

A Fair and Efficient Integrated Protocol for a Full-Duplex ATM Ring Network*

Ho-Ting Wu

Department of Electronic Engineering
National Taipei Institute of Technology
No. 1, Sec. 3, Chung-Hsiao E. Rd.
Taipei 10643, Taiwan
htwu@en.tit.edu.tw

Abstract

We propose an integrated algorithm to jointly support the transmission of synchronous and asynchronous traffic across a full-duplex ATM ring network which employs a destination removal procedure. This algorithm is an extension of the FECCA algorithm introduced in [1]. A framed approach is employed. Each frame consists of a "synchronous subframe" followed by an "asynchronous subframe". Each node can prescribe a synchronous quota in each frame, specifying the maximum number of synchronous connections it wishes to support per frame. The reverse ring is used for the transmission of signaling information. As a result, the integrated scheme enables the occasional transmission of asynchronous data within the synchronous subframe and permits extra cell transmissions beyond specified quotas. Thus, this integrated scheme provides guaranteed bandwidth for the support of a quota of synchronous connections while at the same time is able to efficiently (and in a fair manner) utilize the remaining bandwidth to support asynchronous traffic. Illustrative performance examples are shown to demonstrate the effectiveness and efficiency of this integrated algorithm.

1 Introduction

The integrated support of synchronous and asynchronous traffic streams by a single network has received considerable attention in recent years. These two types of traffic streams have different service requirements: Synchronous (or real-time) traffic connections require guaranteed bandwidth and bounded delay, while asynchronous (or bursty) message streams expect very low error and loss rates. Both of them will be accommodated in a broadband

networking environment. Therefore, it is very important to design an integrated algorithm on a single network such that these two types of traffic streams can share the transmission bandwidth efficiently while being provided with an appropriate grade of service. Many integrated protocols have been proposed for ring [2, 3, 4] or bus [5, 6, 7] networks.

In this paper, we present and evaluate an integrated synchronous/asynchronous access control algorithm for a ring network with spatial reuse [8, 9, 10, 11]. A ring network with spatial reuse allows multiple stations to initiate transmissions across the ring at the same time. As a result, concurrent transmissions over distinct segments of the ring can take place, leading to higher network throughput levels. A spatial reuse mechanism can be implemented in a ring network by employing a destination removal technique. Consider a full-duplex ring network which employs a spatial reuse mechanism. Such a network operation can be governed by an access-control scheme based upon a slotted ring network structure [9, 10, 11] or upon a buffer-insertion ring architecture [8]. Under a uniform destination address distribution pattern, and using a shortest path routing algorithm (utilizing both rings, so that the ring with the shortest path to the destination is used), the dual-ring architecture using such a spatial-reuse mechanism can provide a high network throughput level, approaching the ideal limit of an average channel capacity utilization of 400 percent of that attained by a single ring. However, spatial reuse access mechanisms can lead to fairness problems in distributing the ring's bandwidth among distinct nodes. A heavy traffic node can hog the transmission channel for an excessively long time, preventing its downstream neighboring nodes from accessing the ring. "Unfairness", as it relates to access opportunities of distinct nodes, thus occurs. Hence, an inte-

*This work was supported by National Science Council, Republic of China, under Grant NSC85-2213-E-182-011

grated access algorithm for such a spatial-reuse ring is required not only to support the service requirements for both types of traffic but also to provide fairness in accessing the network and in allocating bandwidths.

Access algorithm designs based upon a framed approach have been used in many network systems for a support of the integrated services [2, 3, 5, 6, 9, 12, 13, 14, 15]. A time frame, which consists of a fixed number of contiguous time slots, is further divided into subframes which are dedicated to different types of traffic. The boundaries between subframes can be fixed or dynamically determined. Under a fixed-boundary scheme, a fixed number of slots per frame is allocated to each type of traffic, which then can only utilize its dedicated bandwidth. Thus, transmission gaps occur, resulting in the inefficient use of bandwidth resources. On the other hand, a movable-boundary scheme, where the boundaries between subframes are dynamically allocated, can allow the channel capacity level which is not used by one type of traffic to be used by the other type of traffic. Such a dynamic bandwidth allocation scheme clearly offers better bandwidth utilization levels than those achieved by employing a fixed-boundary procedure. However, under the movable-boundary scheme, a portion of the channel bandwidth, denoted as the switching overhead, is generally lost when the system is dynamically switching from one subframe to the next one. Such a channel capacity loss increases as the propagation delay normalized to a packet's transmission time increases. Thus, an integrated algorithm design based upon a movable-boundary scheme should be addressed to reduce the switching overhead, specially for a long-range, high speed network.

