Medium Access Control for Synchronous Traffic in the AMNET LAN

David Goodall
Computer and Systems Technology Laboratory (CaST)
School of Computer Science and Engineering
The University of New South Wales
Sydney 2052 Australia

Keith Burston
Manager, Communications Unit
The University of New South Wales
Sydney 2052 Australia

E-mail: castanets@vast.unsw.edu.au

UNSW-CSE-TR-9501 — 23 MARCH 1995

Communicated by Jayasooriah

Abstract

This report presents the medium access control (MAC) scheme designed for supporting synchronous traffic within the AMNET LAN. Synchronous traffic is supported directly at the MAC layer via a table mechanism, and is carried by two different types of cell - synchronous cells, which carry data from a single transmitter, and shared synchronous cells, which carry data from one or more transmitters. The table mechanism provides guaranteed latency and bandwidth to synchronous cell types. The latency involved in getting real-time data from a source device to the LAN is minimised, thus simplifying devices and making internetworking more feasible. This report is intended to describe the different synchronous cell types, the table mechanism, and algorithms for allocation of bandwidth using the table.

1 Introduction

AMNET, (A Multimedia NETwork), is a local area network specifically aimed at bringing synchronous traffic, such as audio and video, to existing bus-based workstations, at low cost [1, 2]. It is a unidirectional slotted ring testbed supporting investigation into various layers of potential multimedia LAN architecture. Destination stripping, or spatial reuse, is used to achieve a relatively high level of utilisation. Slot payloads are 54 byte cells made up of a 6 byte header and 48 bytes of data to maintain compatibility with Asynchronous Transfer Mode (ATM) cells from the adaptation layer upwards. Slots are currently issued at a rate of 256,000 per second, giving a total bit rate of 110.592Mbps. Virtual circuits (VCs) are supported by including a VC identifier in the cell header.

The objective of a medium access control (MAC) scheme for a multimedia LAN is to integrate synchronous and asynchronous traffic, giving pre-emptive priority to synchronous traffic to ensure bounded jitter and delay, while avoiding total starvation of asynchronous traffic. Fairness must be provided for asynchronous traffic sources. This paper presents the MAC scheme designed to support synchronous traffic within the AMNET LAN. The MAC scheme for asynchronous traffic is detailed in [3].

AMNET provides two cell types for carrying synchronous traffic. Support for point-to-point and point-to-multipoint synchronous traffic is provided via synchronous cells, which devote all 48 bytes of data to the one VC. This type of cell may be used by asynchronous traffic if it is empty, thus supporting variable bit rate traffic. Support for multipoint-to-multipoint and multipoint-to-point synchronous traffic is provided via shared synchronous cells, which allow more than one node to place data in a cell or read data from a cell. Bandwidth for synchronous traffic is reserved in advance, using a form of logical time division multiplexing. The master node, referred to as the controller from this point, maintains a table of allocation entries and a pointer to the table. As the pointer moves through the table, the controller issues slots according to the entries. Entries are reserved as part of the set-up phase for synchronous or shared synchronous VCs, and allow bandwidth allocation and regulation per VC. The spaces between entries give shape to the traffic. Therefore a synchronous frame, bandwidth allocation, traffic regulation and traffic shaping are provided in the one mechanism. Synchronous traffic is delivered end-to-end and has pre-emptive priority over asynchronous traffic.

This report is organised as follows. Related work on supporting synchonous traffic in existing network standards is discussed in Section 2. The synchronous and shared synchronous cell types are described in Sections 3 and 4 respectively. Section 5 discusses determination of bandwidth requirements for an example application, and describes a simple algorithm for allocation of table entries to provide this bandwidth. Section 6 contains our conclusions.

2 Background

Motivation for the development of a MAC scheme specifically aimed at multimedia traffic arises from the problems associated with allocating bandwidth to synchronous traffic in existing networking standards. Ethernet is not suitable for applications involving synchronous traffic due to its inability to guarantee delays in accessing the medium.The contention based Ethernet MAC protocol, i.e. CSMA/CD, does not provide a bounded access delay, or support a reservationist model [4]. Research into a modification of CSMA/CD to provide bounded delays for voice related synchronous traffic is reported in [5].

