

## CRMA-II: A Gbit/s MAC Protocol for Ring and Bus Networks with Immediate Access Capability

Harmen R. van As, Wolfram W. Lemppenau, Pitro Zafiropulo, Erwin A. Zurfluh

IBM Research Division, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

### Abstract

CRMA-II is a reservation-based MAC protocol for LANs and MANs designed to operate in the Gbit/s and multi-Gbit/s speed range. It applies equally well to ring and bus networks with or without buffer insertion. By combining a simple busy/free slot-access protocol with a reservation mechanism, fast access is achieved when a node experiences low to medium load conditions whereas fairness and tight access delay bounds are maintained during periods of sustained heavy load or overload. Delay and throughput are further enhanced by spatial reuse. The protocol uses a novel MAC-command transmission technique that is well suited for a high clock-rate implementation and that decouples reservation processing from the high-speed medium. In this paper, we focus on the transmission data structure, the MAC-command transmission technique, and the MAC protocol.

### 1. Introduction

CRMA-II (Cyclic-Reservation Multiple-Access) is a high-performance Medium Access Control (MAC) protocol for Local Area Networks (LANs) and Metropolitan Area Networks (MANs) designed to operate up to multi-Gbit/s speeds. This new protocol is based on the experience gained from a Gbit/s implementation of the earlier CRMA version for bus topologies [1-4]. As a result, a combination of good performance features, suitability for multi-Gbit/s realization, and protocol robustness in the presence of errors has been achieved. The MAC protocol combines the best properties of a simple busy/free protocol for accessing slots with those of a reservation-based protocol. The simple busy/free access component gives fast access when a node experiences low to medium load conditions and allows full exploitation of spatial reuse. The reservation mechanism permits a sustained high network utilization while maintaining fairness and tight access delay bounds during periods of heavy load. The protocol uses a new MAC-command transmission technique that permits a high clock-rate implementation and that decouples reservation processing from the high-speed medium. These qualities are well suited for Gbit/s LANs and MANs where backbone networking as well as high-resolution image-based applications require both high throughput and fast response [5-10].

High performance at Gbit/s speeds requires the MAC protocol to be based on concurrent access. The slotted approach, proposed or used in several LANs [11-18] and MANs [19-26], allows simultaneous access by multiple

geographically separated nodes. Because of this, throughput does not degrade with increasing network speed and geographic coverage. Performance can further be enhanced by spatial reuse which is the ability of destinations to free transmission capacity for reuse by the node itself or downstream nodes. In fact, spatial reuse [13-16], [18, 27-29] potentially increases system throughput well beyond the medium bit rate. In today's LANs, transmission is via entire frames (packets) which reduces additional transmission overhead caused by segment labelling and which simplifies receiver buffering by limiting reception to one single frame at a time. On the other hand, ATM-based cell transmission, according to CCITT's Asynchronous Transfer Mode [30], is being promoted as the solution for Broadband-ISDN (B-ISDN) and is accommodated in the IEEE 802.6 MAN standard DQDB (Distributed Queue Dual Bus) [19]. CRMA-II supports both frame-based transmissions and cell-based transmissions.

Contiguous frame transmission in a slotted LAN is achieved by reservation allowing an entire frame to be loaded into a contiguous set of reserved slots [1-4]. Buffer insertion techniques [18, 27, 31-34] provide concurrent access, spatial reuse and contiguous frame transmission in an unified way. Its operation can be slotted as in [18] and in CRMA-II or unslotted as in [27, 31-34]. The buffer insertion approach is of interest because dual-port RAM technology has progressed to very high clock rates. The fairness problem, originally present in early buffer insertion techniques, is now controllable by credit-based access schemes [18, 27] or by a reservation mechanism such as the one described in this paper. Recently, buffer insertion approaches have also been enhanced with buffer bypass mechanisms [18, 33] to support isochronous traffic as well as synchronous traffic intended for real-time and delay-sensitive connections. Though insertion buffer bypass is supported in CRMA-II, it is not the subject of this paper and will not be considered in further detail.

After a general overview, the paper continues with a description of the transmission data structure and the MAC-command transmission technique used for reservation. We then introduce the CRMA-II access mechanism by describing first the principles for cell-based transmission. We begin with the simple busy/free access (the so-called gratis access) followed by the reservation-based access and conclude with a combination of the two. Thereafter, we focus on the insertion buffer operation and reservation scheduling. Operation principles for a ring and bus network with or without buffer insertion follow next. Finally, priorities and the robustness issue are addressed.

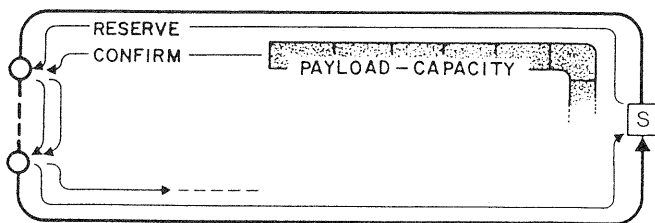


Fig. 1: Reservation principle

## 2. System overview

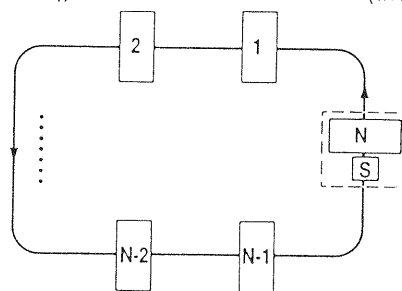
CRMA-II is based on a slotted transmission data structure which is used for a network operation without or with buffer insertion. The transmission structure is therefore based on a slotted or slotted buffer-insertion operation. Frame transmissions take place in contiguous slots whereby contiguity is achieved by buffer insertion as proposed in [18]. ATM-based cells are transmitted in single slots similar to the operation in well-known slotted LANs and MANs [11-26].

For optimum performance, slots are accessed through two distinct mechanisms: (1) immediate and unrestricted access of so-called gratis slots and (2) access of previously reserved slots. The gratis/reserved flag is used to distinguish between these two states of a slot. In either case, the busy/free flag needs to be free for the slot to be accessible. Whereas access to gratis slots is unrestricted, access to reserved slots requires a prior Reserve/Confirm command exchange. These MAC commands are not part of the slots and are inserted as autonomous entities on slot boundaries of the transmission data structure. This is possible because all transmission entities (e.g. slots, commands) are embedded between start/end delimiter pairs. The underlying reservation principle is shown in **Figure 1**. A node with scheduler function S issues a Reserve command that collects transmission requests as it proceeds around the medium. A slot is the unit of reservation so that requests for transmissions are in integer numbers of slots. Return of the Reserve command enables the scheduler to know the transmission requirements of all active nodes. The scheduler is then in a position to allocate payload fairly by issuing a Confirm command informing each node of the number of reserved slots it can use. Subsequently, the scheduler marks a corresponding number of slots as reserved, thereby creating payload containers for the reserved slot transmissions. A reservation is for only one transmission access. When a slot is freed by the destination, subsequent access to that slot is unrestricted (gratis). This allows effective exploitation of spatial reuse. It is this combination of gratis and reserved access that provides the good performance properties of CRMA-II. Gratis access permits fast access when a node experiences a low to medium load condition although other parts of the network might be heavily utilized. Gratis access also enhances network throughput and minimizes overall delay. During heavy load periods, gratis access mainly occurs because of spatial reuse. To maintain fairness, the scheduler allocates more reserved slots to those nodes that had less gratis-access opportunities during the previous reservation cycle. Thus a temporarily difference in

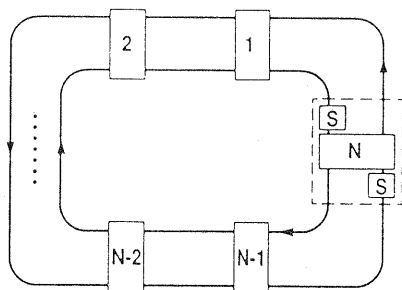
node throughput caused by accessing spatial-reusable gratis slots, is continuously corrected by the reservation mechanism. Apart from maintaining fairness, the reserved access ensures tight access delay bounds for low-traffic users and supports priorities in an effective way.

### 2.1. Network topologies and operating modes

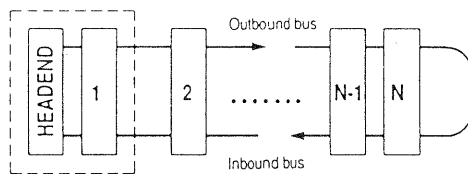
With a single MAC-protocol, CRMA-II covers both ring and bus topologies and operates with or without buffer insertion. For nodes transmitting via ATM-cells, a slotted transmission structure is sufficient. However all nodes supporting contiguous frame transmissions (thus requiring



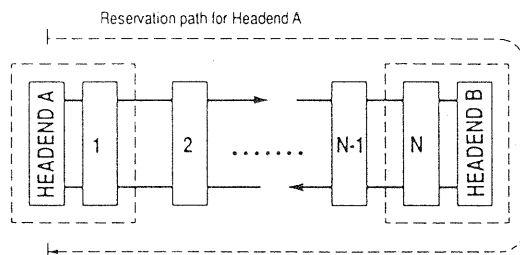
single ring



dual ring



folded bus



dual bus

Fig. 2: Network topologies

contiguous slots) need an insertion buffer. On the MAC-layer, CRMA-II enables both kind of nodes to coexist on the same network. The protocol is primarily designed to operate on a single ring. Its extensions to a dual ring or multiple rings as well as to a folded or dual bus are straightforward. **Figure 2** shows the four basic topologies supported by CRMA-II: single ring, dual ring, folded bus and dual bus. The scheduler S, which can be a microcoded function, is replicated in each node but is active in only one node at a given time. The dual ring operation implies a separate scheduler function on each ring. Both schedulers can be located either in the same or in different nodes. In bus topologies, the scheduler is an integrated part of the headend. Bus operation requires two further peculiarities to be noticed. In the folded bus topology where transmissions occur on the outbound bus and receptions on the inbound bus, spatial reuse is not possible. For a dual bus operation, the MAC commands for the reservation mechanism are relayed back on the reverse bus by the other headend. Nodes distinguish between the actual commands and relayed ones by a flag.

Aside from many commonalities between ring and bus operation, a major distinction is the origin of free slots. In bus topologies, slots are generated at the headend and slots are discarded at the other end of the bus. Slots issued by the headend are always free. On a ring, slots circulate and consequently slots passing the scheduler are either free or busy. Free slots originate at nodes that remove data from the ring. Freeing slots (removal) occurs either at destination nodes or, in the case of broadcasting, at the source nodes.

On a ring the scheduler is part of a monitor, a function similar to the Token Ring monitor [35, 36]. An advantage of incorporating a monitor is that busy slots not properly freed by destination nodes are identified by the well-known Monitor-Count mechanism [35, 36] where a bit is set by the monitor in every passing busy slot. When the Monitor-Count bit is not reset by a receiving node, the slot is freed by the monitor. Since the monitor (headend) including the scheduler function is replicated in each node, automatic network reconfiguration is permitted [36-39].

The unified MAC protocol for ring and bus networks enables one common node design so that nodes can be attached to both network topologies. Essentially only a few inherent network differences must be taken into account. We mentioned above the relayed MAC commands on a dual bus and the difference in the origin of free slots. Note however that nodes on a ring also have a slot generator for network initialization and buffer insertion handling. Other differences relate to network reconfiguration and error recovery, both of which are currently under study. Owing to independent protocol handling at the receive and transmit side, operation without buffer insertion is achieved by omitting the insertion buffer. This does not impact other hardware modules except for the buffer controlling part.