In this paper, we propose a new access protocol to integrate synchronous/asynchronous traffic types of messages across a full-duplex slotted ring network which employs a destination removal procedure. This integrated scheme, denoted as the integrated FECCA algorithm, is extended from the FECCA algorithm introduced by us in [1]. The FECCA algorithm is mainly employed to provide access fairness for asynchronous packets. It is a quota-oriented cycle-based fairness scheme, and has been applied to a full-duplex slotted ATM ring network with spatial reuse. Under the FECCA algorithm, We employ a reverse access control scheme. Thus, some distinct network nodes are able to transmit extra data beyond their prescribed quota without preventing the access opportunities of currently active downstream neighbors. As a result, a network employing the FECCA algorithm can achieve network throughput levels higher than those attained by using a standard cycle-based

algorithm.

Under the proposed integrated scheme, a framed approach is employed. A single time frame, which contains a fixed number of contiguous time slots, consists of a "synchronous subframe" followed by an "asynchronous subframe". Each network node is guaranteed to be able to transmit its predefined quota of synchronous data during each frame. A movable-boundary scheme is used to initiate and terminate subframes. The termination of a synchronous subframe is dynamically determined by the time at which each of network nodes finishes the transmission of its permissible quota of synchronous data. On the other hand, an "asynchronous cycle" is defined and a predefined quota for each node to transmit asynchronous data within each cycle is allocated. By using the reverse access control scheme as is used for the FECCA algorithm, transmission of asynchronous data within the synchronous subframe can take place among certain upstream nodes without interfering with those downstream nodes currently active in synchronous data transmissions. The switching overhead between subframes is thus substantially reduced. Extra asynchronous data transmissions beyond the specified quota are also realized for certain network node pairs. In this manner, this integrated scheme can provide guaranteed bandwidth and bounded delay for the support of a specified quota of synchronous connections while at the same time is able to efficiently (and in a fair manner) utilize the remaining bandwidth to support asynchronous data. In section 3, we present illustrative performance examples to demonstrate the effectiveness and efficiency of such an integrated algorithm.

This paper is organized as follows. In Section 2, we describe the integrated access protocol in detail. In Section 3, illustrative simulation examples are shown to demonstrate the performance effectiveness of this integrated mechanism. Finally, conclusions are drawn in Section 4.

2 Description of the Integrated FECCA Algorithm

In this section, we present the integrated algorithm. Since the integrated FECCA algorithm is based upon the FECCA protocol, we first describe the FECCA access scheme in the following.

2.1 The FECCA Access Algorithm

The FECCA (Fair and Efficient Cyclic Control Algorithm) algorithm is a cycle-based fairness control mechanism, which is applied to a full-duplex ring structure employing an ATM-based slotted ring access protocol. This access control scheme is modified

from the ATMR access protocol [11] to employ the reverse ring for the transmission of signaling information. This network allows concurrent transmissions of data through the use of a destination removal technique. Each one of the two counter rotating rings employs the FECCA algorithm, individually. In the following, we only describe this access algorithm as it is employed in the communication channel of the clockwise rotating ring, denoted as Ring *A*. The counterclockwise rotating ring, denoted as Ring *B*, is used for the transmission of signaling information generated by a node which wishes to transmit a message across Ring *A*.

