Technical Report

UCAM-CL-TR-121
ISSN 1476-2986

Number 121

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

A high-level petri net specification of the Cambridge Fast Ring M-access service

Jonathan Billington

December 1987

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

https://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

https://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

A High-Level Petri Net Specification of the Cambridge Fast Ring M-Access Service

Jonathan Billington*
University of Cambridge
Computer Laboratory
New Museums Site
Pembroke Street
Cambridge CB2 3QG

December 1987

Abstract

Numerical Petri Nets (a high level inhibitor net) are used to characterise the Cambridge Fast Ring Hardware at a high level of abstraction. The NPN model describes the service provided to *users* of the hardware (stations, monitors, bridges and ring transmission plant), known as the M-Access Service. The model has been developed to formalise the M-Access service definition in order to remove ambiguities and as a basis for the development and verification of the protocols using the M-Access service.

Keywords

Formal specification, high-level Petri nets, local area networks, service specification, protocols.

*The author is supported by a Telecom Australia Postgraduate Scholarship and an ORS Award.

1 Introduction

The Cambridge Fast Ring (CFR) Networking System [1] consists of a cluster of CFRs interconnected by bridges. The CFR is a slotted ring designed during the early 1980s to provide a raw 100 MBs transmission speed and to substantially increase the bandwidth between point-to-point users. Hardware for the stations, the monitor and bridges for the Cambridge Fast Ring has recently been fabricated in VLSI. The hardware implements the low level protocols between the various distributed components. The task of designing a set of protocols above the basic hardware to provide application services is underway.

An initial draft of the protocol architectures for the CFR has been compiled in [2], where it is shown that different architectures can co-exist above the basic service provided by the CFR hardware. This service is known as the M-Access Service and has been defined in [3].

The following benefits would accrue from providing an appropriate formal description of the service.

- Ambiguities which arise in narrative descriptions would be removed.
- The formal model could be executed to investigate properties of the service, such as global sequences of service primitives.
- The model can provide the basis for the development or synthesis of protocols to be implemented above the M-Access Service, such as the Unison Data Link Protocol [4].
- The model provides the basis for the verification of such protocols once specified.

The purpose of this paper is to describe one such model, using Numerical Petri Nets (NPNs) [5] as the formal description technique. NPNs have been chosen because of their visual appeal and the availability of computer aided tools to analyse them [6]. The appendix gives a brief description of NPNs.

The paper is structured as follows. Section 2 describes the M-Access Service based on [3] and [1] and discusses some of the assumptions made in deriving the NPN model. Section 3 presents the NPN specifications and various specification issues are discussed in section 4. The final section provides some conclusions.

2 CFR M-Access Service

2.1 Features of M-Access

A draft description of the M-Access Service is given in [3]. The service provides an abstract view of the features of a ring cluster: a set of rings interconnected by gateways. From a *user's* perspective the ring cluster provides the following facilities:

- Error protected communications paths;
- Fixed slots of 32 octets in which to transmit packets;

- A routing mechanism by which slots can be routed to their destinations, given that a 16-Bit address is provided by the user.
- Two types of communication paths:
  1. Point-to-point, where the address indicates the particular destination; and
  2. Broadcast, where a special address (hex FFFF) indicates that the message is to be broadcast to all other stations on the ring cluster.
- Some buffering
- The communications path has the following characteristics
  1. Packets may be lost
  2. Packets may be duplicated (this is a rare event, but possible)
  3. Packets may not be sequenced (this can only happen in an interconnected ring cluster where there is the possibility of two (or more) paths from source to destination).

2.2 Service Primitives

In the style of Open Systems Interconnection, service primitives define the communication between the users and the provider of a communications service in terms of

- what is to be transferred across the interface - this is defined by a set of parameters associated with the service primitive and the service primitive type.
- the allowable sequences of service primitives both at a local interface and globally.
- the relationships between the service primitives at each of the interfaces.

The service specification is provided at an abstract level in order to avoid over specification of the interface between the user and provider.

In the M-Access Service, two service primitives are defined:

- M-DATA request
- M-DATA indication

2.2.1 M-DATA request

This primitive is invoked to initiate the sending of data from one service user to another service user (for point-to-point operation) or to all other service users (broadcast). The primitive therefore has 3 essential information types: the source address, the destination address and the data to be transferred, which are defined as associated parameters, using the following syntax:

$$M-DATA\ request(source-CFR-address, destination-CFR-address, M-data)$$

In [3], two further parameters are defined: *retry-control* and *transmit-status.* We shall model the retry-control parameter at a higher level of abstraction (perhaps more appropriate for a service specification as retry-control does not involve both users (or all users in broadcast mode)), where we will allow the service provider to choose any number of retries non-deterministically. Hence any number of retries (including zero) would be allowable in a realisation of the service, and user control of this number on a packet basis would also be a possibility.

We argue here that the *transmit status* parameter should be removed from the M-DATA request primitive, as the return of its value cannot be considered atomic with the transfer of data from user to provider. This is important from the point of view of defining sequences of service primitives. If a value of *transmit-status* needs to be determined before the M-DATA request can occur, then the corresponding M-DATA indication may have occurred before it! The present time-sequence diagrams quite rightly deny this possibility - so we have a problem. This may be solved by creating a separate TRANSMIT-STATUS indication primitive. This primitive will only occur at a local interface (no global significance) and is normally excluded from service definitions, however, it appears useful to include it in a simplified form as discussed below.

2.2.2 M-DATA indication

This primitive is invoked to receive data sent by a sending user. It complements the M-DATA request primitive. The receiving user has completed the receipt of all the data when the primitive occurs.

The primitive has the same set of parameters as the M-DATA request primitive.

$$M-DATA\ indication(source-CFR-address, destination-CFR-address, M-data)$$

These parameters have the same values as those in the corresponding M-DATA request primitive. (Note: we are not considering address mapping required in multi-cluster configurations in this report. They have also not been considered in [3].)

2.2.3 M-TOG indication

The CFR hardware has the capability of telling a user of the M-Access Service that a transmission of a packet has not succeeded (ie the number of retries has been exhausted). This signal is known as 'Thrown-on-Ground' or TOG for short. A TOG will normally indicate that the receiver is busy. The effect of the TOG is to discard the packet. The user then has the option of retrying the same packet, accepting the loss, or trying another packet to a different destination before retrying sometime later. Hence the TOG signal has important consequences for the way in which the user behaves. It appears to be useful at the service level to explicitly define a primitive to express this characteristic of the service, as a form of notification service. A possible parameterless primitive would be: M-TOG indication, indicating that the current packet is considered lost by the service provider.