The highest priority of data in FDDI-II [6] is directed towards support of 8KHz sampled telephony. Up to sixteen 6.144Mbps, full duplex channels may be allocated. The corresponding facility in AMNET is the shared synchronous channel, which also provides for ISDN compatibility, but is more flexible. Shared synchronous cells allow channels to be set up which connect a number of nodes. This type of cell requires both transmit and receive phases, i.e. slot rotations, to allow a number of nodes to first place data in a cell and then receive data from all the other nodes. In AMNET a full-duplex channel between just two nodes is more efficiently handled by using two simplex, synchronous, VCs.

Schemes for allocating synchronous bandwidth using FDDI's timed token rotation protocol, e.g. for transmission of video data, are explored in [7, 8]. The normalized proportional allocation scheme proposed in [7] guarantees up to 33% of available bandwidth to synchronous traffic. Reference [8] proposes another allocation scheme for synchronous traffic and shows that in the case of a set of N identical real-time channels, an FDDI network can use at most half of its bandwidth for these channels. An extension of this work in [9] indicates that a modification to the FDDI MAC protocol, called FDDI-M by the author, may significantly improve FDDI's ability to support synchronous traffic in the best case, i.e. allocation of up to 100% of the available bandwidth. The proposed modifications increase the bandwidth guaranteed to synchronous traffic in the worst case to only 50% of the total available.

A discussion of the difficulties involved in allocating bandwidth to high priority variable bit rate traffic in DQDB, and a proposed solution to the problem may be found in [10].

AMNET provides a more flexible scheme than any of the existing standards. By allocating bandwidth from a table, fine control of bandwidth allocation and traffic shape can be exercised. The latter should prove beneficial when the project moves to the internetworking stage. The need to shape traffic entering an ATM network to avoid loss of cells through policing mechanisms is discussed in [11]. In the initial implementation of AMNET, which is based on the design presented in [1, 2], the table mechanism will provide allocation of up to 50% of the ring bandwidth to synchronous traffic. Relaxation of the original constraints to allow up to 100% allocation to synchronous traffic is discussed in Section 3.

3 Synchronous Cells

As stated in the Introduction, the ring controller maintains a table of allocation entries and a pointer to the table. As the pointer moves through the table, the controller issues slots according to the entries. Entries are reserved as part of the set-up phase for synchronous or shared synchronous VCs. This allows bandwidth allocation and regulation per VC. Synchronous cells devote all 48 bytes of data to the one VC.

Table 1 shows an example of synchronous cell allocation when there are three slots in the ring. A synchronous VC originating at node 5 and requiring 24.576Mbps of bandwidth has one slot in every four of the table reserved for its use, until the circuit is shut down. Up to half of the available bandwidth, i.e. one slot in two, may be reserved in this manner. The remaining bandwidth is available to asynchronous traffic, which may also use an empty slot reserved for a synchronous cell providing the slot is freed in time for the sender of the synchronous cell to use it. This supports devices which produce a variable bit rate. The restriction to half the available bandwidth is due to the potential for VCs to overlap in mutually exclusive ways. In addition, VCs may span the controller issuing the cells and require two slot revolutions for pickup and delivery. This may cause starvation of lower priority traffic sources which are delivered end-to-end. A more complete explanation of this phenomenon is given in [3].

The table reservation mechanism allows synchronous VCs, with possibly overlapping topography, alternating use of the ring. A slot which is reserved for a synchronous VC is marked as booked for one revolution, thus allowing any existing cell payload to be delivered, and then marked as owned for the next revolution, which allows the node at the origin of the VC, i.e. the sender or owner, the exclusive use of the slot. The ring controller is responsible for marking the slots according to its table. It places the identity of the node for whom the slot is booked in the owner field of the cell header in the booked phase, and the channel identification into another header field in the owned phase. Thus in both booked and owned phases, all nodes know that the slot is reserved for a VC originating at the owner node. The owner node, i.e. the sender, uses the slot in the owned phase. The channel identification is used by the node to determine which of its multimedia devices is to provide the data to be transmitted.

To implement the mechanism it is only necessary to store the slot type information and the owner/channel information in the table. An address pointer is used to access the table entries. The pointer is incremented after a slot is issued by the controller. There must be an odd number of slots contained in the ring so that the change in phase for each slot is automatic. It can be seen from Table 1 that the phase of each slot is toggled from Booked to Owned, or vice versa, as it is issued by the controller. The number of slots in the ring will be passed to a bandwidth allocation process when the ring is initialised. It will be the responsibility of this process to correctly space the slot type data in the table. For example, in Table 1 the booked phase for a particular slot is three table entries ahead of the owned phase because there are three slots in the ring.