Under this fairness scheme, each node is guaranteed to be allowed to transmit a number of ATM cells which is equal to its prescribed window size, denoted by W_{in} [cells], within any single cycle. The duration of a fairness cycle is determined as follows. A special busy address field is embedded in the header of each slot. After a new cycle starts, while an active node (*i.e.*, a node which has cells ready for transmission across Ring *A*.) waits for incoming empty slots (on Ring *A*) to transmit its cells, it overwrites its own address into the busy address field of each incoming (idle or busy) slot on Ring *B*. An active node (on Ring *A*) always insists to overwrite its own address into the busy address field of every incoming slot (on Ring *B*), even when the underlying busy address field may have been already written to by another active network node. A node becomes inactive (on Ring *A*) either when it finishes the transmission of its permitted window size, W_{in} [cells], within the current cycle or when it has no more cells queued at its buffer. (Note that in the latter case such a node will become active again once new ATM cells arrive at the node.) Once a node becomes inactive (on Ring *A*), it stops overwriting its own address into the busy address fields of subsequent incoming slots (on Ring *B*). Therefore, whenever a node detects its own address appearing in the busy address field of an incoming slot (on Ring *B*), it concludes that all of the other network nodes have become inactive (on Ring *A*). At this time, if this node is in the inactive state, it immediately issues a reset (cycle restart) signal. The reset signal circulates around the ring network to inform other network nodes that a new cycle is established. Through the use of this busy address field control scheme, a quota-oriented cycle-based fairness regulation scheme is implemented. Noting that the most significant advantage for employing such a fairness control scheme (by using the reverse ring channel to signal quota exhaustion) is that each upstream node (on Ring *A*) can detect the on-going activity status of

its downstream neighbors (on Ring *A*). That is, the busy address field of an incoming slot (on Ring *B*) observed by a node indicates the nearest downstream (currently) active neighbor of this node. Using this information, an inactive upstream node (which has completed its transmission within the current cycle) is permitted to transmit extra cells (beyond its quota) in subsequent empty slots, provided it has detected that its nearest active downstream neighbor is not located across the path of those extra cells. It is noted that the improved access opportunities gained by a properly situated node, do not interfere with the access abilities of this node's active downstream neighbors. While transmitting such extra cells, a node does not overwrite its address in the busy address fields of incoming slots. As a consequence, the cycle duration is not affected by the occurrence of such extra cell transmissions. As a result, the FECCA algorithm is able to reduce key bandwidth inefficiencies exhibited by a standard cycle-based global fairness scheme, thus attaining higher network throughput levels.

2.2 The Integrated FECCA Algorithm

As described for the FECCA algorithm, the integrated FECCA algorithm is also applied to a full-duplex ring structure which employs an ATM-based slotted ring access protocol. The length of an ATM cell is equal to a single slot payload duration. The reverse ring is used for the transmission of signalling information. Since this integrated mechanism is applied to each of the two unidirectional rings, individually, we describe in the following the application of the access algorithm to only one of the two rings.

Network Organization

A framed approach is employed. A time frame consists of a "synchronous subframe" followed by an "asynchronous subframe". The duration of a single frame is fixed. For example, it can be set to be 6ms, which is the duration that a 64kbps voice connection takes to generate a single ATM cell (48 bytes). The number of slots included in a frame thus depends upon the ring channel's data rate. A master station is selected to periodically generate frame start markers. A frame marker, which can be embedded in the header of a slot (which circulates around the ring) is used to designate the start of a new frame. The ring topology is divided into several contiguous segments. Network nodes which reside in a single segment are associated with the same Segment i.d.(SID).

Slot Format

The slot format is shown in Figure 1(a). A slot consists of header and information fields. The information field can accommodate a single ATM cell. The header, used as an access control field, is further di-

vided into several subfields, including the following information (relevant to our scheme):

B (1 bit): The busy (or full/empty) bit. If this bit is set to be 1 (or 0), it indicates that this slot is full (or empty).

M (1 bit): The monitor bit. The master station always sets this bit to 1, for a received slot whose M bit is equal to 0. This bit is set to 0 by a network node when it is performing the (synchronous or asynchronous) overwriting procedure (see below).

D (1 bit): The data type bit. This bit is set to 0 (or 1) by a network node which is performing the synchronous (or asynchronous) overwriting procedure (see below).

SID (a few bits): The segment i.d. field. The number of bits in SID is a system parameter which we discuss in section 4. When a network node is performing the (synchronous or asynchronous) overwriting procedure, it overwrites its own segment i.d. into this field.

Synchronous Data Access Protocol

Each node can prescribe a quota (Q_s) of synchronous data which it is allowed to transmit in each frame. Within the duration of a single frame, each node resides in either a "synchronous-mode active" state or a "synchronous-mode inactive" state. At the start of each frame (by detecting a frame marker), each node becomes synchronous-mode active and starts to transmit its synchronous cells up to its prescribed quota. At the same time, it performs the "synchronous overwriting procedure"; that is, it overwrites the header's SID field of each incoming slot (on the reverse ring) with its own segment I.D. and sets D (the data type bit) = 0, M (the monitor bit) = 0. After it has transmitted Q_s synchronous cells (or there is no more queued synchronous cells left), it becomes synchronous-mode inactive and stops its synchronous overwriting process. It is then ready to transmit its asynchronous data (if any) during the remainder of the current frame. That is, at any single frame, a node can start to transmit its asynchronous cells only after it has become synchronous-mode inactive.