This paper focuses on single ring operation for two reasons. Firstly, a dual ring topology is essentially a duplication of two single ring operations. Secondly, the bus topology where slots generated at the headend are always free can be regarded as a special case of a the ring. The extension of CRMA to rings is also being consid-

ered by others [40] where ATM cell transmissions and a speed target around 100 Mbit/s are envisaged. In contrast to our scheme, spatial reuse in [40] is limited to nodes that received confirmations for reserved slots so that spatial reuse cannot be exploited with full effectiveness.

## 2.2. Addressing

To permit rapid removal of data from slots, each slot carries two short addresses which are locally administered. They have also been proposed in [15, 18]. Removal actually means the marking of a previously busy slot as free. The first short address identifies the destination node, the second specifies the removal node freeing the slot. For data destined to a single node, the destination and removal addresses are the same. For data addressed to a group of nodes (broadcasting, multicast), the destination address is a group address and the removal address is that of the source node. Using short addresses is crucial at Gbit/s speeds because the decision to free the slots must be made at the speed of the medium. As discussed in [18], the two short addresses are introduced to minimize node latency. Their address domain is restricted to nodes on a single CRMA-II network. Further addressing is based on the address schemes of the transported data like IEEE 802.5 frames or ATM-cells.

## 2.3. Atomic Data Units

At Gbit/s speeds, MAC internal clocking time periods come in the range of single-gate propagation delays. Therefore we are approaching a point where one should never save bit positions at the expense of being forced to reduce speed. Prototyping has shown that the best way to cope with Gbit/s speed is to organize the transmission data structure into multi-byte words. To emphasize that these entities are not merely a grouping of bits into data words but rather indivisible processing units at the MAC-layer, they are called Atomic Data Units (ADUs). Apart from never being divided into individual smaller portions, ADUs are different from data words in several respects. Firstly, operations without feedback loops such as insert, copy and remove are always related to an entire ADU, never to certain portions of an ADU. Secondly, bits used for synchronization, control or short addressing are organized such that they allow regular pipelined processing without multiplexer stages. And finally, MAC-related ADUs are strictly separated from payload ADUs. Slots therefore consist of payload-carrying ADUs embedded between MAC-related ADUs: an start ADU along with an addressing ADU at one side and an end ADU at the other side. The start/end ADU pair concept is robust because simple matching achieves erroneous slot as well as erroneous command rejection. On the medium, we propose using an 8B/10B transmission coding scheme which transforms 8-bit data words into 10-bit transmission codewords as a function of an additional control-signal (control bit) and vice versa. This enables the coding of data words as well as an additional set of control words. 16-bit ADUs for bit rates up to 1.2 Gbit/s assuming today's BICMOS technology result in an ADU clocking frequency of up to 60 MHz. For higher speeds, 32-bit ADUs become necessary to maintain the same clocking frequency range and are the assumed ADU-size in this paper.

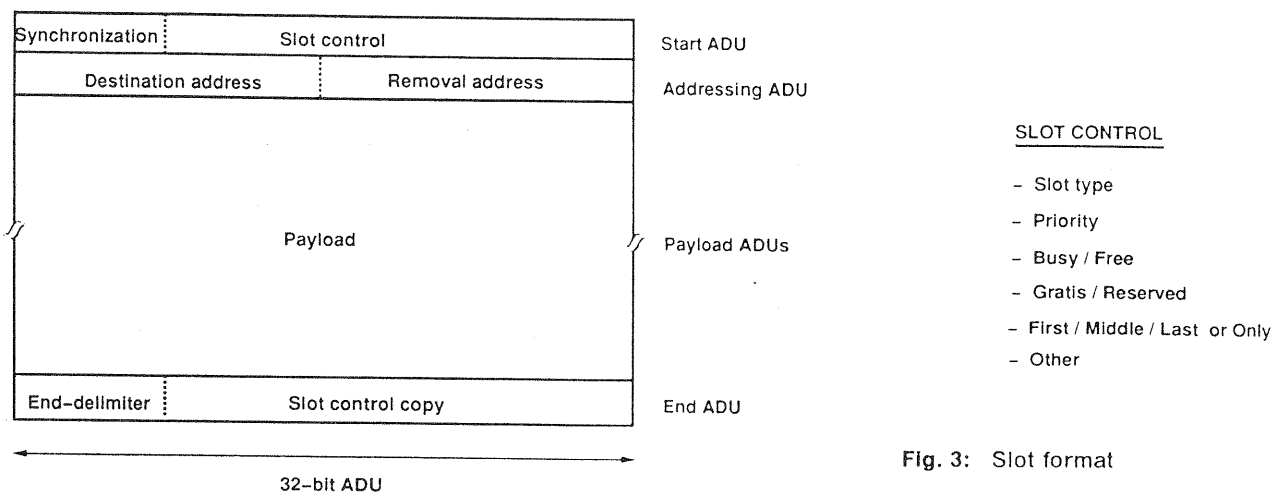


Fig. 3: Slot format

#### 2.4. Transmission data structure

The basic transmission entities are slots of constant size. **Figure 3** shows a slot format on the base of a 32-bit ADU size consisting of a start ADU with an 8-bit synchronization part (8B/10B control code) and a 24-bit slot control part carrying the slot type, slot priority and other parameters, where every bit position is not necessarily used. The slot type identifies the dynamically changing slot class as asynchronous, synchronous or isochronous whereby in this paper only asynchronous slots are considered. The slot priority defines a further differentiation in transmission service. The other parameters relevant to this paper are the slot status flags busy/free and gratis/reserved as well as the first/middle/last or only indication discussed below. The next ADU contains the two short addresses as mentioned above permitting rapid address identification for data copying and data removal. The number of ADUs as payload is selectable at network initialization time. To carry an ATM cell of 53 bytes, fourteen ADUs are required. The slot terminates with an end ADU which primarily contains the same slot control information as the start ADU. Owing to the slot control redundancy, a robust error detection is achieved by simple matching of unique start/end ADU pairs.

MAC commands are not part of the slots but are autonomous transmission entities that are inserted on demand between consecutive slot boundaries and are removed at their destination. **Figure 4** illustrates a slot sequence with a multi-slot consisting of a variable number of concatenated slots, and a MAC command consisting of a variable number of ADUs. A multi-slot is intended for frame transmission in contiguous slots.

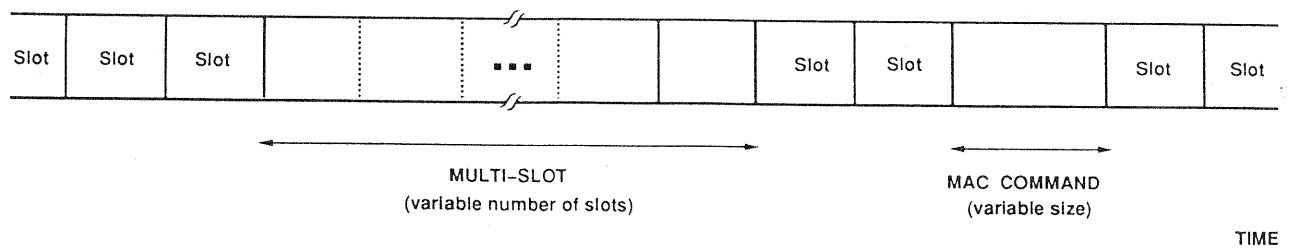


Fig. 4: Example of slot, multi-slot and MAC command sequence

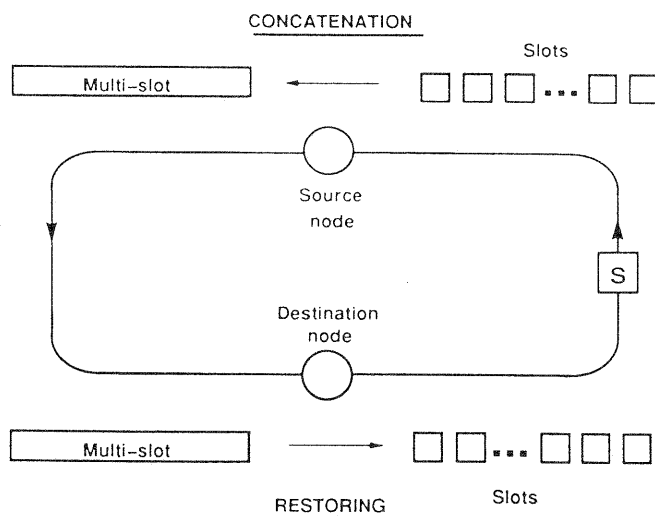


Fig. 5: Dynamic creation of a multi-slot

As shown in **Figure 5**, a multi-slot is dynamically created at a source node via slot concatenation and is restored to individual slots at the removal node which is here the destination. Concatenation enforces intermediate addressing and end ADUs to be overwritten by the frame payload, thereby reducing the transmission overhead. In other words, since the slot size remains the same, an increased payload per slot results. Start ADUs are maintained in a multi-slot transmission entity primarily to allow individual slot marking by the scheduler.

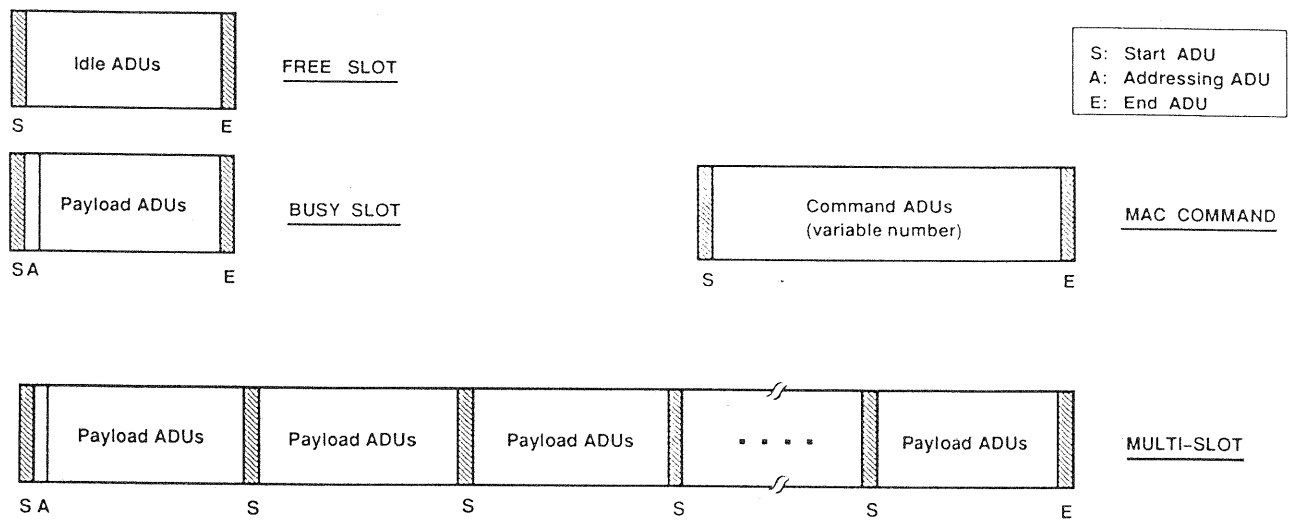


Fig. 6: Embedded transmission entities

Figure 6 illustrates the discussed transmission entities. Free slots carry idle ADUs between start ADU S and end ADU E. A busy slot carries the addressing ADU A and a number of payload ADUs. MAC commands consist of a variable number of command (parameter) ADUs which are inserted into or removed from the command on a demand basis. This is the subject of the next section. Finally, the figure shows the structure of a multi-slot with an addressing ADU A only in the first slot, start ADUs S in each slot, and an end ADU E only in the last slot. Concatenated slots are marked in their start ADUs as first, middle or last. Individual slots are marked as only.