2.3 Sequences of Service Primitives

2.3.1 Point-to-Point

In [3], the sequences are partially determined by the time-sequence diagrams of figures 3a) and 3c). The M-DATA indication either follows the corresponding M-DATA request, or it does not occur at all. What is not covered in [3] is a statement as to how much buffering will be allowed - ie how many M-DATA requests can occur, before a M-DATA indication must occur in the case where there is no loss? This is obviously implementation-dependent and an implementation-independent specification must allow for the choice to be arbitrary.

Other important points are that duplicates are possible (although rare) and that the medium does not preserve sequence (but only in CFR clusters where there are multiple paths between source and destination).

Now that we have introduced the M-TOG indication, its affect on the allowable sequences of primitives will need to be defined. This will be done in the NPN specification.

2.3.2 Broadcast

The sequences of primitives for a successful broadcast are partially given in figures 3b) and 3c) of [3]. There are a number of questions that need to be discussed here. The set of M-DATA indications that may arise from a broadcast must occur (if at all) after the originating M-DATA request. However, nothing is stated regarding the sequences of occurrences of the resulting M-DATA indications. Obviously this depends on the ring topology, the delays through the receiver and whether or not retries are allowed when broadcasting. Retries (and hence duplicates) can occur if the source station does not receive its own broadcast because it is busy receiving another packet for example. (Aside: If it is accepted that duplicates can occur in the CFR, then rebroadcasting on failure may be an acceptable strategy.) If no retries are allowed, then the medium cannot duplicate packets. Out of sequence messages are still a possibility. A reasonable abstraction may be to assume that the occurrences of M-DATA indications are not ordered across the different receiving stations. Although it is possible that M-DATA indications will occur in the order that stations are encountered on the ring, this cannot be guaranteed, due to packets being flow controlled across the M-Access interface.

Figure 3c) suggests that if any packet is lost, all are lost. It is possible however, that a number of stations could receive a packet, while others do not. A M-DATA indication need not occur due to the receiver being busy or because of a transmission error (or other reasons). It appears reasonable to abstract from the ring topology and assume that loss by a particular station is independent of its position on the ring, even though loss due to a corrupted packet would imply that all further destinations on the ring concerned would discard the packet (except in the unlikely event of a packet's CRC being made good due to noise). Because packets can be lost due to the receiver being busy, this position-independence assumption for loss is required as the most general case.

The use of M-TOG indication in broadcast mode is rather problematic. The broadcast protocol appears to work in the following way. A broadcast is initiated by a source by setting the destination address to all ones. If other stations on the CFR are willing to accept packets from the source of the broadcast, they will do so if they are not busy. On

detecting that it is a broadcast packet, the CRC is not changed. If the CRC is bad, the packet is not received, but continues on its way around the ring. If the CRC is good the packet is received and sent on its way, again with the same CRC (good). If the sender receives back the broadcast packet with a good CRC, it notes that it is a broadcast packet and no retransmissions are initiated. Hence no TOG will occur. Of course, the broadcast may well have failed to be received by many stations that were busy at the time. That is hard luck, they only get one chance in this scenario and no M-TOG indication will occur.

It is possible, however, on a multislot ring, that the sender of the broadcast packet is busy receiving another packet when its broadcast packet returns. In this case, no signal is sent from the receiver to indicate that it was a broadcast packet from itself, and it is treated like a normal packet. The good CRC will indicate that the packet has not been received and should be retransmitted. This could cause a string of duplicate broadcast messages to be received until the retry limit is exceeded. At this stage a TOG signal occurs. The user will find it difficult to use the TOG in any sensible way - as its interpretation can be that all or none of the destinations (or anything in between) have received the packet and any number of duplicates. It appears to be advisable for the TOG signal to be ignored when broadcast mode is used. Hence, at the service level, the M-TOG indication will not occur.

2.4 List of Assumptions

This section summarises the assumptions that have been discussed above.

2.4.1 Point-to-point

1. A M-DATA request must have preceded the occurrence of a corresponding M-DATA indication.
2. The parameters associated with the M-DATA indication are identical to those of the corresponding M-DATA request.
3. Duplication is possible, but only if retries are allowed.
4. In ring clusters, sequencing is not maintained in general.
5. Single packet buffering occurs in the transmitter and receiver in the current CFR implementation. There will also be buffering in any gateways.
6. Loss of M-SDUs is possible and handled in two ways:
   - reported to a user in an M-TOG indication if a retry limit is exceeded; but
   - otherwise not reported to the user.

2.4.2 Broadcast

1. A M-DATA request must have preceded the occurrence of a corresponding M-DATA indication.

2. The parameters associated with the broadcast set of M-DATA indications are identical to those of the originating M-DATA request.
3. Loss of M-SDUs is possible. In general it does not depend on the position of the station on the ring. It is not reported to the user.
4. Duplication is probable on multislotted rings, if retries are allowed.
5. Occurrences of M-DATA indications are not ordered across destination stations.
6. In general misordering is possible.
7. Same buffering as for point-to-point.
8. M-TOG indications do not occur.

3 NPN Specification

In this section we will specify various characteristics of the M-Access Service using Numerical Petri Nets. The following aspects will be investigated for both Point-to-Point and Broadcast operation.

1. An arbitrary cluster of rings, with unlimited storage.
2. A single CFR with a single packet buffer for sending and a single packet buffer for receiving in each of its stations, corresponding to the CFR implementation.

3.1 General Comments

The aim of this section is to start with the most general M-Access Service that may be envisaged and then to refine it towards the actual Cambridge Fast Ring Architecture. It will be assumed that there can be an arbitrary number of stations communicating over the service, where the number is limited by the Source Address space.

3.2 Abstracting from CFR slots

At the top level of abstraction we shall assume that any station can access the CFR simultaneously with any other station. This implies no contention over slots. In a more detailed specification the slot contention could be modelled.

3.3 Abstracting from Ring Topology

We shall also abstract from the ring topology by assuming that the service is independent of the positions of the stations around the ring. For ring clusters we shall also assume that the service is independent of the ring to which a station is attached. For example, a station on ring 5 communicating with another station on ring 1 will be treated as identical to stations communicating on the same ring. This is the general case for the CFR and CFR clusters when considering possible global sequences of service primitives as there can be arbitrary delays caused by a receiving station being busy.