Asynchronous Data Access Protocol

Each node can prescribe a predefined quota (Q_a) of asynchronous cells in each "asynchronous cycle". Within any asynchronous cycle, each node stays in either an "asynchronous-mode active" state or an "asynchronous-mode inactive" state. The duration

of a single asynchronous cycle is dynamically determined and each cycle generally contains a large number of frames. A node is defined to be asynchronous-mode active if it has queued-ready asynchronous cells to send and it hasn't exhausted the transmission of its Q_a asynchronous cells within the underlying asynchronous cycle. While a node is both synchronous-mode inactive and asynchronous-mode active, three following conditions are noted: (1) It has observed an incoming slot (on the reverse ring) with D=0 indicator and the SID field of such a slot contains a I.D. different from the node's segment I.D. This indicates one or more of its downstream neighbors located at other segments are still synchronous-mode active; therefore, currently the ring network is still operated in the "synchronous subframe" of the underlying frame, although this node itself has become synchronous-mode inactive. No overwriting procedure will be performed by this node under such a condition. However, it can insert its asynchronous cell into an incoming empty slot provided that it has detected that the nearest synchronous-mode active downstream segment is not located across the path of its queued asynchronous cell. (2) It has observed that its own segment I.D. appearing in the header's SID field of an incoming slots with D=0 and M=1. It then concludes that all of the network nodes residing in other segments of the ring are synchronous-mode inactive. Thus, it performs the "asynchronous overwriting procedure"; that is, it overwrites the header's SID field of the incoming slot (on the reverse ring) with its own segment I.D. and sets D = 1, M = 0. It will insert its asynchronous cell into an incoming empty slot. It is noted that only a network node which is situated as the most downstream node of some segment will have such an observation. (3) It has observed a D=1 indicator. Under such a condition, each of its downstream neighbors is synchronous-mode inactive. It thus performs the asynchronous overwriting procedure and inserts its asynchronous cell into an incoming empty slot.

If a node has become asynchronous-mode inactive, it may still transmit extra asynchronous cells (beyond its assigned asynchronous quota) provided that it finds that these transmissions will not prevent the network nodes of its nearest active downstream segment from accessing the channel. A more detailed description regarding asynchronous data access protocol is depicted in Figure 1(b).

Asynchronous Cycle Reset Scheme

When the master station detects an incoming slot with D=1 and M=1, it concludes that every node in the network has already exhausted its asynchronous quota or completed the transmission of asynchronous

cells. The master station then issues an asynchronous traffic reset signal. This reset signal circulates around the ring network to inform network nodes that a new asynchronous cycle is established.

3 Numerical Results

In this section, we present simulation results to demonstrate the effectiveness of the integrated FECCA algorithm. We assume the destination addresses of ATM cells to follow a uniform distribution over a range for which the maximal travel distance is equal to half of the ring's length. Since the queueing behavior exhibited by each ring is statistically identical, we only simulate a single unidirectional ring. The ring network is assumed to connect 128 network nodes. Each pair of neighboring nodes is assumed to be at the same distance as any other pair. For synchronous traffic, we assume that each node supports the same number, a system parameter, of synchronous sessions for each simulation case. A single synchronous session alternates between "ON" and "OFF" states. The holding time [in frames] at each state is geometrically distributed. While in the ON state, a single ATM cell is generated at the start of each frame. No cells are generated during the OFF state. Each node is assigned the same Q_s value, representing the node's quota, for transmitting its synchronous cells within any single frame. For asynchronous traffic, a Poisson arrival process at each node is assumed. Asynchronous cells can be generated at each slot. Each node is assigned the same Q_a value for transmitting its asynchronous cells within any single asynchronous cycle. In order to evaluate the bandwidth gain that can be realized due to extra transmission opportunities offered by the integrated FECCA algorithm, we exhibit performance results attained under the use of the integrated FECCA algorithms as well as results obtained when extra transmission opportunities are not realized.