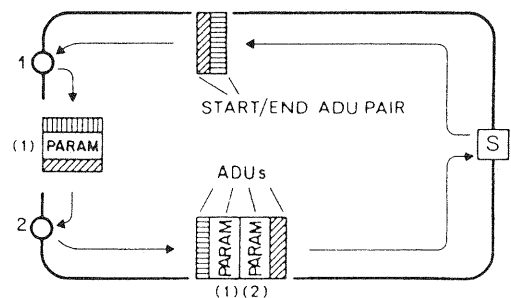
### 3. Insert/Remove MAC-command transmission technique

The ADU-based data structure enables the realization of a novel MAC-command transmission technique for performing information exchange between scheduler and nodes as well as between nodes. It reduces 'on the fly' operations at the MAC layer to copy, insert and remove. These three operations are tailored for Gbit/s speed implementations.

Figure 7 illustrates the principle of the insert/remove MAC-command transmission technique as used for a general request/response mode of operation. The scheduler node issues a command by transmitting a start/end ADU pair back-to-back. Each node that has requests inserts an ADU with appropriate parameters behind the start ADU thereby increasing the command length by one ADU. The so-called ADU delay register of the node is switched into the data path to hold up the arriving ADU during this insertion action. All data passing this node is from this point on delayed by one ADU.

On command return, the scheduler responds by issuing the next command containing one response ADU for each node that previously issued a request ADU. The response ADUs are positioned in the reverse order as the requesting ADUs so that nodes always receive their respective response parameter by removing the first ADU following the start ADU. This avoids individual node

addressing. As the nodes successively remove their ADU, the command gradually shrinks and ultimately returns to the scheduler consisting of the two delimiter ADUs. Each removal also causes the withdrawal of the 32-bit wide ADU delay register from the data path, thereby cancelling the delay insertion. Note that by this method, node-specific information is exchanged between nodes and a scheduler without explicit addresses and that on-demand bandwidth allocation with minimum control overhead is made possible.



Request/Response mode of operation

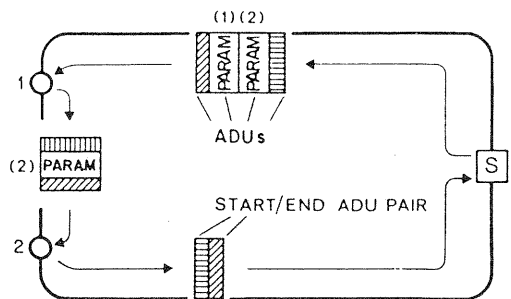


Fig. 7: Insert/Remove MAC-command transmission technique

#### 4. Media access protocol

We first describe the MAC protocol for the slotted operation and then focus on the slotted buffer-insertion mode of operation in the next section. For optimum performance, slots are accessed through two distinct mechanisms: (1) immediate and unrestricted access of so-called gratis slots and (2) access of previously reserved slots. The dynamic gratis/reserved attribute is used to distinguish between these two states of a slot. In either case, the busy/free flag needs to be free for the slot to be accessible. Based on the reservation requests received, the scheduler employs a scheduling algorithm to fairly allocate bandwidth among the competing nodes. As a result, it confirms the current allocation to the nodes using a technique described in Section 3 and marks the necessary number of slots as reserved. Slots are however only temporarily marked as reserved and become gratis when they are accessed. The reservation mechanism serves two purposes. First, it controls node throughputs according to a selected fairness discipline. Second, it sets tight access-delay bounds during heavy load periods. This is important for low-traffic users or priority handling.

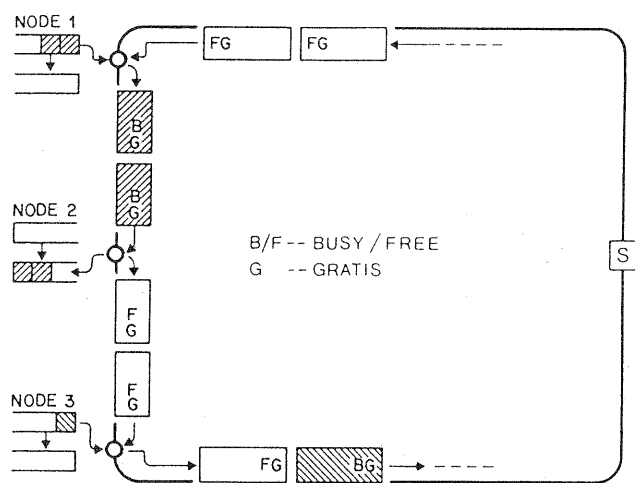


Fig. 8: Free access via gratis slots

Figure 8 shows the gratis access where node 1 has two pending slot transmissions. Two passing free-gratis (FG) slots are accessed which results in a busy-gratis (BG) slot status. Node 2 is the destination that copies the data and frees the slots by reverting their status to free-gratis. Transmission by node 3 via one of the previously used slots exemplifies spatial reuse. This mode of operation ensures fast access during light and medium load conditions and exploits spatial reuse effectively.

Figure 9 details the reservation access which occurs in three phases. During reservation, the scheduler issues an empty Reserve command. Upon arrival, node 1 inserts a reservation for  $i$  pending slots and node 2 for  $k$  slots. The scheduler then applies a fairness algorithm thereby creating confirmations  $i' \leq i$  and  $k' \leq k$  that are issued by the Confirm command via the previously described MAC command transmission technique. In the last phase, nodes 1 and 2 access  $i'$  and  $k'$  free-reserved (FR) slots,

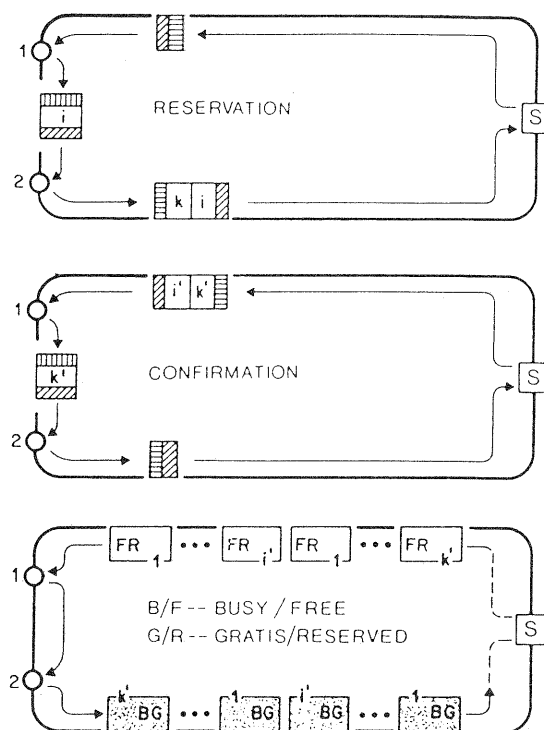


Fig. 9: Reserved slot access

respectively. Reserved slots follow immediately after the Confirm command. All accessed free-reserved slots become busy-gratis at the source nodes, whereby destination nodes free the slots for spatial reuse by making them free-gratis.

Figure 10 illustrates the transition from a predominantly gratis mode in a lightly loaded network to a reserve mode during heavy load periods. In the upper figure, a node has six pending slot transmissions which are all successfully completed before the Reserve command arrives. Consequently, the Reserve command remains empty and reservation access is not invoked. In the middle figure, the same node faces a medium loaded traffic environment where gratis and reservation access coexist. The node is only able to send three of its six pending slot transmissions in gratis slots. Thus it reserves three slots while simultaneously declaring the three gratis transmissions already completed. As shown in the figure, the inserted ADU consists therefore of two numbers: the upper one indicating the number of reservations and the lower one specifying the number of gratis slots used since the previous Reserve command. We shall see in Section 6 that declaration of gratis slots used is necessary to achieve fairness. The lower figure portrays a heavy load situation where the node is forced to make a reservation for all six pending transmissions. At the same time, it informs the scheduler that no gratis slots were used. It should, however, be noted that even in a heavily loaded network, a considerable proportion of the transmissions still occur via gratis access owing to spatial reuse.

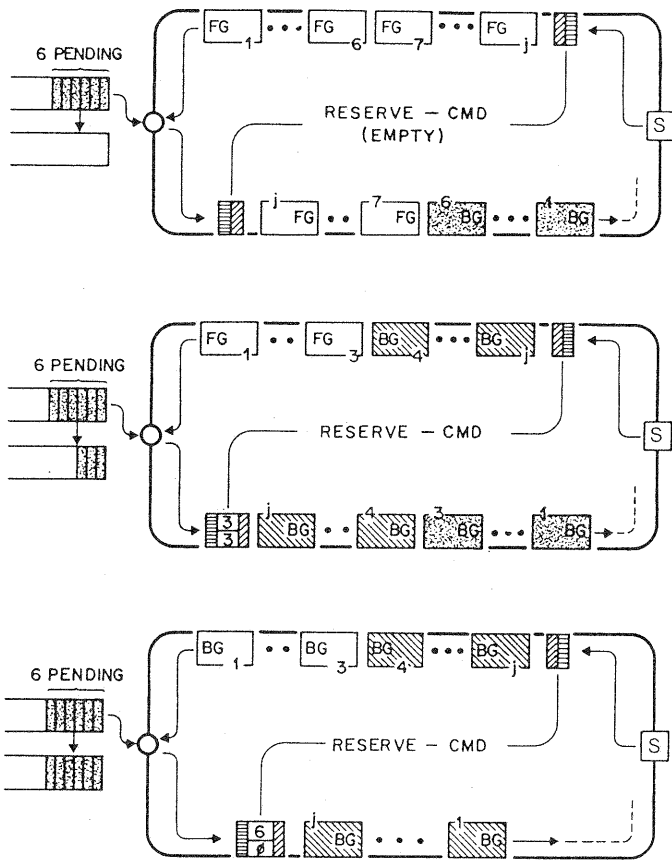


Fig. 10: Gratis access with reservation mechanism

We next track the busy/free and gratis/reserved dynamics of a single slot as it travels around the ring. Figure 11 shows a ring with a scheduler S plus seven nodes numbered 1 through 7. Starting at the asterisk, the slot of interest visits node 7 as free-gratis (FG). Node 7 uses it to transmit to node 1, thus converting the slot status to busy-gratis (BG). Let us assume the scheduler is in the process of marking slots as reserved so that our slot is changed to

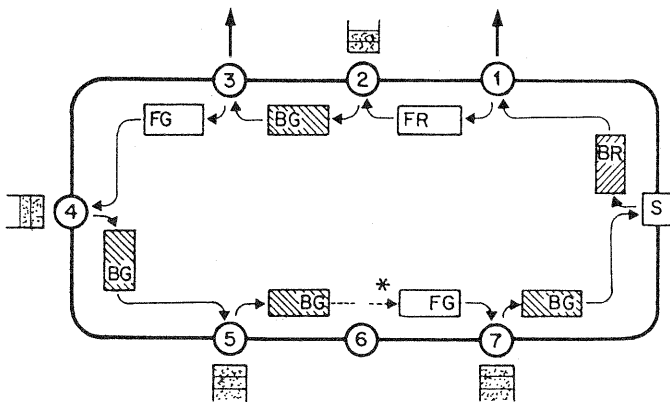


Fig. 11: Slot status dynamics

busy-reserved (BR). From now on, the slot can only be used by a node with confirmed reservations which we shall henceforth denote as confirmations. When the slot reaches its destination, node 1 changes the slot status to free-reserved (FR) if, as is the case here, it has no confirmations. Node 2, which has pending slot transmissions and confirmations, then accesses the slot, thereby rendering it busy-gratis (BG). Assuming this data is destined for node 3, the slot becomes free-gratis (FG) at that node. Now, node 3 itself or any downstream node can use it. This dynamic slot status shows that during one circulation, a single slot may be accessed several times although it was reserved for only one transmission.