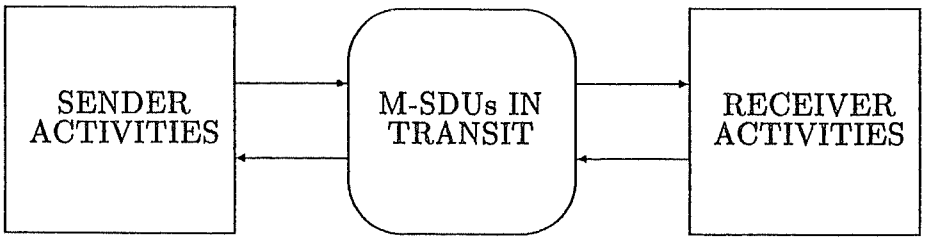


Figure 1: Means/Activity Net of CFR M-Access Service

3.4 Structure

It is assumed that the sending and receiving operations in each of the stations are identical and independent. We therefore only need to model a generic sender communicating over the ring with a generic receiver, each parameterised by the station address.

The general structure is given by Means/Activity nets. In Means/Activity nets, *activities* (actions) are represented by rectangles and *means* (resources) by ovals (or rectangles with rounded corners). An arrow from a *means* to an *activity* implies that the *means* is necessary for the *activity* to occur and arrow from an *activity* to a *means* implies that the *means* is modified by the *activity*, often by the production or consumption of a resource associated with the *means*.

The structure of the CFR M-Access Service is given in figure 1. "Sender Activities" and "Receiver Activities" involve the invocation of service primitives associated with sending and receiving data, respectively. The M-SDUs are the M-Access Service Data Units that are in transit from sender to receiver. It is assumed that M-SDUs are available for the sender and that they are forwarded on to the destination user.

3.5 NPN Specification: CFR Cluster

3.5.1 Implicit Interaction with Users

The Means/Activity Net of figure 1 may be refined into a general M-Access Service where the dynamics are specified by the Numerical Petri Net of figure 2. Implicit interaction with the M-Access Service Users is modelled by the occurrence of service primitives. Explicit interaction with M-Access Service Users is considered in the next section. Service primitives associated with sending are drawn on the left side of the diagram, those associated with describing the channel between the sender and receiver are drawn in the centre and those associated with receiving on the right.

The NPN comprises 4 transitions and one place. Three of the transitions model the three service primitives, while the forth models possible loss of M-SDUs that is not reported to a user. The place 'SP-storage' (Service Provider storage) models an unbounded number of buffers in the service provider and may be regarded as a queue with a non-deterministic service discipline. The place may store structured tokens which are triples. These tokens

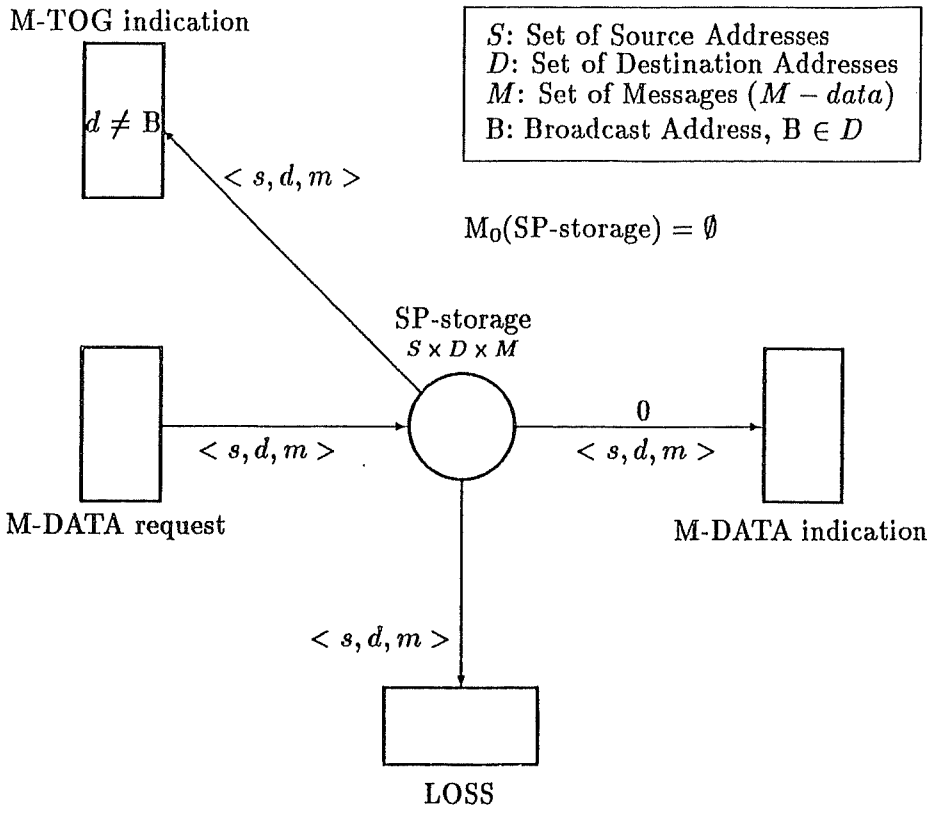


Figure 2: Top Level NPN of CFR M-Access Service

represent the M-SDUs that are in transit from source to destination. The first variable of the triple represents the source address; the second, the destination address; and the third the data to be transferred.

The following properties of the service provider are modelled..

- Service Primitive Occurrences. The occurrences of the service primitives M-DATA request, M-DATA indication and M-TOG indication are modelled by the firing of the transitions labelled with the corresponding names. The station at which the service primitive occurs is determined by the address variables in the associated M-SDUs. The M-DATA request and M-TOG indication primitives occur at the *source* address of the associated M-SDU and similarly the M-DATA indication primitive occurs at the *destination* address of its associated M-SDU. The occurrence of the M-TOG indication indicates to the source user that the provider has discarded the M-SDU and believes that it has not been delivered to its destination.
- M-SDUs. SDUs are modelled as tokens placed in 'SP-storage'. The tokens are triples containing 3 variables: the source address, s whose domain is the source address space (integers 1 through 65534 for the CFR parameter Source-CFR-address); the destination address, d, which corresponds to the source address plus a special broadcast address, B, (the broadcast address is 65535 for the CFR) and the monitor

address, M, (0 for the CFR) and; the data field, m, corresponding to the M-data parameter of the CFR, which is an arbitrary string of length 32 octets.

- Arbitrary Buffering within the service provider. In order to allow any amount of storage in the M-Access Service Provider, it is necessary to allow the place 'SP-storage' to be unbounded or to have a finite but indeterminate capacity. We have modelled the unbounded case here as it is (slightly) easier to represent. This allows a particular implementation to have any countable number of buffers and still conform to the Service Specification.