In Figure 2, we have used simulations to illustrate throughput performance results exhibited by a ring network with 64 rotating slots across the network. Each node is assumed to be fully loaded with asynchronous traffic. We assume that the ring network is uniformly divided into 128 segments. Each segment thus covers a single network node. A single (synchronous) frame is assumed to consist of 400 slots. Each node is assigned the same Q_s value of 6 [cells/frame], specifying the maximum number of synchronous cells it is permitted to transmit within any single frame. Note that to be able to guarantee the realization of a quota Q_s for a ring which accommodates 128 nodes, at a frame size of 400 slots, we must set $Q_s \leq 6 = \lfloor \frac{400}{64} \rfloor$. Each node is assigned the same asyn-

chronous quota $Q_a = 80$ [cells/asynchronous cycle]. Network throughput levels as well as the throughput performance of asynchronous traffic are depicted vs various loading of synchronous traffic. For any particular loading of synchronous traffic, the use of the integrated FECCA algorithm leads to a higher realized network throughput level (as well as asynchronous throughput level) by a wide margin. In comparing these throughput levels with those obtained when no extra transmissions are realized, a throughput margin larger than 0.6 is demonstrated. Note that a global throughput level which is close to the theoretical upper bound value of 4.0 is achieved. These results demonstrate the advantage of the integrated FECCA algorithm in effectively utilizing extra cell transmissions.

In Figure 3, we show delay and throughput performance of asynchronous traffic for a network scenario similar to that used for Figure 2. The loading of synchronous traffic is set to be 0.76. We study the delay performance behavior as the asynchronous traffic load is gradually increased (while in Fig. 2 we assumed asynchronous traffic overloading conditions). The better delay-throughput performance achieved by the integrated FECCA scheme is clearly exhibited.

In Figure 4, we consider a network scenario where there are 16 rotating slots across the ring. The integrated FECCA algorithm is employed. Each node is fully loaded with asynchronous traffic and the total loading of synchronous traffic is set to be 1.54. Network throughput performance and asynchronous throughput performance results are depicted vs a varying number of configured segments across the ring. (Note that the total number of nodes stays constant at 128.) The ring architecture is uniformly divided into segments for each particular selected number of segments. In such a manner, each segment covers the same number of network nodes. The achieved throughput levels increase (very slowly) as the number of segments increases. These results account for the fact that as the number of segments increases (*i.e.* each segment contains a smaller number of network nodes), network nodes are able to identify a clear path for extra cell transmissions at a higher level of precision. Thus, extra transmission opportunities can be better utilized. However, the differences in the throughput levels attained through the use of different numbers of segments are shown to be very small. Therefore, a small number of segments may be selected, resulting in reduction of the header overhead introduced by the bits required for representing the segment's i.d. field.

4 Conclusions

In this paper, we have presented an integrated FECCA algorithm to jointly support the transmission of synchronous and asynchronous traffic across a full-duplex ATM ring network with spatial reuse. A framed approach is used in this integrated algorithm. A frame consists of a synchronous subframe followed by an asynchronous subframe. A movable-boundary scheme is employed to dynamically establish the boundaries between subframes.

We use the reverse ring for the transmission of signaling information. Network nodes can thus identify clear paths for extra cell transmissions. In this manner, bandwidth utilization levels are improved. Bandwidth utilization inefficiencies caused by switching between subframes are greatly reduced. As a result, this integrated scheme provides guaranteed bandwidth and bounded delay for the support of a specified quota of synchronous connections while at the same time is able to efficiently (and in a fair manner) utilize the remaining bandwidth to support asynchronous traffic. We have presented a number of system examples to demonstrate the effectiveness and efficiency of this integrated algorithm.