The time-space diagram in Figure 12 shows consecutive circulations of a slot as it travels from the scheduler around the ring and back. Each of the seven node positions is represented by a horizontal line. We describe slot state transitions that have not yet been covered by the previous example. A free-reserved (FR) slot arrives at node 1 that has confirmations but no more data to send as either all or its remaining transmissions were sent in gratis slots. Node 1 therefore converts the slot to free-gratis (FG) so that it becomes available for transmission by node 2 to node 4. The slot is then accessed by node 7 for a broadcast message. Thereafter, the scheduler marks it so that its status becomes busy-reserved (BR). Broadcast implies source removal so that node 7 changes the returning slot status to free-reserved (FR) since we assume it has no outstanding confirmations. The slot then passes the scheduler, unmodified this time since it is already marked as reserved, to be accessed finally by node 2 for transmission to node 6. Assuming the scheduler has terminated reservation marking, the free-gratis (FG) set by node 6 remains unchanged and the slot is subsequently used by node 1.

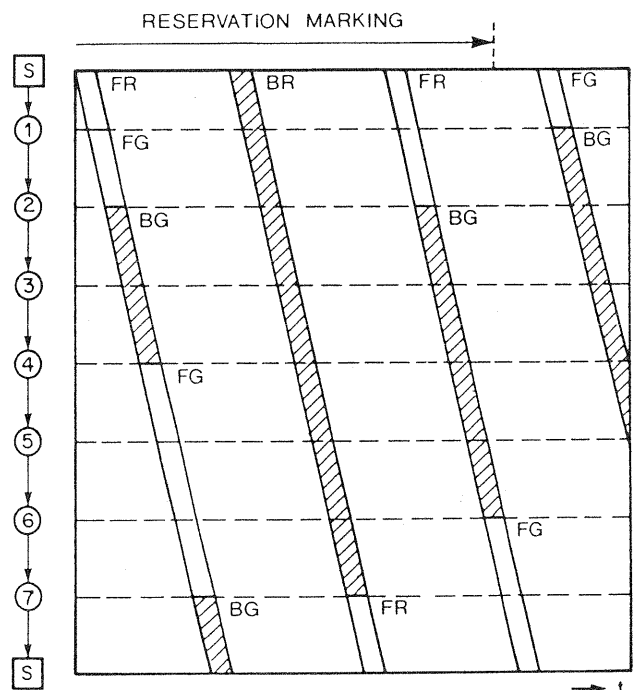


Fig. 12: Time-space diagram showing slot status dynamics

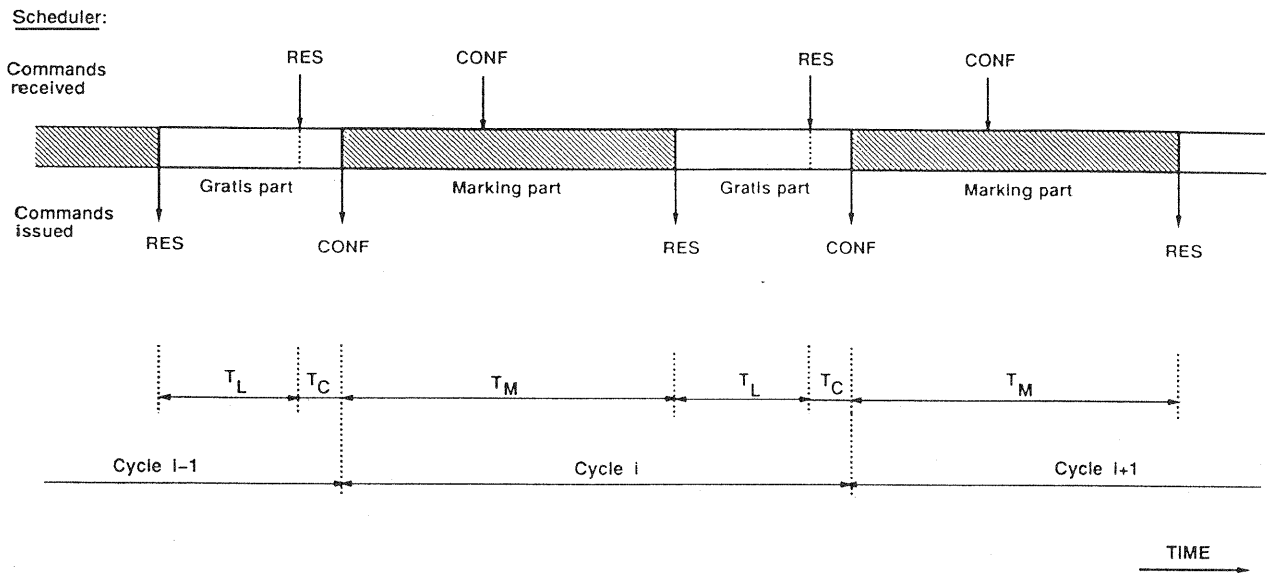


Fig. 13: Reserve/Confirm command sequence

Figure 13 shows the alternating Reserve/Confirm command sequence as seen by the scheduler and defines the different parts of the cycle. The arrows at the lower side of the time diagram represent commands issued by the scheduler. The arrows at the upper side are those received. To collect reservation requests for cycle  $i$ , the scheduler issues a Reserve command. On its return after one network latency  $T_L$ , processing of the Reserve command begins. After a processing time  $T_C$ , the Confirm command is issued which starts then the reservation cycle  $i$ . This command returns to the scheduler after network latency  $T_L$ . Immediately after sending the Confirm command, the scheduler starts to mark all passing gratis slots as reserved until the number of marked slots corresponds to the sum of confirmations allocated to the nodes. We call this the marking part of the cycle. Occasionally, free-reserved slots (as discussed in Figure 12) return to the scheduler which simply lets them pass. Each passing free-reserved slot extends the marking time  $T_M$  by one slot transmission time. When marking finishes, the Reserve command is issued to collect reservations for the next cycle ( $i+1$ ). No further marking occurs until the next Confirm command is issued. During this period, most of the slots passing the scheduler are gratis (busy or free) but can include returning free-reserved slots. Thus during reserve collection and processing the network remains fully utilized whereby fairness enforcement is always ensured in the next cycle. In the case that an empty Reserve command returns to the scheduler, it is immediately issued again.

##### 5. Buffer insertion-based access

We begin with a short review of buffer insertion-based access where an insertion buffer is placed in the data path of each node. Figure 14 shows the location of the receive buffer, the insertion buffer and the transmit buffer. As long as the insertion buffer is empty, passing data is cut through and no insertion delay is involved. When data from an upstream source arrives at a transmitting node, it

is held up in that node's insertion buffer until its transmission is terminated. From this point on, all passing data is delayed until the insertion buffer becomes empty. The delay experienced corresponds to the amount of data held up during the node's frame transmission. An underlying condition is that the insertion buffer is greater than or equal to the maximum frame size. Frame (packet) transmission can start when the node detects the medium to be idle and its insertion buffer is empty. More generally, one could allow transmission to begin when the insertion buffer occupancy is below a given threshold. After accessing the medium, data traverses the insertion buffers of intermediate nodes until it arrives at the destination which removes it prior to entering that node's insertion buffer. Buffer insertion has the inherent advantage of immediate access in a lightly loaded environment while maintaining frame transmission contiguity for all load conditions. A heavy user, however, can induce unfair behavior by not letting downstream nodes empty their insertion buffers thus hindering transmission. Fairness enforcement such as that described in [18, 27] is therefore necessary. CRMA-II offers an alternative fairness solution.

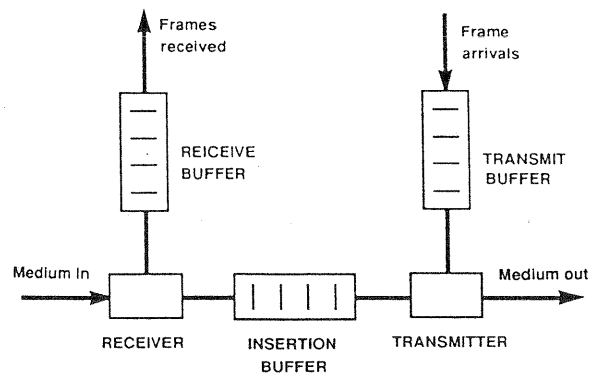


Fig. 14: Node with insertion buffer



Using a reservation-based fairness algorithm implies operating in slots as units of reservation. Thus all operations are related to slot boundaries. A node with an empty insertion buffer begins transmission only when a free slot arrives. In this way, it is avoided that frames passing a node are interrupted. A node empties its insertion buffer by discarding free-gratis slots and free-reserved slots when the node has confirmations (confirmed reservations). Nodes with confirmations, i.e. nodes whose confirm count is greater than zero, access free-reserved slots as free slots whereby each such access decrements the confirm count. Non-reserving nodes or nodes that have used up their confirmations, i.e. those whose confirm count has reached zero, treat free-reserved slots as busy. **Figure 15** demonstrates how the reservation mechanism establishes fairness. Consider an example where the upstream node, shown on the left-hand side, is a heavy user with its transmit buffer permanently loaded. During the time period of interest, we assume that the frame transmissions of the heavy user do not interfere with frames from upstream, so that its insertion buffer remains empty. Thus the user at the downstream node cannot transmit because of a permanent non-empty insertion buffer. Consequently, the downstream node makes reservations to empty both its transmit and insertion buffer. The posted reservation value is seven: three for transmission and four to empty its insertion buffer. As a result of the reservation process, only five slots are allocated by the scheduler so that the confirm count of the downstream node becomes five. At time  $t = t_0$ , the upstream node's insertion buffer has been filled up with four free-reserved slots. Then because of its zero confirm count, the upstream node cannot use these reserved slots and treats them as busy slots. The upstream node continues however its frame transmission in free-gratis slots generated on its own. As the frame transmission terminates at time  $t = t_0 + 1$ , the backlogged free-reserved slots can flow into the downstream node's insertion buffer and empty it. Every time a free-reserved slot enters, it decrements its confirm count. Eventually, the insertion buffer empties at time  $t = t_0 + 5$  so that at time  $t = t_0 + 6$ , the next passing free-reserved slot initiates the frame transmission at the downstream node. Since the insertion buffer of the upstream heavy user is again empty, it also transmits the first slot of its next frame. The heavy-user's frame is however held up in the insertion buffer of the downstream node until it ends its transmission.