- Sequence: The place 'SP-storage', acting as a non-deterministic queue, allows arbitrary overtaking of an M-SDU by another M-SDU and hence models the "non-sequence preserving" nature of the service. Note that FIFO order is also a possibility.

- Arbitrary Loss: Loss not reported to the service user is modelled by the occurrence of the "LOSS" transition.

- Normal Transfer: M-SDUs are placed in 'SP-storage' on the occurrence of an M-DATA request. The choice of values for the parameters is chosen arbitrarily from the domains of the variables. Any number of M-DATA requests may occur, initiated by any station and destined for any station. So long as there is an M-SDU in 'SP-storage', an M-DATA indication may occur. The M-SDU is retained to allow for possible duplication (see below) or for broadcast. Normal transfer is modelled by the occurrence of the M-DATA indication transition followed by the occurrence of the LOSS transition for the same SDU, without any intervening occurrence of another M-DATA indication for the same SDU. The possibility exists for a destination not to receive M-SDUs from a source on the *hate list.* This corresponds to sequences in which M-DATA requests for a particular source-destination pair are always followed by either a LOSS event or a M-TOG indication, but not by a M-DATA indication.

- Duplication: Duplication is modelled by the occurrence of the M-DATA indication transition 2 or more times for the same M-SDU.

- Broadcast: This is indistinguishable from duplication except that the 'destination' address parameter must be the Broadcast address value, B. The individual destination address for each occurrence of the M-DATA indication for the broadcast is not known at this level of abstraction. The next section details how this information may be incorporated by a refinement of the specification in figure 2. Duplication of broadcast M-SDUs is allowed. If duplication is not intended (as is the case with the CFR) it may be removed in a further refinement as shown in a further section.

- M-TOG indication. The occurrence of this transition for a particular M-SDU prevents any further occurrences of the M-DATA indication for the same SDU (by removing it from the queue), and indicates to the source that the service provider believes (rightly or wrongly) that the M-SDU has been discarded. The main reason for this in the CFR is that the receiving station is busy and has refused to accept the M-SDU (on a number of occasions determined by the retry limit). We have deliberately forbidden the occurrence of an M-TOG indication for a broadcast M-SDU, by associating the condition $d \neq$ B with the corresponding transition. Of course, this

restriction could be removed if it was felt useful for an M-TOG indication to be used for broadcast M-SDUs.

- Retry control. In this specification retry control is handled implicitly and non-deterministically. The retries are the mechanism for duplication. Duplication can only occur if retries are allowed, but it may not occur even if retries are allowed. As far as the occurrence of service primitives is concerned, the number of retries is not relevant, and is invisible to the users. One important factor to users is the number of duplicates and that they can be limited to zero by retry prevention. The present specification models arbitrary duplication. This is more general than the Cambridge Fast Ring, where the number of duplicates is bounded by the retry limit. A more detailed specification can be given to accommodate this limit by explicitly modelling a retry control parameter which passes the limit to the M-Access Service provider. At this stage the transmit side of the service specification is being refined to an interface specification.

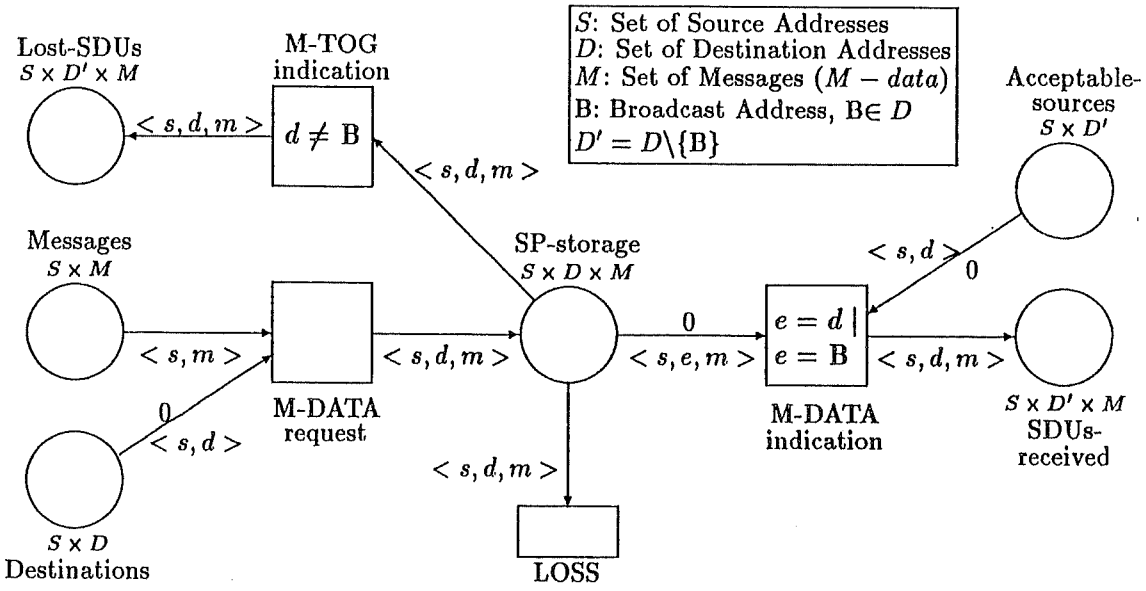
- Quality of Service. Quality of service parameters have not been included in [3] and have thus been ignored in the current NPN specification. The Retry Control parameter may be considered as an implementation-dependent QOS parameter, as it affects a) the transfer delay, b) the probability of SDU loss and c) the probability of duplication. In a service specification it is important to abstract away from implementation choices. This is why the present service specification does not include the retry control parameter. It is considered inappropriate at the service level of specification.

- Ring Broken. It appears to be useful to include in the definition of the service a *Ring-Broken* primitive. This has not been modelled as again it does not form part of the M-Access Service definition in [3]. Given that the effect of a broken ring is to lose M-SDUs and possibly to allow for duplicates, the present specification does model this behaviour without the introduction of a specific primitive.

3.5.2 Explicit Interaction with Users

The specification of figure 3 shows how the service interacts with its users and also specifically indicates the destinations which receive M-SDUs as a result of a broadcast. Five places (and associated arcs) have been added, 3 for the source user and two for the destination user. It is quite arbitrary whether or not any, all or none of the destinations receive a broadcast M-SDU.