References

- [1] I. Rubin and H.-T. Wu, "FECCA - A New Access Algorithm for an ATM Ring Network with Destination Removal" *Proc. of IEEE INFOCOM'93*, Mar. 1993, pp. 368-375.
- [2] J.-H. Huang, C.-W. Chen and M.-C. Lee, "A Distributed Fair, and Efficient Protocol for Integrated Voice/Data Services on Token Ring Networks," *Proc. of IEEE ICC'91*, pp. 1362-1366.
- [3] W.-T. Chen, R.-R. Lee, J.-H. Yu and C.-H. Huang "A Fair Integrated Voice/Data Protocol for Token Ring Networks," *Proc. of IEEE GLOBECOM'90*, pp. 1416-1420.
- [4] H.-T. Wu, Y. Ofek and K. Sohraby, "Integration of Synchronous and Asynchronous Traffic on the MetaRing Architecture and its Performance Study," *Proc. of IEEE ICC'92*, pp. 147-153.
- [5] B. Mukherjee and J. Meditch, "Integrating voice with the p_i -Persistent Protocol for Unidirectional Broadcast Bus Networks," *IEEE Tran. on COMM.*, Vol. 36, No. 12, Dec. 1988 pp. 1287-1295.
- [6] M. Fine and F. Tobagi, "Packet voice on a Local Area Network with Round Robin Service," *IEEE Trans. on COMM.*, Vol. 34, Sept. 1986 pp. 906-915.
- [7] H. C. Du and D. S. Ghanta, "A New Access Protocol for Uni-directional Twin-bus Architectures," *Proc. of IEEE INFOCOM'88*, pp. 805-812.
- [8] I. Cidon and Y. Ofek, "MetaRing - A Full-Duplex Ring with Fairness and Spatial Reuse," *Proc. of IEEE INFOCOM'90*, pp. 969-981.
- [9] A. A. Lazar, A. T. Temple and R. Gidron, "MAGNET II: A Metropolitan Area Network Based on Asynchronous Time Sharing," *IEEE J. on Selected Areas in Comm.*, Vol. SAC-8, Oct. 1990, pp. 1582-1594.
- [10] R. M. Falconer and J. L. Adams, "Orwell: a protocol for an integrated services local network," *British Telecom Tech. J.*, Vol. 3, No. 4, Oct. 1985, pp. 27-35.
- [11] H. Ohnishi, N. Morita and S. Suzuki, "ATM Ring Protocol and Performance," *Proc. of IEEE ICC'89*, 13.1, pp. 394-398.
- [12] S.-Q. Li and M. Zarki, "Dynamic Bandwidth Allocation on a Slotted Ring with Integrated Services," *IEEE Trans. on COMM.*, Vol. 36, July 1988, pp. 826-833.
- [13] J. Meditch and Y. Zhao, "Framed TDMA/CSMA for Integrated Voice-Data Local Area Network," *Proc. of IEEE INFOCOM'85*, pp. 10-17
- [14] H. Lee and C. Un, "Performance Analysis of Statistical Voice/data Multiplexing Systems with Voice Storage," *IEEE Trans. on COMM.*, Vol. 33, Aug. 1985, pp. 809-819.
- [15] A. Konheim and R. Pickholtz, "Analysis of Integrated Voice/data Multiplexing," *IEEE Trans. on COMM.*, Vol. 32, Feb. 1984, pp. 140-147.

