position with Offered Load = 105% (RCPC)

10 give the simulation results of BBM and RCPC mechanisms at four different levels of offered load. Figure 11 depicts the performance behavior of the three different protocol versions (DQDB, BBM, and RCPC) in terms of MAC delay in overload condition. We find that the access delay of RCPC is slightly higher than DQDB with BBM enabled or disabled in underload conditions. However, when the offered load is over the medium capacity, the RCPC has lower access delay and that delay is nearly constant for all nodes, as shown in Figure 11. From the values of fairness metrics (U and D) depicted in Figure 7 – 11, we know that the values in RCPC are smaller than these values in BBM. In other words, RCPC is more fair than BBM in normal conditions, because it has the characteristics that both U and D are nearly zero.

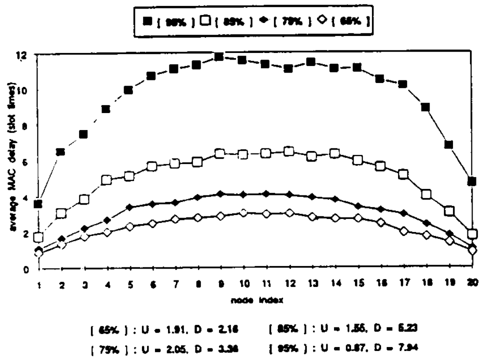


Figure 9. MAC Delay vs. Node Positions (BBM)

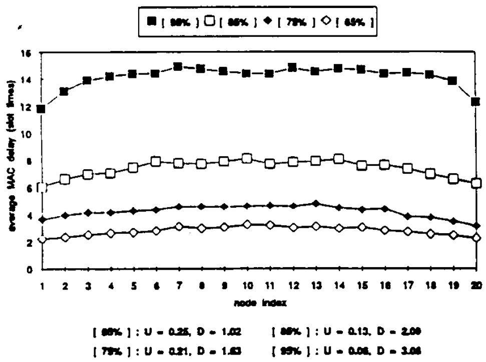


Figure 10. MAC Delay vs. Node Positions (RCPC)

In asymptotic conditions, Figures 12 and 13 exhibit the behavior of BBM and RCPC under the same scenario. The two figures show the throughput convergence during simultaneous file transfer of equal priority by three nodes. In Figure 12, the node throughputs approach their steady state gradually. The BBM equalizes node throughputs only after a relatively long convergence time and its total throughput is always less than one. While from Figure 13, the RCPC makes an equal sharing of bandwidth only within a round trip delay and the total throughput is near or equal to one all the times.

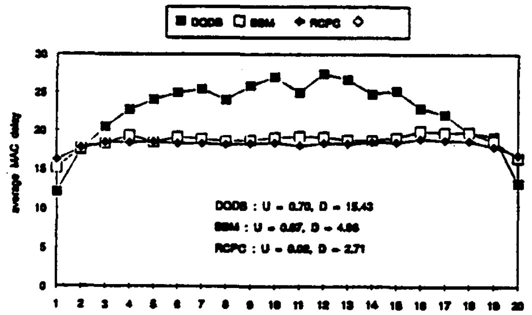


Figure 11. MAC Delay vs. Node Positions with Offered Load = 105% Capacity

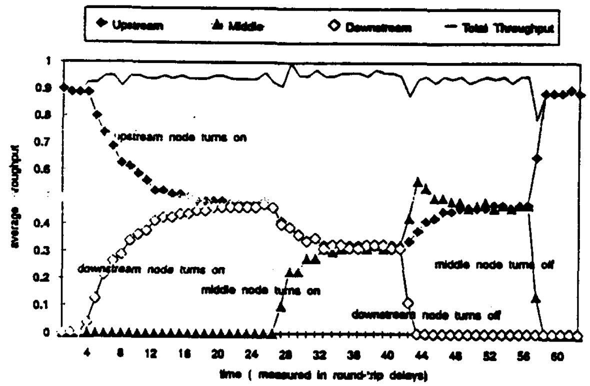


Figure 12. Throughputs for Three Nodes (BBM)

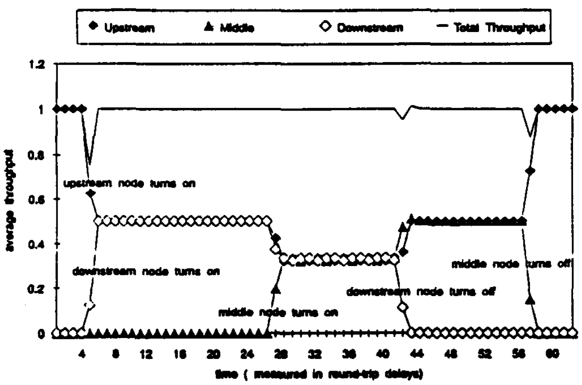


Figure 13. Throughputs for Three Nodes (RCPC)

6. Summary and Conclusions

A fully distributed bandwidth balancing mechanism RCPC for solving the fairness problem in original DQDB has been discussed. The mechanism allows the distributed queuing control to behave like a centralized queue with a round robin discipline without losing the simplicity, low overhead and high utilization characteristics of the DQDB. This is accomplished by providing each node with status information about downstream nodes. A little modification to the DQDB draft standard is required for the implementation. The simulation is performed using M/G/1 queuing discipline with vacations system.

The RCPC mechanism has high degree of fairness in the bandwidth sharing. A comparison between BBM and RCPC shows that RCPC is more fair than BBM under any traffic conditions. However, the RCPC has potential reliability problem if some error occurs in access control field. It may need a monitor process to reset the counters automatically whenever some error conditions are detected. Our future work is to remove these two drawbacks in RCPC for improving both efficiency and reliability.

References

[1] IEEE (1989) "Proposed Standard Distributed Queue Dual Bus (DQDB) Metropolitan Area Network (MAN), P802.6/D9

[2] IEEE (1990) "Proposed Standard Distributed Queue Dual Bus (DQDB) Metropolitan Area Network (MAN), P802.6/D12

[3] Z. L. Budrikis et al., "QPSX: A Queue Packet and Synchronous Circuit Exchange", Proc. ICCC'86, Munich, pp. 288–293

[4] R.M. Newman and J. L. Hullet, "Distributed Queueing: A Fast and Efficient Packet Access Protocol for QPSX", Proc. ICCC'86, Munich, pp. 294–299

[5] H. Kaur and G. Campbell, "DQDB – An Access Delay Analysis", INFOCOM'90 Conf. Proc., June 1990

[6] J. W. Wong, "Throughput of DQDB Networks Under Heavy Load", Proc. of EFOC/LAN, Amsterdam, June 1989

[7] M. Conti, E. Gregori and L. Lenzini, "DQDB/FBS: a Fair MAC Protocol Stemming from DQDB Fairness Analysis", CNUCE Report C89–28, October 1989

[8] M. Conti, E. Gregori and L. Lenzini, "DQDB Under Heavy Load: Performance Evaluation and Unfairness Analysis", CNUCE Report C89–27, October 1989

[9] M. Conti, E. Gregori and L. Lenzini, "Asymptotic Analysis of DQDB", Proc. of EFOC/LAN'90, June 1990, Munich

[10] K. H. Liang and Y. C. Chen, "An Approach for Fairness Improvement in DQDB Networks", Thesis, CSIE Department, National Chiao Tung University, June 1990, Hsinchu, Taiwan, R.O.C.

[11] E. L. Hahne et al., "Improving The Fairness of Distributed Queue Dual Bus Networks", Proc. of INFORCOM'90., Jun 1990, San Francisco