Each station's source has a set of messages stored in place 'Messages' which it wishes to transmit to any one of a set of destinations. (In the CFR, each message is restricted to an arbitrary string of 32 octets.) The source may also wish to broadcast the message. The Broadcast and destination addresses are stored in place 'Destinations'. When a M-DATA request occurs, an M-SDU is formed for the particular source-destination pair and the associated message and stored in the M-Access service provider. This M-SDU may now be lost (transition 'LOSS' occurs); discarded by the service provider while informing the source ('M-TOG indication' occurs); or it may be delivered to an allowed destination ('M-DATA indication' occurs). The M-TOG indication may not occur for a broadcast M-SDU. When it does occur, the discarded M-SDU is saved in the place 'Lost-SDUs'. Any number of M-DATA requests may occur concurrently from any number of stations.



Initial Marking

M_0(SP-storage)=M_0(SDUs-received)=M_0(Lost-SDUs)=$\emptyset$
M_0(Messages)$\in [S \times M \longrightarrow N]$
M_0(Destinations)$\subseteq S \times D$
M_0(Acceptable-sources)$\subseteq S \times D'$

Figure 3: NPN of CFR M-Access Service: Explicit interaction with users

Each destination is prepared to receive messages from a set of sources (cf the 'hate list' and select register of the CFR). These are stored in the place 'Acceptable-sources'. If the source address of an M-SDU in 'SP-storage' is on the list of acceptable sources, the M-DATA indication may occur and the M-SDU is passed to the destination and stored in 'SDUs-received'. Duplication is allowed by the M-SDU remaining in 'SP-storage'. If it is a broadcast M-SDU, then the source address must still be acceptable to the destinations that receive it. Two points should be made regarding broadcast.

1. A destination station which receives the broadcast M-SDU is now identified.
2. Arbitrary duplication of broadcast M-SDUs is allowed to each destination that finds the source acceptable. If the specification is to be restricted to disallowing duplicates when broadcasting then a more complicated specification results. The details are presented in the next section.

3.5.3 No Duplication when Broadcasting

In this section a specification of the CFR M-Access service is developed where no duplication occurs in broadcast mode. We will not include interaction with the users explicitly, as this can be done in exactly the same manner as in the previous section.

We assume that the broadcast to each receiving station is not ordered and that any number