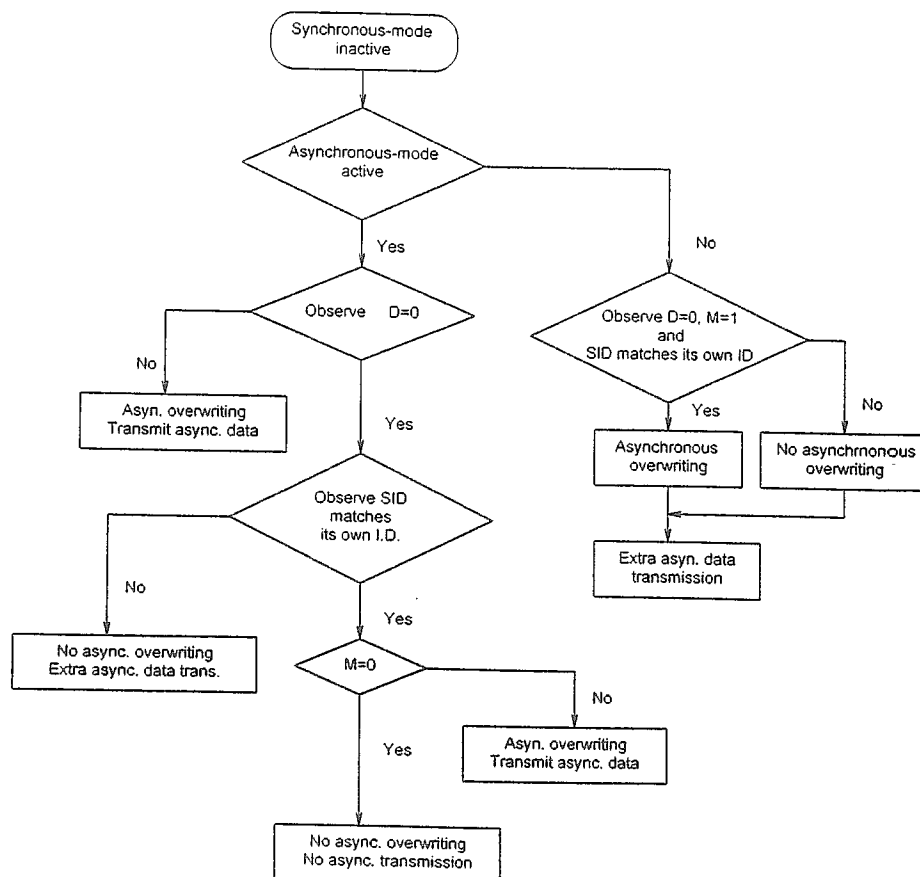
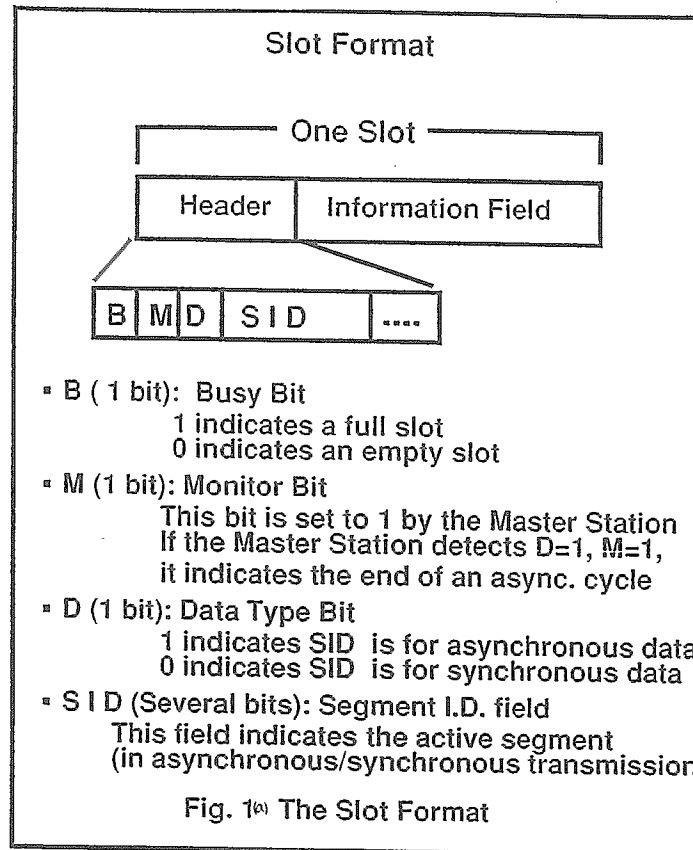