Table 1: Example of table allocation for synchronous cells.

Slot Number	Slot Type	Phase	Owner/Channel
0	Normal Asynchronous	Booked	0x00
1	Synchronous	Owned	0x24 - channel
2	Booked Asynchronous	Booked	0x05 - node
0	Normal Asynchronous	Owned	0x00
1	Normal Asynchronous	Booked	0x00
2	Synchronous	Owned	0x24 - channel
0	Booked Asynchronous	Booked	0x05 - node
1	Normal Asynchronous	Owned	0x00
2	Normal Asynchronous	Booked	0x00
0	Synchronous	Owned	0x24 - channel
1	Booked Asynchronous	Booked	0x05 - node
2	Normal Asynchronous	Owned	0x00
0	Normal Asynchronous	Booked	0x00

A way of removing the requirement for an odd number of slots appears to be possible, and is for future study.

The current implementation in AMNET allows a variable table size of up to 65536 entries. The table is large to allow experimentation. For example, a table size such as 48000 is divisible by three as well as by two. This should allow flexibility in meeting data rate requirements and fine granularity of bandwidth allocation. Two copies of the table are kept. A working copy is used by the finite state machines of the ring controller. A second copy is used by the bandwidth allocation process to allocate new bandwidth and possibly to rearrange the table entries in case of fragmentation. This may occur over time as VCs are setup and torn down. The extent of fragmentation is dependent on the bandwidth allocation algorithm and is an area for future study.

The AMNET scheme ensures that a large number of synchronous VCs with conflicting topographies can co-exist on the ring. In future implementations the mechanism could be relaxed to allow booking of, for example, ten consecutive slots in the table. The allocation would start with a booked phase to allow delivery of existing slot contents and would be followed by ten owned phases guaranteeing bandwidth to the VC sending synchronous data. Lengthy reservations may lead to fragmentation of the allocation table, and starvation of lower priority traffic, and may require the provision of larger buffers at devices.

4 Shared Synchronous Cells

The packetisation delay associated with filling a 48 byte cell from a 64Kbps stream is 6ms. With advances in compression, packetisation delay becomes a problem for audio streams, and may lead to wastage of bandwidth if only part of a cell is used for transmission. In the case of an average conversation, removal of periods of silence from a 64Kbps audio stream can create a 32Kbps audio stream. The associated packetisation delay for a 48 byte cell becomes 12ms,

Table 2: Example of table allocation for shared synchronous cells.

Slot Number	**Slot Type**	**Phase**	**Owner/Channel**
1	Normal Asynchronous	Booked	0x00
2	Normal Asynchronous	Owned	0x00
0	Booked Asynchronous	Booked	0x00 - owner
1	Normal Asynchronous	Owned	0x00
2	Normal Asynchronous	Booked	0x00
0	Shared Synchronous Send	Owned	0x56 - channel
1	Normal Asynchronous	Booked	0x00
2	Normal Asynchronous	Owned	0x00
0	Shared Synchronous Receive	Booked	0x56 - channel
1	Normal Asynchronous	Owned	0x00

which leaves little time for internetworking delays, before such delays become audible. Shared synchronous cells provide an answer to this problem.

Shared synchronous cells work in a similar fashion to synchronous cells with respect to bandwidth allocation. A booked phase is followed by a send phase in which all nodes participating in the VC place data in the cell. This in turn is followed by a receive phase in which all participating nodes read data from the cell. A node is informed during VC setup as to which bytes of the cell it should write and read during the different phases. The owner of a cell booked for shared synchronous usage is always node 0.

As an example of potential use, one shared synchronous cell can carry 48 different 64Kbps channels, i.e. one in each byte of the cell. If two slots in every 16 of the 256,000 issued each second were allocated to a shared synchronous channel, the resulting channel bit rate would be 12.288Mbps. This would provide a bandwidth of 6.144Mbps in both the send and receive phases, and would be suitable as a unit for the construction of a gateway between three European transmission structures, consisting of 32 $\times$ 64Kbps channels each, and four North American transmission structures, consisting of 24 $\times$ 64Kbps channels each. Table 2 shows a part of the table demonstrating the different phases. Any node participating in the VC will have a relevant mapping of the example channel number 0x56, to a particular device, e.g. audio. Data will be transmitted when the node finds that it has such a mapping and the phase bit in the cell header indicates an owned phase. Data is received when there is a mapping and the phase bit indicates a booked phase.