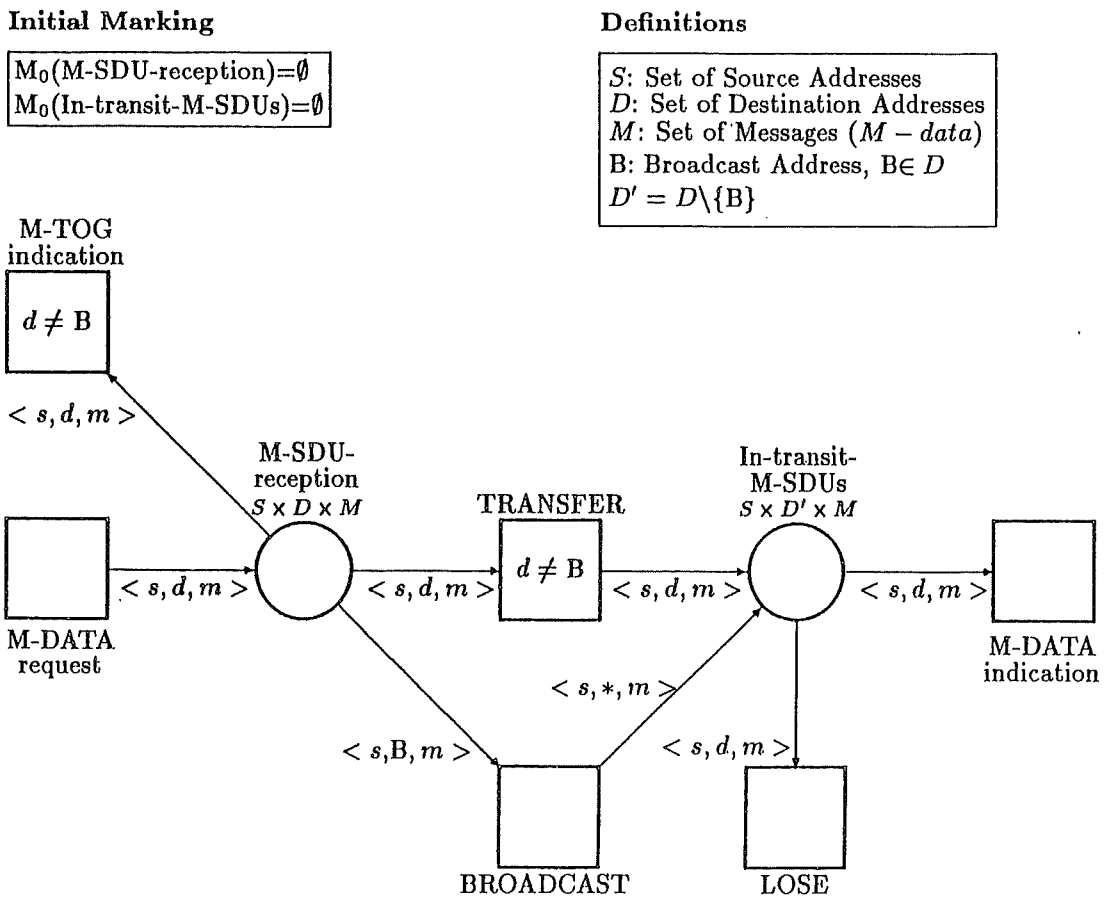


Figure 4: M-Access Service: No Duplication

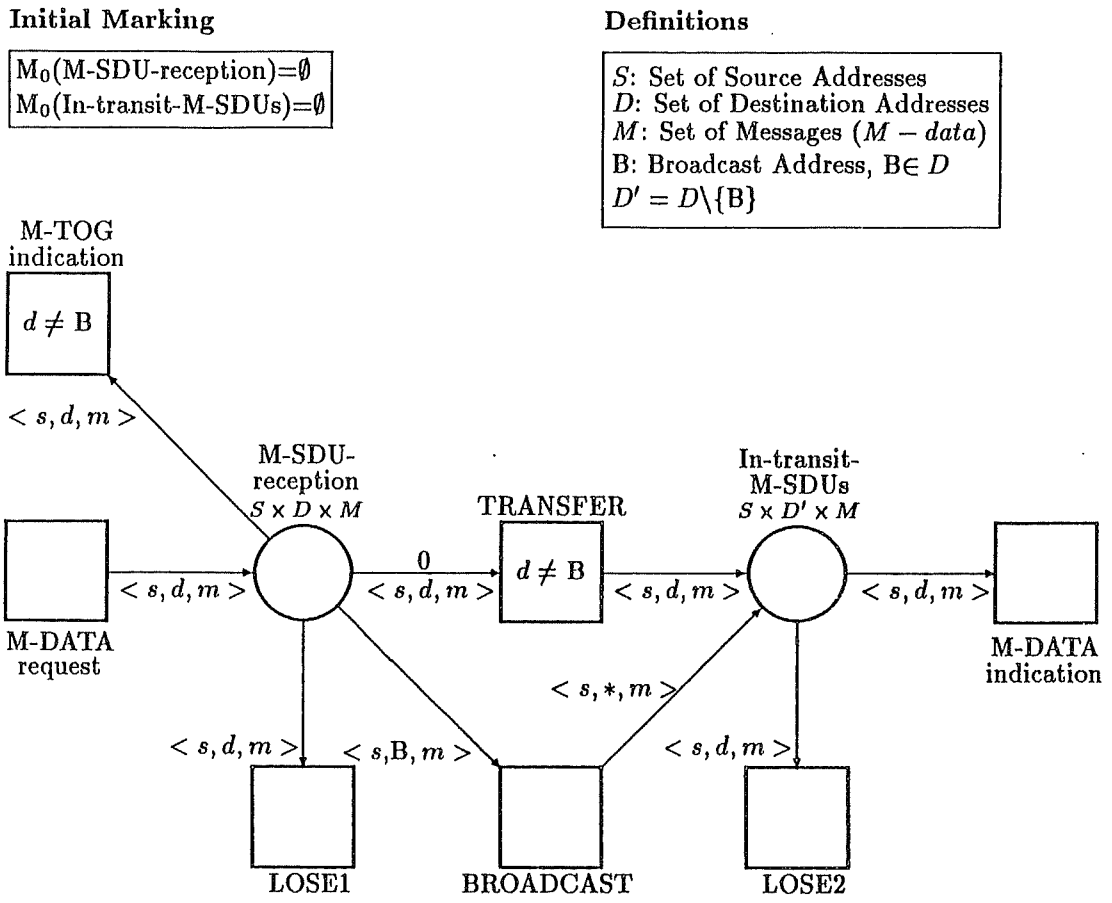


Figure 5: M-Access Service: No Duplication for Broadcast

of stations may not receive the broadcast. We firstly consider the simplest situation where there is no duplication for point-to-point. (This may be regarded as close to the initial expectation of the service to be provided by the CFR.)

The NPN specification is given in figure 4. It has been necessary to refine the service provider storage into two places: 'M-SDU-reception' which stores M-SDUs of the initial M-DATA request; and 'In-transit-M-SDUs' which stores all possible broadcast and point-to-point M-SDUs. (This has been done to ensure that M-TOG indications may only occur for point-to-point M-SDUs.) Point-to-point M-SDUs are simply transferred from 'M-SDU-reception' to 'In-transit-M-SDUs' by transition 'TRANSFER'. The transition 'BROADCAST' converts a broadcast M-SDU into a set of M-SDUs, one for each possible destination. Any M-SDU may be lost (transition 'LOSE') or successfully delivered to its destination ('M-DATA indication').

In order to allow for duplication of point-to-point M-SDUs, we retain a copy of the M-SDU in 'M-SDU-reception' and allow any number of duplicates by successive firing of 'TRANSFER'. An extra transition is included to allow the M-SDU to be removed from the provider. The situation is depicted in figure 5.

3.5.4 CFR M-TOG indications

The above specifications allow a M-TOG indication to occur any time after a M-DATA request has occurred, so long as the M-SDU remains in 'SP-storage' (figure 2) or 'M-SDU-reception' (figures 4 and 5). This allows any number of M-DATA requests to have occurred at a particular station, before an M-TOG indication occurs which relates to any one of the previous M-DATA requests.

This is more general than the situation which exists in the CFR implementation, where only single buffering is provided in each station for the transmission of M-SDUs. Thus after a M-DATA request, a M-TOG indication must occur before the next M-DATA request, if it occurs at all. In other words, for a particular CFR station, the M-TOG indication relates to the M-DATA request which immediately preceded it. Thus a strict order is imposed.

This may be specified in the NPNs of figures 4 and 5 by introducing a place (and associated arcs) which restricts the capacity of 'M-SDU-reception' to one M-SDU per source station. This construction is illustrated in the next section and is not repeated here.

In the original design of the CFR, the idea of double buffering (ie allowing two M-SDUs to be stored in the transmit and receive FIFO buffers) was considered. This would allow two M-DATA requests to have occurred before the M-TOG indication occurred for the first M-DATA request. This can also be modelled very simply by a change in the initial marking of the control place which determines the capacity of 'M-SDU-reception'. The initial marking represents the number of buffers available for each station's transmit FIFO. We can therefore allow a mixture of single and double buffering in different stations (or in general a mix of any number of buffers) by altering the initial marking of this control place.

3.6 NPN Specification: Single CFR

Each station in the CFR has two buffers: one for sending M-SDUs and the other for receiving M-SDUs. Each buffer has the capacity for just one M-SDU.

The more general case of unlimited storage was specified in the previous section. We may now refine the NPN of figure 2 to the specific case of single buffering for the CFR. In this section we shall only consider the case of implicit interaction with the users. Explicit interaction can be added trivially, in a similar way to that shown in figure 3.

We shall consider the following characteristics of the CFR

- Arbitrary number of stations
- Point-to-point and broadcast modes
- Single transmit buffer and single receive buffer for each station
- Sequence of M-SDUs preserved per source-destination flow
- Single broadcast by each station (only one broadcast per station is allowed at any one time due to the single transmit buffer)
- Arbitrary loss of M-SDUs

- Three modes of duplication:
  1. Arbitrary duplication in both point-to-point and broadcast mode;
  2. No duplication in broadcast mode, but arbitrary duplication for point-to-point operation; and
  3. No duplication

The duplication case 2 is close to the operation of the CFR, although duplication for point-to-point is rare and limited. A limit to the amount of duplication can be incorporated into the specification in a straightforward way if desired. (It requires an extra place to store the duplication limit for each station.)

We shall consider the three modes of duplication in separate NPN specifications. As usual, the left side of each diagram represents the transmitter and the right side the receiver. The transitions in the centre represent various ways in which the CFR can operate. We represent a set of transmit buffers, one for each station, by the single place 'Transmit-buffers' and we record the stations which have buffers that are empty in place 'Empty-transmit-buffers'. (This is the same as the control place for determining the capacity of M-SDU-reception mentioned above.) A similar situation exists for the receive buffers. We also include explicitly which stations are acceptable sources of M-SDUs for each of the destinations, by storing them in place 'Acceptable-sources' as we did in figure 3.