This example also shows that a frame transmission in progress continues regardless of the status of arriving slots or the value of the confirm count. All arriving busy slots and free-reserved slots, when the confirm count is zero, are held up in the insertion buffer. Each time, no free slots from the medium can be used, the node generates free-gratis slots on its own thereby temporarily increasing the total amount of slots in the network. Continuation of frame transmission in case of a zero confirm count, does not impact fairness because the gratis slots used are taken into account in the next reservation cycle. The fact that nodes with and without an insertion buffer use gratis and reserved slots in a similar way, is the reason that both node types can coexist on a single network.

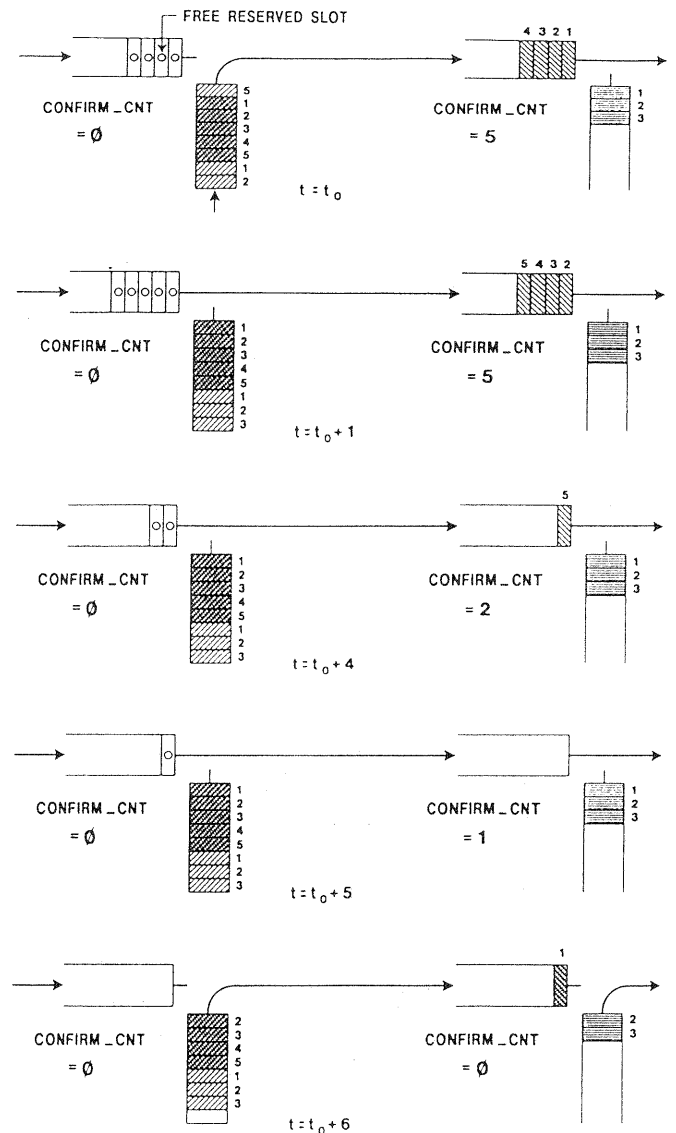


Fig. 15: Buffer insertion with reservation fairness

## 6. Scheduler operation

The scheduler function, i.e. computation of the Confirm command parameters, is the mechanism for ensuring fairness and bounded access delays. The reservation is based on slot units whereby the actual frame boundaries are not considered. Owing to this, the same scheduling algorithm applies both to slotted and slotted buffer-insertion operation. During the processing time of a received Reserve command, full network utilization can be maintained. In other words, scheduling is a separate process from slot transmission and can be executed at the speed of the node's MAC microcontroller without the need for dedicated hardware. This also means that different scheduling algorithms can be executed on the same hardware.

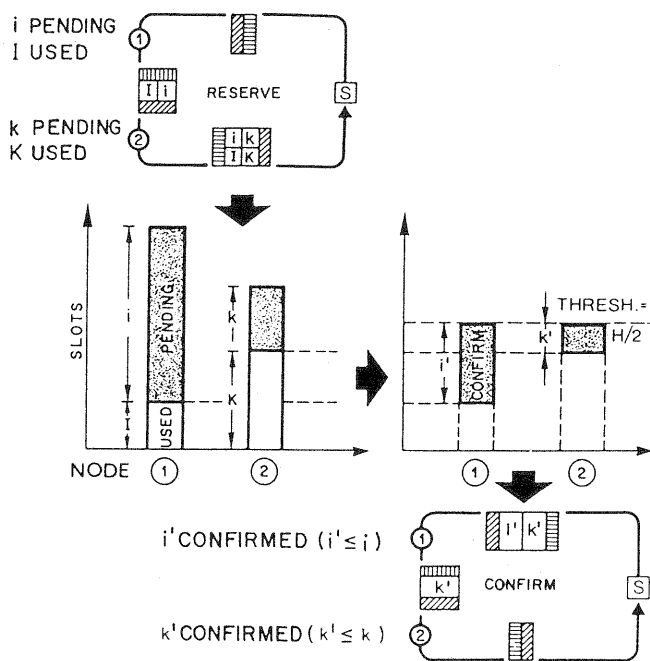


Fig. 16: Scheduling algorithm example

For illustrative purposes, we consider the simple algorithm presented in **Figure 16**. It describes a two-node reservation followed by slot reservation allocation and the confirmation to the nodes involved. Reservation consists of the nodes issuing two parameters: first the pending transmission requests  $i$  and  $k$ , secondly the already completed transmissions  $I$  and  $K$  via gratis slots since the previous Reserve command. Confirmations are computed at the scheduling node by adding these two parameters for each node and then truncating the results at a threshold. The parameter  $H$  represents the aggregate network capacity in slots for a selected period of time over which fairness is to prevail. The threshold is obtained by dividing  $H$  by the number of reserving nodes, in our case 2. The confirmed values  $i'$  and  $k'$  are then derived by subtracting the corresponding  $I$  and  $K$  values from the threshold. This mechanism ensures that a node having already accessed many gratis slots will receive a correspondingly smaller number of confirmations so as to give less fortunate nodes a larger share. Figure 16 would lead one to believe that the pending transmit parameters  $i$  and  $k$  are superfluous. These parameters are needed when, for example,  $i$  and  $I$  add up to less than the threshold, in which case  $i'$  is set equal to  $i$ .

The scheduler must also sum up the number of reserved slots needed to honor the issued confirmations. In the above example this number is  $z = i' + k'$  which is loaded into the mark-counter. It is then decremented each time an incoming gratis slot is marked as reserved. Marking terminates when the counter becomes zero.

### 7. Node operation

We next describe the related MAC operations in a node. Arriving frames are segmented into units of slot payload before being enqueued in the transmit buffer. The resulting slot number is added to the pending transmit

counter which is used for subsequent reservations. Individual frame boundaries only play a role when a frame is being transmitted via buffer insertion. Each transmission of a payload unit causes the pending transmit counter to be decremented. In each node, two further counters are required for CRMA-II operation:

- **gratis counter  $g$**  indicates the number of gratis slots used since the previously passing Reserve command;
- **confirm counter  $c$**  indicates the number of confirmed slot reservations (confirmations).

Upon arrival of a busy slot, its payload is copied when there is a destination address match. Additionally, the node compares the removal address to decide whether the slot is to be made free. If so, the slot becomes available for the removal node itself or its downstream nodes whereby the free-reserved or free-gratis slot is to be accessed according to the protocol rules. When concatenated slots belonging to a frame are freed, individual slots are restored, i.e. set free plus reformatted as shown in Figure 6. A node accesses free-gratis slots (if available) as long as its pending transmit count is nonzero or uses them to empty its insertion buffer. For each gratis slot used the gratis counter is incremented. When the Reserve command passes, the node inserts the pending transmit count plus the insertion-buffer occupancy count together with the gratis count into the command. However, it does so only when there is a need for reservation (i.e. for transmissions and/or emptying the insertion buffer). The gratis counter is always reset by the Reserve command. When the Confirm command passes, the confirm counter is loaded provided the node previously made reservations. Now, the node can also use free-reserved slots, either for transmission or for emptying the insertion buffer whereby the confirm counter is decremented each time. When this counter becomes zero only free-gratis slots can be used. Finally, if the pending transmit count is zero and the insertion buffer is empty then the confirm counter still continues to be decremented by each passing free-reserved slot until it becomes zero. In this way a surplus of free-reserved slots are changed to free-gratis.

### 8. Operation on ring and bus network

The principle operation of the MAC protocol on a ring is illustrated in **Figure 17**. It shows the slot state transitions at the scheduler and at a source-destination node pair. The reservation mechanism and loading or resetting counters have not been included. The scheduler marks passing gratis slots (free or busy) as reserved as long as the mark counter  $m$  is nonzero. For each reservation marking of a slot, the mark counter is decremented. Slots passing as free-gratis (FG) become free-reserved (FR) and busy-gratis (BG) reverts to busy-reserved (BR). Slots already marked simply pass (these are occasionally returning free-reserved slots as discussed for Figure 12). When the mark counter is zero all slots pass unchanged. The source node transmits in free-gratis slots or in free-reserved slots when its confirm counter  $c$  is nonzero. In both cases the slot status changes to busy-gratis. This allows the scheduler to mark the slot as reserved when it is still busy. For all free-gratis slots used, the gratis counter  $g$  is incremented

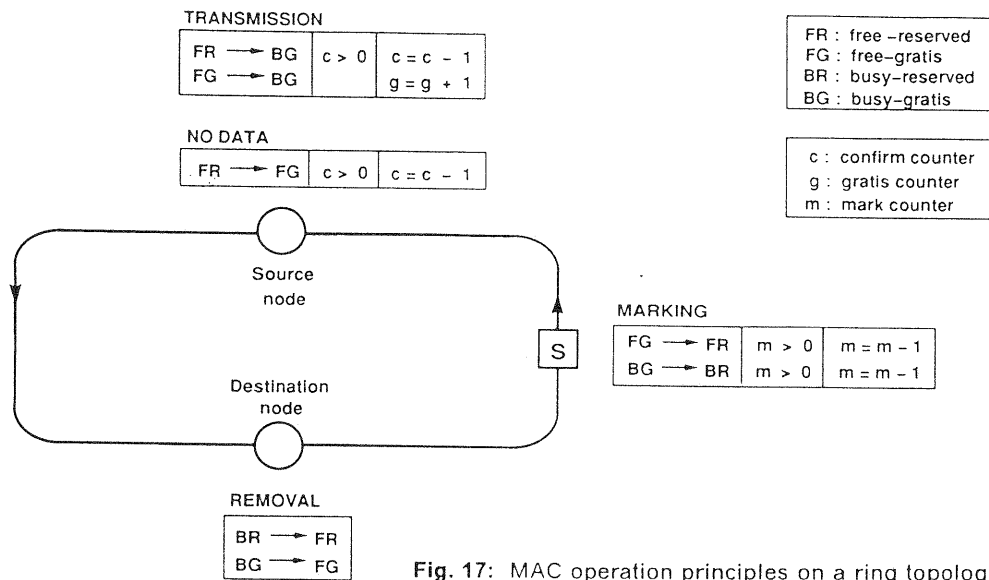


Fig. 17: MAC operation principles on a ring topology

whereas the confirm counter  $c$  is decremented when using a free-reserved slot. The decrementing process stops when the counter becomes zero. For the further discussion, we divide free-reserved slots into accessible ( $c > 0$ ) and inaccessible ( $c = 0$ ) slots. A source node always lets busy and inaccessible free-reserved slots pass. Furthermore, in the case that the source node has a surplus of confirmations (i.e. no data to transmit because all its transmissions occurred in gratis slots), the source node turns free-reserved slots into free-gratis and decrements its confirm counter. Upon arrival at the destination (removal) node, busy slots become free. A busy-gratis slot changes into free-gratis, whereas a busy-reserved slot reverts to free-reserved so that it can be accessed by a node having confirmations. **Figure 18** shows the same scenario for a dual bus topology. In principle, there is only one difference. The slots leaving the headend are always free so that they are free-reserved during the marking process and otherwise free-gratis. Therefore the slots will never be in a busy-reserved state.