Other granularities of bit rate can also be produced using the table reservation mechanism. Therefore it becomes possible to produce flexible, low latency gateways between different ISDN transmission structures.

5 Bandwidth Reservation Calculations

This section shows an example of bandwidth reservation using the table mechanism. The example is based on the needs of the AMNET Audio Subsystem which uses shared synchronous cells [12, 13].

5.1 Calculation of Table Slot Spacing

Bandwidth supplied to a VC via synchronous cells is reserved by allocating entries in the table according to the following formula.

$$SlotSpacing = (SlotIssueRate \times 8 \times BytesUsed)/BitRate \tag{1}$$

SlotSpacing is the space between allocated entries in the table. For example, a one in four slot allocation results in a spacing of one allocated, i.e. synchronous, slot followed by three slots which are either allocated for other synchronous VCs, or left unallocated.

$$SlotIssueRate = NetworkClockFrequency/NB \tag{2}$$

Where *NB* is the number of bytes in a slot. In the current implementation *SlotIssueRate* is equal to

$$13.824MHz/54Bytes = 256,000Slots/Second \tag{3}$$

BytesUsed is the number of bytes of the cell which are used, i.e. from 1 to 48 bytes. *BitRate* is the bit rate produced by the device serviced by the VC. An Analog-To-Digital Converter without compression will produce data at a constant bit rate. If compression is involved, the bit rate may be variable. Unused cells of a synchronous VC may be used for asynchronous traffic, provided the cell can be emptied in time for the VC for which the slot is reserved, to use it. Bandwidth can be allocated in the table to match an average rate, or a peak rate. Any excess may be used by asynchonous traffic sources. This is not possible for shared synchronous cells, which do not have a recognisable empty state at the MAC level.

For shared synchronous cells, the formula for *SlotSpacing* is still valid but there will be a send shared synchronous cell immediately followed by a receive shared synchronous cell. *SlotSpacing* will represent the number of spaces between consecutive send or receive shared synchronous cells for the VC in question. Table 2 shows an example of this type of allocation. A practical example of allocation now follows.

The AMNET Audio Subsystem is designed to support a conference call involving up to eight stereo audio sources. The audio sampling rate is 32KHz and the sample length 16 bits. The sample rate and length conform to the recommendations for transmission related applications in [14]. There is an additional overhead of eight bits for each sample, for experimentation and for the construction of a personalised sound mix at each node. Therefore each shared synchronous cell contains sixteen 16 bit samples with 8 bits of overhead each, i.e. eight stereo channels. The required *SlotSpacing* is calculated as follows.

From equation 3 *SlotIssueRate* is equal to 256,000, and since we use the entire cell, *BytesUsed* is equal to 48. Therefore

$$BitRate = SR \times (NS \times (SL + POPS)) = 12,288,000Bps \tag{4}$$

Where *SR* is the sample rate, *NS* is the number of samples, *SL* is the sample length, and *POPS* is the protocol overhead per sample. So from equation 1

$$SlotSpacing = (256,000 * 8 * 48)/12,288,000 = 8 Slots \tag{5}$$

This means that one in eight slots of the table are reserved for the send phase and another one in eight slots reserved for the receive phase. If the sample rate was 44.1KHz, the required slot spacing would be :

$$(256,000 * 8 * 48)/16,934,400 = 2560/441 = 5.805 Slots \tag{6}$$

Therefore the required allocation would be 441 slots out of every 2560 at an average spacing of 5.805 for each phase of the shared synchronous VC. If we assume a table size of 65,536 entries, the number of entries in the table for each phase can be determined from the ratio :

$$441/2560 = NumberOfTableEntries/65,536 \tag{7}$$

Therefore to the nearest integer value, the required number of table entries would be 11,290, again at an average spacing of 5.805.