3.6.1 Arbitrary Duplication

The single CFR M-Access service with arbitrary duplication in both broadcast and point-to-point modes is specified by the NPN in figure 6. The initial state of the service is specified by the initial marking of the net. Each station connected to the CFR will have an empty buffer for transmitting and one for receiving and these are stored as tokens with a single variable that takes the value of the station address, in the places 'Empty-transmit-buffers' and 'Empty-receive-buffers' respectively. The addresses of the source stations acceptable to each destination are stored in place 'Acceptable-sources' as tokens which are pairs. The first variable stores the source address and the second the destination address. Initially all the transmit and receive buffers are empty (places 'Transmit-buffers' and 'Receive-buffers' are empty).

With this initial state, any number of stations may request the sending of an M-SDU. This is achieved by firing transition 'M-DATA request'. A token representing an M-SDU, (a triple consisting of the source address, destination address and the message contents) is placed in 'Transmit-buffers' and the token representing that the buffer was empty for that station is removed from 'Empty-transmit-buffers'. If the M-SDU is not broadcast, then one of three events may occur:

1. The M-SDU is successfully transferred to the chosen destination. This may only occur if the source is acceptable to the destination. This is achieved in the NPN by firing transition 'TRANSFER'. A copy of the M-SDU is maintained in the transmit buffer while it is transferred to the destination's receive buffer which is removed from the list of empty buffers. The M-SDU may then be removed from the transmit-buffer which would then be marked free by the occurrence of transition 'LOSE'.

Definitions

S: Set of Source Addresses
D: Set of Destination Addresses
M: Set of Messages ($M - data$)
B: Broadcast Address, B$\in D$
M: Monitor Address, M$\in D$
$D' = D \backslash \{B\}$

Initial Marking

M_0(Transmit-buffers)=M_0(Receive-buffers)=$\emptyset$
M_0(Empty-transmit-buffers)$\subseteq S$
M_0(Empty-receive-buffers)=M_0(Empty-transmit-buffers)$\cup\{M\}$
M_0(Acceptable-sources)$\subseteq S \times D'$

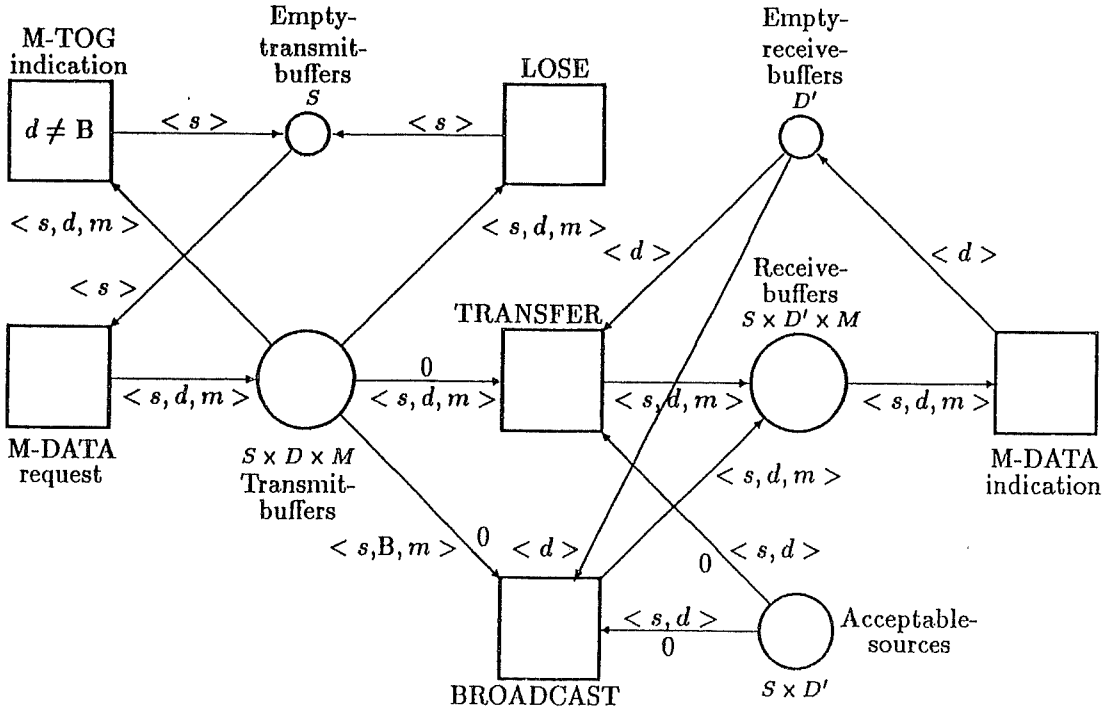


Figure 6: Single CFR M-Access Service: Duplication

Concurrently, an M-DATA indication may occur at the destination, with the M-SDU being removed from the receive-buffer which is marked free. This may be considered as the normal operation of the service. Duplication may occur by firing 'TRANSFER' twice (or more) before the occurrence of the 'LOSE' transition.

2. The M-SDU is refused by the destination and this is reported to the source user. This is achieved by firing 'M-TOG indication', which removes the M-SDU from the transmit buffer and marks it free.

3. The M-SDU is lost. The CFR transmitter hardware falsely believes that the M-SDU has been accepted by the destination, due to a CRC error in the return path. This is represented in the NPN by the firing of the 'LOSE' transition. The M-SDU is discarded and the transmit buffer marked free.

For broadcast M-SDUs, there are two possibilities.

1. The M-SDU is lost by firing transition 'LOSE'.

2. The M-SDU is broadcast one at a time to any of the allowable destinations by repetitively firing transition 'BROADCAST'. When this transition occurs, a copy of the M-SDU is retained in the transmit buffer, the M-SDU is transferred to an accepting destination and its buffer is removed from the empty list. An M-DATA indication may then occur with the consequent release of the receive buffer. This then allows duplication of the broadcast M-SDU, as the 'BROADCAST' may occur again for the same destination. It may also occur again for any other destination. The broadcast ends with the occurrence of the 'LOSE' transition, which empties the transmit buffer.

Before finishing this section, a comment is in order on transition folding. The specification of figure 6 could be made more compact by folding transitions 'TRANSFER' and 'BROADCAST' using the Transition Condition '$e = d \mid e = B$' and changing the Input Condition associated with the arc from place 'Transmit-buffers' to $< s, e, m >$ for the new transition. Exactly the same procedure has been followed in figure 3 (see transition 'M-DATA-indication'). We have chosen not to do so, in order that point-to-point and broadcast modes are clearly separated as this helps with the development of the specifications in the next two subsections.