Fig. 1(b) Diagram of Asynchronous Data Access Protocol

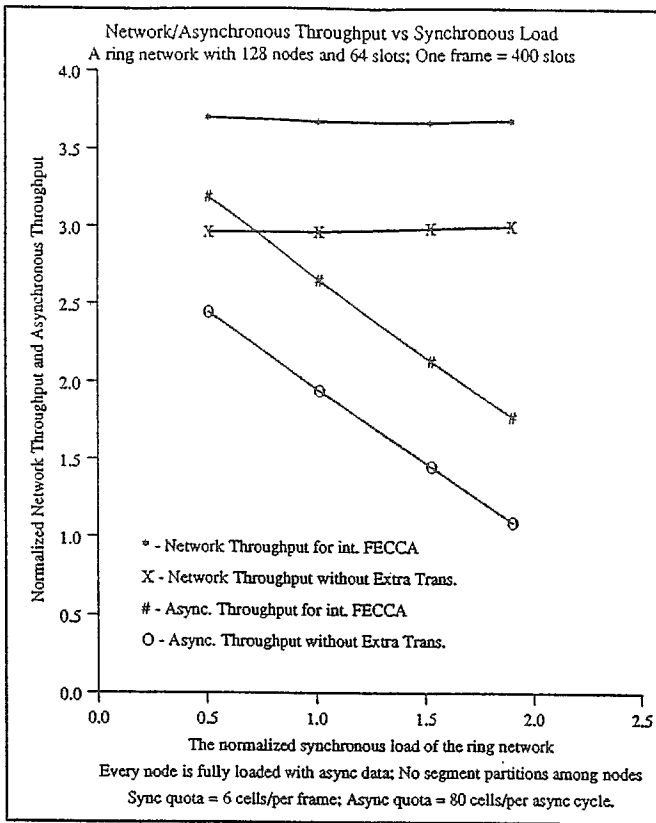


Fig. 2 Throughput Performances for the Integrated FECCA Algorithm

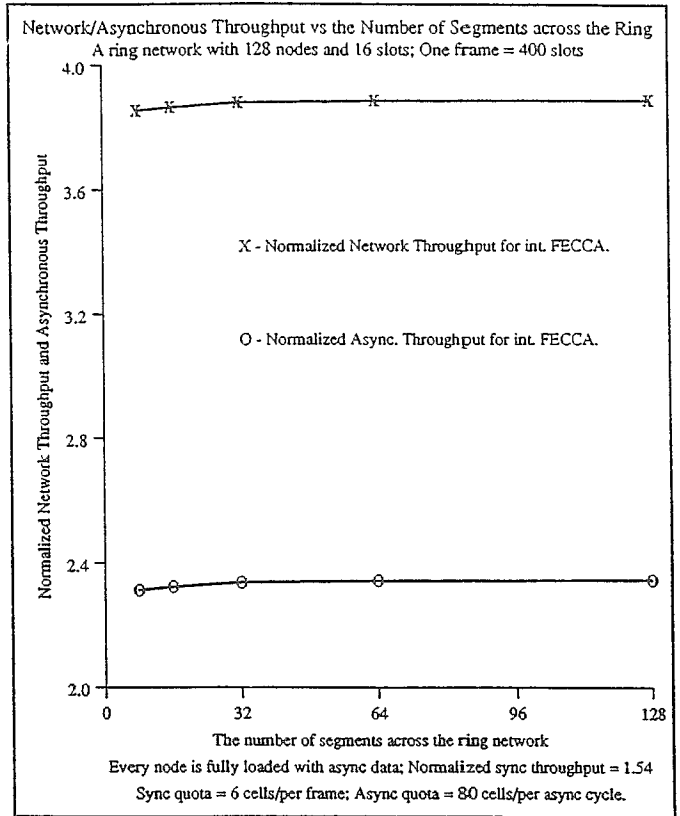


Fig. 4 Throughput Performances for the Integrated FECCA Algorithm

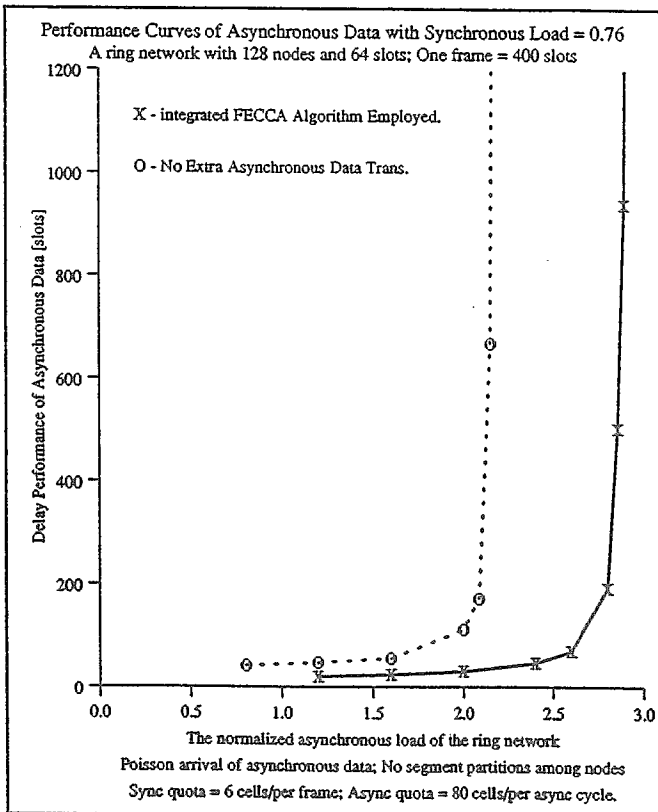


Fig. 3 Delay and Throughput Performances for the Integrated FECCA Algorithm