5.2 A Simple Allocation Algorithm

The reserved slots are spaced as evenly as possible within the frame represented by the table size, to reduce jitter. Spacing within the table represents a similar problem to that found in graphics applications where a world coordinate space in real numbers is transformed to a screen coordinate space in integer numbers, for the purpose of viewing the model. A simple spacing algorithm for the above example is as follows :

```
Unit = Slot Spacing = 5.805;
Accumulator = 0.0;
Success = True;

while (Accumulator < 65536.0 and Success)
do
    /* convert to nearest integer value */
    TablePosition = IntegerValue(Accumulator);
    /* find nearest free slot to TablePosition */
    Success = ReserveSlot(TablePosition,MaxBound);
    Accumulator = Accumulator + Unit; /* increment pointer */
od
if Success
    AcceptCall;                        /* inform relevant processes */
    ConfirmAllocation;                 /* perform table update */
else
    RejectCall;
fi
```

The spacing needs to be kept within a maximum jitter bound as set by the buffering capabilities of the device, or the human perception requirement. This is the purpose of the Success variable. If the bound cannot be met, the setup call for the channel must be rejected. The behaviour and performance of table allocation algorithms under dynamic conditions is an area for future study.

6 Conclusions

This report has presented the MAC scheme for supporting synchronous traffic in the AMNET LAN. Synchronous traffic is supported directly at the MAC level by a table mechanism, and may travel in synchronous or shared synchronous cells. The latter provides a fine granularity of bit rate suitable for ISDN applications. Fine control over the allocation of synchronous bandwidth, and bounded jitter and delay, are features of the design. The determination of bandwidth requirements, and a simple algorithm for allocation of appropriately spaced entries in the table, have also been discussed.

The current implementation of the AMNET testbed reflects the initial design presented in [1, 2]. Future work for the AMNET project includes research into software architecture for multimedia communications. This report is meant to provide a basis for a software architect to begin work on table management and various low level call admission aspects of the project.

7 Acknowledgements

We thank David Johnson for his many contributions to the construction of the AMNET testbed.

References

[1] A.K. Burston, *An Architecture for a Broadband LAN*, Second International Conference on Private Switching Systems and Networks, IEE Conference Publication No. 357, London, 1992, pp. 169-176.

[2] A.K. Burston, *AMNET - A Low Cost Broadband LAN*, Australian Broadband Switching and Services Symposium, 1, Melbourne, July 1992, pp. 137-144.

[3] D.S. Goodall and A.K. Burston, *Medium Access Control for Asynchronous Traffic in the AMNET LAN*, International Journal of Communication Systems, To Appear.

[4] D. Anderson, *Metascheduling for Continuous Media*, ACM Transactions on Computer Systems, 11, (3), August 1993, pp 226-252.

[5] N.F. Maxemchuk, *A Variation on CSMA/CD That Yields Movable TDM Slots in Integrated Voice/Data Local Networks*, The Bell System Technical Journal, 61, (7), September 1982, pp 1527-1550.

[6] F.E. Ross, *FDDI - a Tutorial*, IEEE Communications, 24, (5), May 1986, pp 10-17.

[7] G. Agrawal, B. Chen, W. Zhao and S. Davari, *Guaranteeing Synchronous Message Deadlines with the Timed Token Protocol*, 1992 IEEE 12th International Conference on Distributed Computing Systems, pp 468-475.

[8] Qin Zheng and Kang G. Shin, *Synchronous Bandwidth Allocation in FDDI Networks*, Proceedings First ACM International Conference on Multimedia, 1993, pp 31-38.

[9] Qin Zheng, *Real Time Fault Tolerant Communication in Computer Networks*, PHD Dissertation, University of Michigan, 1993.

[10] C. Bisdikian and A.N. Tantawy, *A Mechanism for Implementing Preemptive Priorities in DQDB Subnetworks*, IEEE Trans. on Comms., 42, (2/3/4), Feb./Mar. 1994, pp 834-839.

[11] J.L. Adams, *Orwell*, Computer Networks & ISDN Systems 26 (1994), pp 771-784.

[12] D. Goodall and K. Burston, *Medium Access Control for Multi-Media Communications*, Australian Telecommunications & Applications Conference (ATNAC) 1994, pp 437-442.

[13] C.W.S. Chow, *AMNET Audio Subsystem*, Undergraduate Honours Thesis, 1994, School of Computer Science and Engineering, University of New South Wales, P.O. Box 1 Sydney 2052, Australia.

[14] AES 5-1984, *AES Recommended Practice For Professional Digital Audio Applications Employing Pulse-Code Modulation – Preferred Sampling Frequencies*, J. Audio Eng. Soc., 32, (10), October 1984.