**Figure 19** illustrates the operation of a source node with an insertion buffer. Operations of the scheduler and the destination node remain as discussed above. The operation is identical for ring and bus topologies when one takes into account that on a bus topology busy-reserved slots never exist. The slot handling in nodes with an insertion buffer is done in two independent steps. First the insertion buffer part takes place, then the actual slot access decision is taken. The tables next to the illustrations summarize the details related to the status of arriving slots for the three situations described. The tables specify the operation at the buffer input side (action in), the mode of buffering, the output side operation (action out), and the change in insertion buffer occupancy  $b$ . The tables further consist of the transmit condition (TX) and the corresponding counter modification which concerns either the gratis counter  $g$  or the confirm counter  $c$ . These counters change when the node transmits or empties its insertion

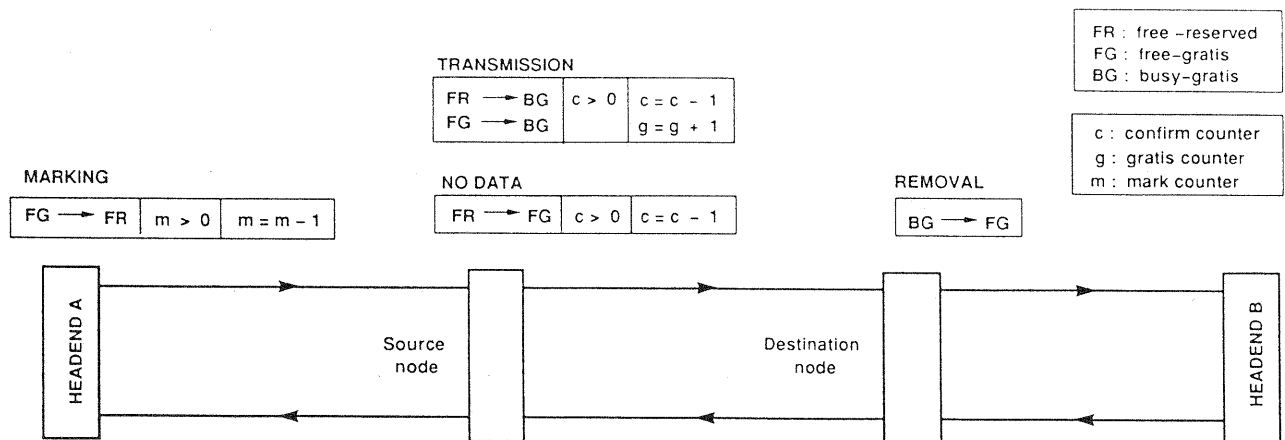
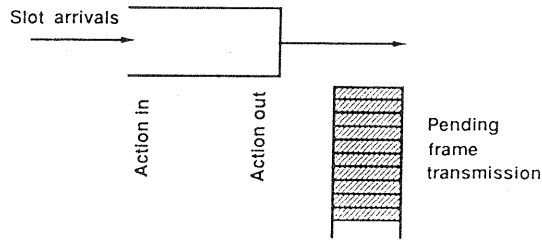


Fig. 18: MAC operation principles on a bus topology

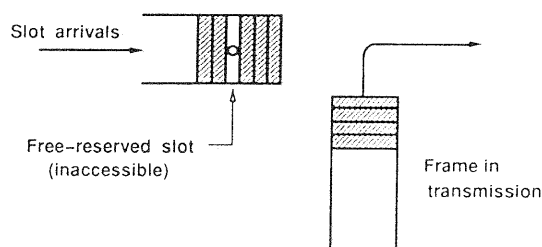
① **FRAME WAITING FOR TRANSMISSION**  
(empty insertion buffer)



| Slot status |           | Action In | Buffering   | Action Out | Occupancy | TX     | Counters    |
|-------------|-----------|-----------|-------------|------------|-----------|--------|-------------|
| FR          | $(c > 0)$ | enqueue   | cut through | dequeue    | -         | $TX_0$ | $c = c - 1$ |
| FG          | -         | enqueue   | cut through | dequeue    | -         | $TX_0$ | $g = g + 1$ |
| FR          | $(c = 0)$ | enqueue   | cut through | dequeue    | -         | -      | -           |
| BR          | -         | enqueue   | cut through | dequeue    | -         | -      | -           |
| BG          | -         | enqueue   | cut through | dequeue    | -         | -      | -           |

Frame waiting for transmission (empty insertion buffer)

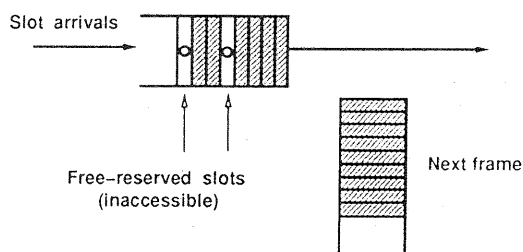
② **FRAME IN TRANSMISSION**



| Slot status |           | Action In | Buffering | Action Out | Occupancy   | TX | Counters    |
|-------------|-----------|-----------|-----------|------------|-------------|----|-------------|
| FR          | $(c > 0)$ | discard   | -         | generate   | -           | TX | $c = c - 1$ |
| FG          | -         | discard   | -         | generate   | -           | TX | $g = g + 1$ |
| FR          | $(c = 0)$ | enqueue   | delay     | generate   | $b = b + 1$ | TX | $g = g + 1$ |
| BR          | -         | enqueue   | delay     | generate   | $b = b + 1$ | TX | $g = g + 1$ |
| BG          | -         | enqueue   | delay     | generate   | $b = b + 1$ | TX | $g = g + 1$ |

Frame in transmission

③ **FRAME WAITING FOR NEXT TRANSMISSION**  
(non-empty insertion buffer)



| Slot status |           | Action In | Buffering | Action Out | Occupancy   | TX | Counters    |
|-------------|-----------|-----------|-----------|------------|-------------|----|-------------|
| FR          | $(c > 0)$ | discard   | -         | dequeue    | $b = b - 1$ | -  | $c = c - 1$ |
| FG          | -         | discard   | -         | dequeue    | $b = b - 1$ | -  | $g = g + 1$ |
| FR          | $(c = 0)$ | enqueue   | delay     | dequeue    | -           | -  | -           |
| BR          | -         | enqueue   | delay     | dequeue    | -           | -  | -           |
| BG          | -         | enqueue   | delay     | dequeue    | -           | -  | -           |

Frame waiting for next transmission (non-empty insertion buffer)

FR : free-reserved                      b : insertion buffer occupancy  
 FG : free-gratis                        c : confirm counter  
 BR : busy-reserved                      g : gratis counter  
 BG : busy-gratis

Fig. 19: MAC operation principles with buffer insertion

buffer. In the description we differentiate between enqueueing with cut through and enqueueing with delay. In the first case, the same slot is dequeued immediately. Otherwise the head-of-the-queue slot is taken. In general all slots are cut through the insertion buffer when it is empty. Generation of additional free-gratis slots occurs when an arriving slot on the medium is enqueued in the insertion buffer during frame transmission. The transmission of the first slot of a frame is indicated by  $TX_0$ , further transmissions by TX. Accessing slots for transmission means setting free-gratis or accessible free-reserved slots to busy-gratis and loading the payload into the slot.

In the upper part of the figure, the node of interest is ready for frame transmission while its insertion buffer is

empty. As long as busy slots are cut through the buffer, the node must refrain from starting its transmission. This holds also for inaccessible free-reserved slots which are treated as busy slots. Transmission starts however in the first arriving slot that is free-gratis or accessible free-reserved. When a frame transmission has been initiated, transmission continues regardless of the status of arriving slots. The operations during frame transmission are shown in the middle part of the figure. Free-gratis and accessible free-reserved slots are discarded at the input side of the insertion buffer and free-gratis slots are generated at its output side. Thus in fact free slots from the medium are used and the insertion buffer occupancy remains the same. Arriving busy slots and inaccessible free-reserved slots are held up in the insertion buffer.

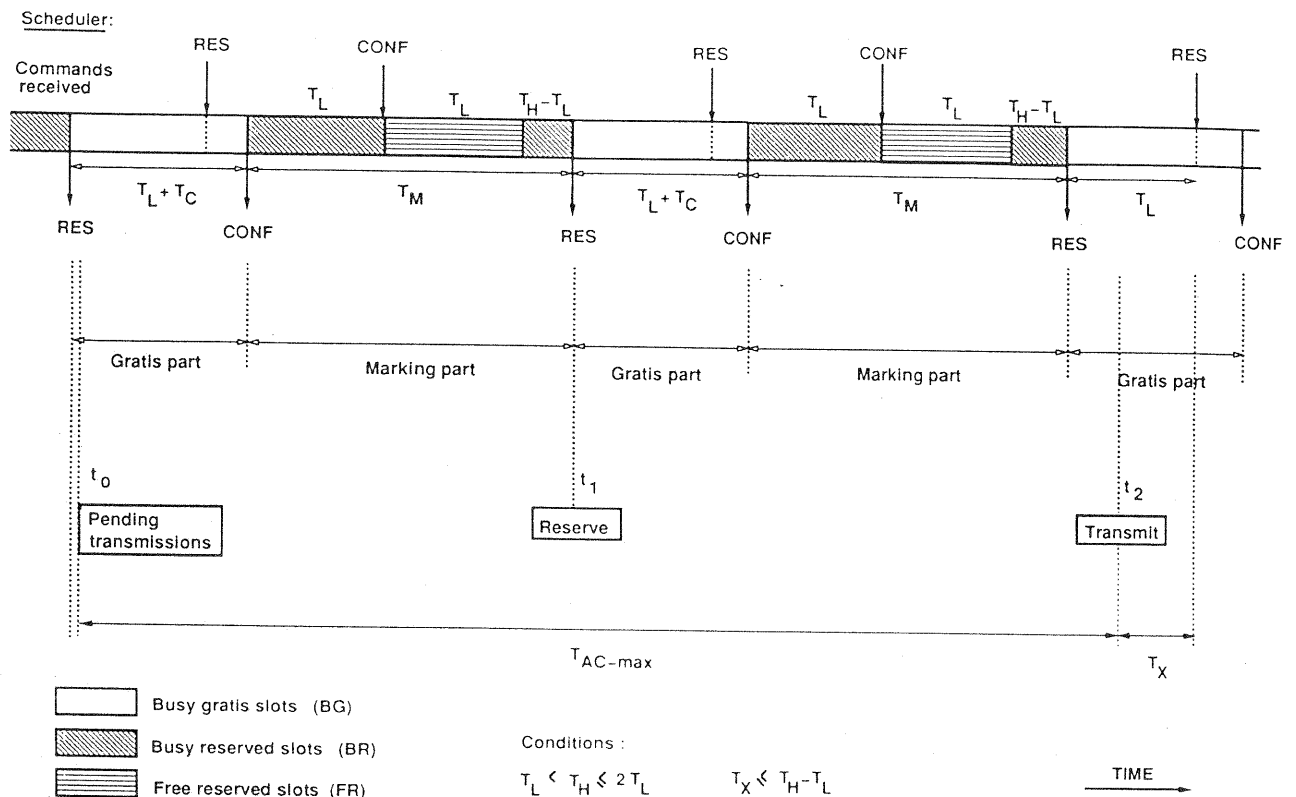


Fig. 20: Worst-case access delay scenario (ring topology)

Each time this happens, the node generates an extra free-gratis slot on its own. Thus the contiguous slots used to transmit a frame might consist of cut through slots and additionally generated slots. Before the node is permitted to transmit its next frame, the insertion buffer must be emptied. Buffer emptying is shown in the lower part of the figure. The insertion buffer is emptied either by free-gratis slots or accessible free-reserved slots. Here the reservation-based fairness mechanism comes into play because the number of gratis slots used (already on the medium or generated) might influence the number of allocated confirmations. So insertion buffer emptying might be slowed down when a node exceeds its reserved slot allocation during heavy load. We recall that a node makes reservation requests both to transmit and to empty its insertion buffer. The buffer is emptied by discarding an arriving free-gratis or accessible free-reserved slot and dequeuing the head-of-the-queue slot from the buffer. The dequeued slot subsequently travels to the next node. An arriving busy slot or an inaccessible free-reserved slot is enqueued in the insertion buffer while the head-of-the-queue slot is simultaneously dequeued. Thus the amount of buffered slots remains unchanged. We finally want to point out that in an implementation, discarding of free slots means not clocking ADUs into a dual-port RAM buffer, in contrast to enqueueing where ADUs are clocked into it. Analogously, by not clocking ADUs out of the insertion buffer, slots are held up. Clocking ADUs out causes dequeuing of slots.

## 9. Access delay bounds

One of the features of reservation-based access is that it enforces tight bounds on the access delay during heavy load periods. In this section, we consider these bounds for the slotted operation. Figure 20 illustrates a worst-case access-delay scenario for a group of confirmed cell reservations. Let us consider a node located just behind the scheduler that generates new pending slot transmissions at time  $t_0$ , just after a Reserve command has passed by. We further assume that no free-gratis slots are available to the considered node. A simple scenario creating this situation is a heavy user located just before the scheduler and sending all its data to a downstream node of the node in question. Because of the heavy user, all gratis slots seen by the scheduler and the considered node are busy. To make reservations, the node must first wait for the next Reserve command. This delay is the command roundtrip delay  $T_L$  followed by the Confirm computation delay  $T_C$  and the marking delay  $T_M$  needed by the scheduler to mark  $H$  slots (maximum slot number marked reserved within one reservation cycle). Thus reservation takes place at time  $t_1$  after a total delay of  $T_L + T_C + T_M$ . The most unfavorable case is when the node in question is the last one to be serviced so that another delay of  $T_L + T_C + T_M$  occurs. Again in the most unfavorable situation the slots in question are marked busy-reserved so they can only be accessed one ring latency  $T_L$  later at time  $t_2$  when they finally arrive as

free-reserved. The access waiting time terminates when transmission begins, thus the derived delay needs to be diminished by  $T_X$ , the transmission time for the confirmed slots.

Therefore on a ring, the worst-case access delay becomes

$$T_{AC-max} = 2(T_L + T_C + T_M) + (T_L - T_X) \quad (1)$$

with

$$T_M = T_H + nT_L,$$

$$n = \text{integer} \left[ \frac{T_H - T_S}{T_L} \right] = 0, 1, 2, \dots, \quad (2)$$

$$T_H = H \cdot T_S, \quad T_L > 0, \quad T_X \leq T_L, \quad H = 1, 2, \dots,$$

where  $T_S$  represents one slot transmission time and  $T_H$  is the time required by the scheduler to mark  $H$  consecutive slots as reserved. Relations (2) give the maximum marking time  $T_M$  required by the scheduler to mark  $H$  slots. The cause for the integer  $n$  quantization effect is as follows. All slots passing the scheduler while a Confirm command propagates around the ring are busy-gratis, since by that time, each slot marked as reserved in the previous reservation cycle has been accessed, i.e. set to gratis. Consequently, the scheduler marks all slots fol-

lowing the Confirm command. Let us assume a worst case where these slots maintain their busy-reserved status when passing the nodes having confirmations. Then destination removal will ensure a return of these slots (to the scheduler) as free-reserved. The scheduler refrains from marking these as they are already marked. It therefore interrupts its marking process until all these free-reserved slots have passed. This interruption lasts one ring latency with the process repeating itself when the scheduler resumes marking. Thus scheduler marking in this worst-case scenario consists of an active marking period followed by a pause followed by a marking period and so on until all  $H$  markings have occurred. This results in  $n$  pauses each of which lasts one ring latency  $T_L$ . Figure 20 portrays this effect for  $n = 1$ , corresponding to the condition  $T_L < T_H \leq 2T_L$ .

Although the derivation is based on a node location just behind the scheduler, the delay bounds are independent of the node position on the ring. Moreover, the bounds are not a function of the number of simultaneously accessing nodes. Figure 21 shows the access delay bounds in ms versus the total number of confirmed slots  $H$  as given by relations (1) and (2) for a 20 and 100 km ring. The values relate to a transmission net rate of 1 Gbit/s and 64-byte slots. This corresponds roughly to ten slots per km, each with a duration of  $0.5 \mu s$ . Processing and transmission times have been neglected because they have only a minor effect on the delay bounds. The curves reflect the previously discussed quantization effect.

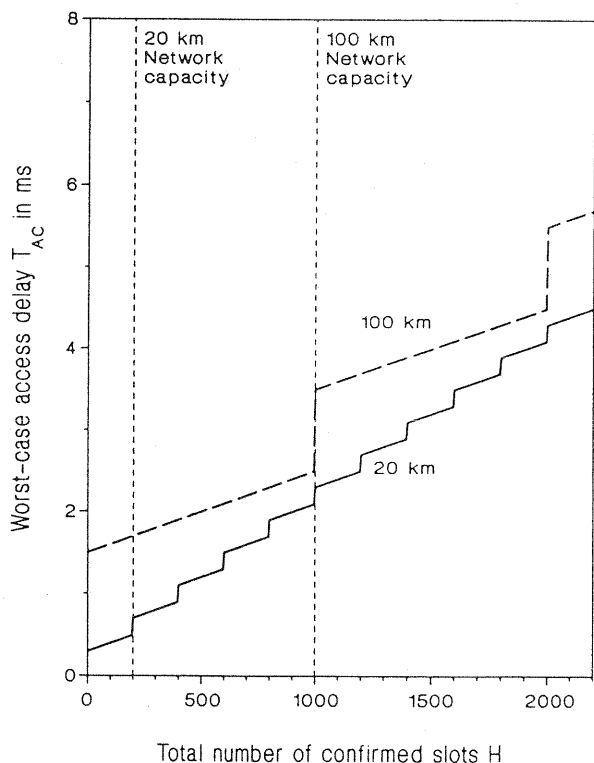


Fig. 21: Worst-case access delay bounds (ring topology)

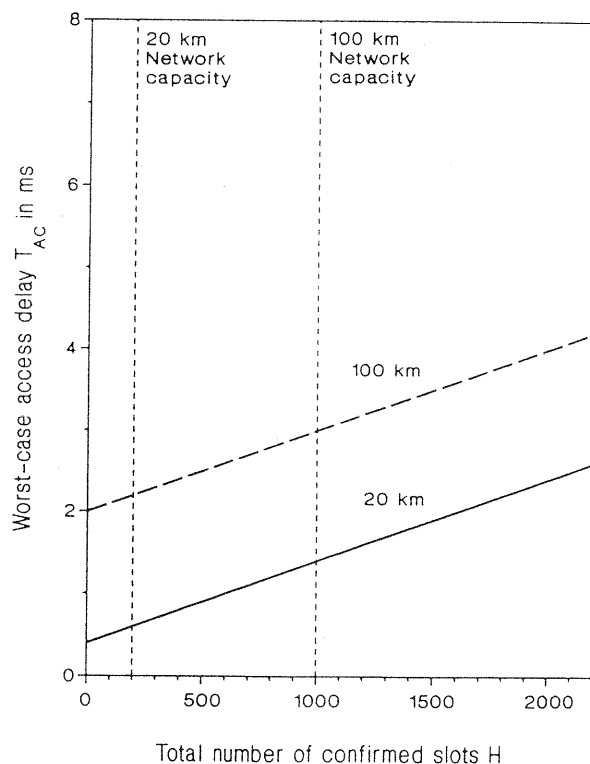


Fig. 22: Worst-case access delay bounds (bus topology)

The derivation for a bus topology is similar. Two differences have to be taken into account. First, the headend generates the free slots and reserved slots are marked consecutively. Thus the marking time  $T_M$  is equal to  $T_H$ . Second, the last latency  $T_L$  in Figure 20 does not exist because the reserved slots are always free and must not circulate a second time as in a worst-case ring scenario. Thus the worst-case access delay for a bus becomes

$$T_{AC-max} = 2 (T_L + T_C + T_H) - T_X \quad (3)$$

The roundtrip delay  $T_L$  is the roundtrip of a command relayed by the other headend. For a comparison with a ring covering the same geographical area we must consider that the command roundtrip delay  $T_L$  for a bus is twice as long as for a ring.

Figure 22 shows the corresponding worst-case access delay bounds for a bus topology where the total length of the forward and the reverse bus is 40 km and 200 km, respectively. The curves exhibit a higher delay at low values of  $H$  because the roundtrip delay of a Reserve command is twice as long. Moreover the curves increase strictly linearly because of the absence of marking pauses.

## 10. Priorities

Support of multiple priorities is conceptually very simple. It consists of replicating the described access mechanism as many times as the number of supported priorities. Each priority level executes its own Reserve/Confirm command handshake whereby no special coordination is necessary between commands belonging to different priorities. The only dependency enforced by the scheduler is that if mark-counters for different priorities are nonzero then marking for the highest priority takes precedence, whereby a slot's priority field is contained in both the start and end ADU. Note that when a slot has been marked for a given priority, that marking is never changed. Thus if a busy-gratis slot is marked busy-reserved and happens to return as free-reserved, it passes the scheduler unchanged, even when a higher priority marking is currently taking place.

A higher priority command will interrupt a lower priority scheduling process by initiating its correspondingly higher priority scheduler. Thus no added hardware is needed to support multiple priorities at the scheduler, but nodes clearly require a separate transmit queue per priority. When a free-reserved slot is accessed, it is changed to a busy-gratis slot which has no defined priority. Therefore all nodes can reuse previously reserved slots when they become free-gratis regardless of what the original priority was. Owing to gratis access and spatial reuse, nodes can have confirmations for a given priority but no more data for that priority. In that case, the nodes in question change the corresponding prioritized slots to free-gratis.

## 11. Robustness in the presence of errors

The transmission data structure with embedded slots allows synchronization on every slot and is particularly robust through simple finite state machines supervising matching delimiter pairs. The repetitive Reserve/Confirm command sequence is robust against all single and mul-

iple transmission errors. Bit changes in the contents of these commands only temporarily affect the performance of the active nodes whereby Confirm commands implicitly reset the new cycle, thus providing a regeneration point.

## 12. Conclusion

CRMA-II is a generalization of the reservation-based MAC principles of the original bus-based CRMA protocol to both ring and bus topologies. It combines good performance features, suitability for multi-Gbit/s implementation, and protocol robustness. CRMA-II exploits the fast access capability of a simple busy/free slot-access protocol when a node experiences low to medium load conditions. By combining this fast or immediate access capability with a reservation mechanism, CRMA-II ensures that the unique properties of the simple busy/free access are integrated into a MAC protocol that enforces fairness and permits priorities to be effective during heavy load periods. As the amount of confirmed reservations is kept to the minimum value so as to ensure fairness, tight upper bounds on access delay can be maintained for low-traffic users. Delay and throughput are further enhanced by spatial reuse. The combination of gratis and reserved access is fundamental to achieving simultaneously sustained high-network utilization, enforced fairness and fast response for low-traffic users or priority requests. The slot-reservation principle used in CRMA-II applies both to a slotted and a slotted buffer-insertion operation. Moreover, nodes with and without an insertion buffer can coexist on a single CRMA-II network. Thus any combination of frame-oriented LAN traffic and cell-oriented ATM traffic can be supported. In addition, the unified MAC protocol enables one common node design so that nodes can be attached to ring and bus networks operating with or without buffer insertion. The medium access is based on a simple Reserve/Confirm command sequence issued by a node executing the scheduling function. This function can be microcontroller-based so that no extra hardware is required. As part of the network reconfiguration capability, the scheduler function is present in each node. The novel MAC-command transmission technique is based on experience gained with prototyping the earlier CRMA version at Gbit/s speed. It is tailored for a high clock-rate implementation and minimizes the amount of fast logic.

## Acknowledgement

The authors would like to thank F. Closs and D.A. Pitt for their support, involvement and suggestions, M.M. Nassehi for his contributions during the extension of CRMA to a ring topology, as well as W. Bux for his suggestions on improving the paper.

## References

- [1] M.M. Nassehi, "CRMA: An Access Scheme for High-Speed LANs and MANs," IEEE International Conference on Communications, ICC 90, Atlanta, GA, April 1990, pp. 1697-1702.
- [2] H.R. Müller, M.M. Nassehi, J.W. Wong, E. Zurfluh, W. Bux, P. Zafiropulo, "DQMA and CRMA: New Access Schemes for Gbit/s LANs and MANs," IEEE INFOCOM 90, San Francisco, CA, June 1990, pp. 185-191.

- [3] M.M. Nassehi, "Cyclic Reservation Multiple-Access Scheme for Gbit/s LANs and MANs based on Dual-Bus Configuration," Eighth European Fibre Optic and Local Area Networks Conference, EFOC/LAN 90, Munich, June 1990, pp. 246-251.
- [4] E.A. Zurfluh, "A Gbit/s LAN/MAN Prototype Implementation of the CRMA Protocol," Fourth IEEE Workshop on Metropolitan Area Networks, Fort Myers, FL, November 4-7, 1990.
- [5] L.J. Rosenblum (Section Editor), "Scientific Visualization at Research Laboratories," IEEE Computer, Vol. 22, No. 8, Aug. 1989, pp. 68-101.
- [6] J. Kohli, "Medical Imaging Applications of Emerging Broadband Networks," IEEE Communications Magazine, Vol. 27, No. 12, Dec. 1989, pp. 8-16.
- [7] A. Karmouch, L. Orozco-Barbosa, N.D. Georganas, M. Goldberg, "A Multimedia Medical Communications System," IEEE J. Select. Areas Commun., Vol. SAC-8, No. 3, April 1990, pp. 325-339.
- [8] T.D.C. Little, A. Ghafoor, "Network Considerations for Distributed Multimedia Object Composition and Communication," IEEE Network Magazine, Vol. 4, No. 6, Nov. 1990, pp. 32-49.
- [9] W.P. Lidinsky, "Data Communications Needs," IEEE Network Magazine, Vol. 4, No. 2, March 1990, pp. 28-33.
- [10] P. Zafiropulo, "On LANs and MANs : An Evolution from Mbit/s to Gbit/s," Eighth European Fibre Optic and Local Area Networks Conference, EFOC/LAN 90, Munich, June 1990, pp. 15-22.
- [11] A. Hopper, "The Cambridge Ring - A Local Network," in Advanced Techniques for Microprocessor Systems, F.K. Hanna (Editor), Peter Peregrinus, Ltd., Stevenage, UK, 1980, pp. 67-71.
- [12] J.O. Limb, C. Flores, "Description of Fasnet - A Unidirectional Local Area Communications Network," BSTJ, Vol. 61, No. 7, Sept. 1982, pp. 1413-1440.
- [13] R.M. Falconer, J.L. Adams, "Orwell: A Protocol for an Integrated Services Local Network," British Telecom Technology Journal, Vol.3, No.4, Oct. 1985, pp. 27-35.
- [14] T.J. King, I. Gallagher, "Orwell: A Multiservice Network Protocol," Eighth European Fibre Optic and Local Area Networks Conference, EFOC/LAN 90, Munich, June 1990, pp. 91-96.
- [15] A. Patir, T. Takahashi, Y. Tamura, M. El Zarki, A.A. Lazar, "An Optical Fiber-Based Integrated LAN for MAGNET's Testbed Environment," IEEE J. Select. Areas Commun., Vol. SAC-3, No. 6, Nov. 1985, pp. 872-881.
- [16] H. Ohnishi, N. Morita, S. Suzuki, "ATM Ring Protocol and Performance," IEEE International Conference on Communications, Boston, MA, June 1989, pp. 394-398.
- [17] A. Hopper, R.M. Needham, "The Cambridge Fast Ring Network System," IEEE Trans. Comp., Vol. 37, No. 10, Oct. 1988, pp. 1214-1223.
- [18] P. Heinzmann, H.R. Müller, D.A. Pitt, H.R. van As, "Buffer-insertion Cell-synchronized Multiple Access (BCMA) on a Slotted Ring," Second International Conference on Local Communications Systems: LAN and PBX, June 26-28, 1991, Palma, Balearic Islands, Spain.
- [19] "Proposed Standard Distributed Queue Dual Bus (DQDB) Subnetwork of a Metropolitan Area Network (MAN)," IEEE Unapproved Draft P802.6/D15, Oct. 1990.
- [20] U. Rossi, A. Trondoli, F. Ferrero, A. Luvison, P. Roffinella, "LION: An Innovative Approach to Multi-Service Business," IEEE Global Telecommunications Conference, GLOBECOM 89, Dallas, TX, Nov. 1989, pp. 1308-1312.
- [21] A.A. Lazar, A.T. Temple, R. Gidron, "MAGNET II: A Metropolitan Area Network Based on Asynchronous Time Sharing," IEEE J. Select. Areas Commun., Vol. SAC-8, No. 8, Oct. 1990, pp. 1582-1594.
- [22] W.K. Giloi, P. Behr, G. Zuber, "Upperbus : A High Speed Backbone for Metropolitan Area Networks," Fourth European Fibre Optic and Local Area Networks Conference, EFOC/LAN 86, Amsterdam, June 1986, pp. 286-289.
- [23] M.W. Garrett, J.O. Limb, A. Albanese, "Multiple Gb/s Fiber-Optic Metropolitan Area Network," IEEE International Conference on Communications, ICC 86, Toronto, June 1986, pp. 934-936.
- [24] D. Navaro, "A Metropolitan Area Network Architecture Based on a Multiplexed Slotted and Token Ring Access Protocol," Fifth European Fibre Optic and Local Area Networks Conference, EFOC/LAN 87, Basel, June 1987, pp. 353-358.
- [25] D. Greaves, A. Hopper, "The Cambridge Backbone Network," Sixth European Fibre Optic and Local Area Networks Conference, EFOC/LAN 88, Amsterdam, June 1988, pp. 399-402.
- [26] K. Imai, T. Honda, H. Kasahara, T. Ito, "ATMR: Ring Architecture for Broadband Networks," IEEE Global Telecommunications Conference, GLOBECOM 90, San Diego, CA, Dec. 1990, pp. 1734-1738.
- [27] I. Cidon, Y. Ofek, "Metaring - A Full-Duplex Ring with Fairness and Spatial Reuse," IEEE INFOCOM 90, San Francisco, CA, June 1990, pp. 969-981.
- [28] M.A. Rodrigues, "Erasure Node: Performance Improvements for the IEEE 802.6 MAN," IEEE INFOCOM 90, San Francisco, CA, June 1990, pp. 636-643.
- [29] M. Zafirovic-Vukotic, I.G. Niemegeers, D.S. Valk, "Performance Analysis of Slotted Ring Protocols in HSLANs," IEEE J. Select. Areas Commun., Vol. SAC-6, July 1988, pp. 1011-1024.
- [30] "B-ISDN ATM Functional Characteristics," Draft Recommendation I.150, CCITT Study Group XVIII, Geneva Meeting, 23-25 May 1990.
- [31] D.E. Huber, W. Steinlin, P.J. Wild, "SILK: An Implementation of a Buffer Insertion Ring," IEEE J. Select. Areas Commun., Vol. SAC-1, No. 5, Nov. 1983, pp. 766-774.
- [32] F. Abdul-Ghani, P.A. Davies, "High-Speed Optical-Fibre Ring Network with a Register Insertion Access Protocol," IEE Proceedings, Vol. 131, Part H, No. 2, April 1984, pp. 83-86.
- [33] J.S. Grant, C.R. Grant, "A Buffer-Insertion-Ring Suitable for Time-Critical Applications," Seventh European Fibre Optic and Local Area Networks Conference, EFOC/LAN 89, Amsterdam, June 1989, pp. 71-75.
- [34] J.Y. Chao, W.T. Lee, L.Y. Kung, "A New Buffer Insertion Ring with Time Variant Priority Scheme to Facilitate Real-Time Image Transmission on a High Speed Integrated Local Area Network," Fourteenth Conference on Local Computer Networks, Minneapolis, MN, Oct. 1989, pp. 212-218.
- [35] W. Bux, F.H. Closs, K. Kümmerle, H.J. Keller, H.R. Müller, "Architecture and Design of a Reliable Token-Ring Network," IEEE J. Select. Areas Commun., Vol. SAC-1, No. 5, Nov. 1983, pp. 756-765.
- [36] ANSI/IEEE Standard 802.5-1989, ISO 8802.5, 1989.
- [37] M.J. Johnson, "Reliability Mechanisms of the FDDI High Bandwidth Token Ring Protocol," Computer Networks and ISDN Systems, Vol. 2, No. 2, 1986, pp. 121-131.
- [38] E.A. Zurfluh, H. R. Müller, "Automatic Configuration for Fiber-Based LANs," Sixth European Fibre Optic and Local Area Networks Conference, EFOC/LAN 88, Amsterdam, June 1988, pp. 408-413.
- [39] P. Heinzmann, H. Müller, K. Wilson, "Configuration Control for Bus Networks," Fifteenth Conference on Local Computer Networks, Minneapolis, MN, Oct. 1990, pp. 49-59.
- [40] J. Pihlaja, "Cyclic Ring," RACE document 1011 (BCPN), A8/NOK/62.2, March 11, 1991, pp. 49-59.