3.6.2 No Duplication in Broadcast mode

In order to avoid duplication in broadcast mode we must keep a record of the stations to which we have broadcast. In a single CFR this is relatively easy as no simultaneous transmissions by a particular station are allowed due to single buffering. For each station, only a single point-to-point or broadcast transmission is possible and this must have completed (successfully or not) before the next transmission can occur. This allows us to use the list of allowed source-destination pairs stored in 'Acceptable-sources' to determine which station has received a broadcast M-SDU.

The NPN specification is shown in figure 7. It is the same as figure 6, except that an extra place, 'Broadcast-destinations', transition, 'LOSE2', and associated arcs have been added. Two further changes have been made. Firstly, the Transition Condition, $d \neq$ B, has been

Definitions

S: Set of Source Addresses
D: Set of Destination Addresses
M: Set of Messages ($M - data$)
B: Broadcast Address, B$\in D$
M: Monitor Address, M$\in D$
$D' = D \backslash \{B\}$

Initial Marking

M_0(Transmit-buffers)=M_0(Receive-buffers) =M_0(Broadcast-destinations)=$\emptyset$
M_0(Empty-transmit-buffers)$\subseteq S$
M_0(Empty-receive-buffers)=M_0(Empty-transmit-buffers)$\cup\{M\}$
M_0(Acceptable-sources)$\subseteq S \times D'$

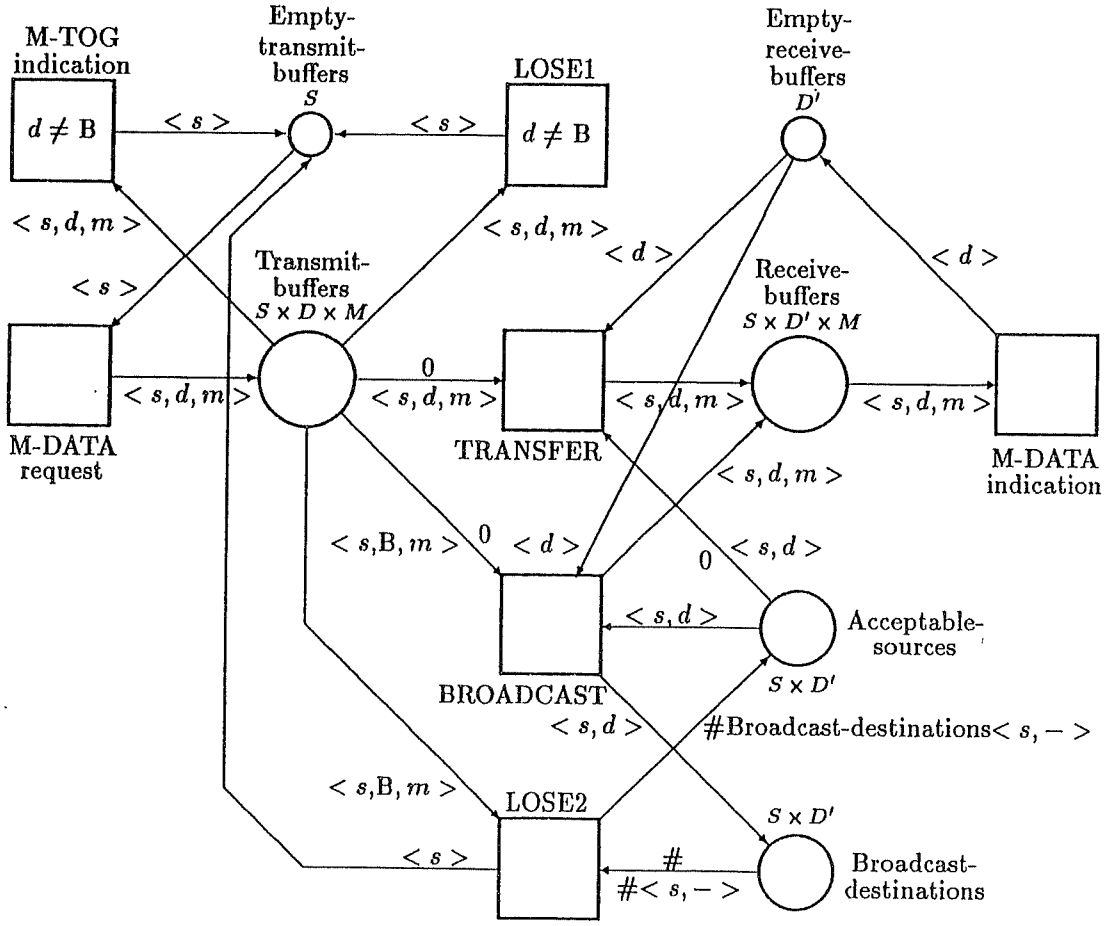


Figure 7: Single CFR M-Access Service: No Duplication for Broadcast M-SDUs

associated with the 'LOSE' transition and it has been renamed 'LOSE1'. 'LOSE1' may only lose point-to-point M-SDUs and 'LOSE2' may only lose broadcast M-SDUs. Secondly, the Destroyed Tokens inscription, '0', has been removed from the input arc from 'Acceptable-sources' to transition 'BROADCAST'. Thus when 'BROADCAST' fires, a destination which will accept M-SDUs from the broadcasting source, is removed from the list of accepting destinations stored in 'Acceptable-sources', and is placed in a list of destinations which have received a broadcast M-SDU. The list is stored in place 'Broadcast-destinations'. The broadcast will continue until the list of accepting destinations is exhausted (there will no longer be a source-destination pair token in 'Acceptable-sources' with the broadcast source address - hence 'BROADCAST' will not be enabled (for this source address) and the only remaining possibility for the broadcast M-SDU is that it is removed from the transmit buffer by firing 'LOSE2') or the M-SDU is lost by firing 'LOSE2'.

'LOSE2' is enabled by a broadcast M-SDU being in a transmit buffer. When it fires, the following actions occur atomically:

1. A particular source's broadcast M-SDU is removed from the transmit buffer;
2. The transmit buffer is marked empty (returned to 'Empty-transmit-buffers'); and
3. All tokens representing broadcast destinations that have successfully received the source's broadcast have been stored in 'Broadcast-destinations'. These tokens are removed and returned to 'Acceptable-sources'.

Of course, any number of stations could be active at the same time.

3.6.3 No Duplication

The single CFR M-Access service with no duplication at all is shown in figure 8. This differs from figure 7 in only two respects. Firstly, the Destroyed Tokens inscription, '0', on the input arc from 'Transmit-buffers' to 'TRANSFER' has been removed. Secondly, an arc from 'TRANSFER' to 'Empty-transmit-buffers' has been added, with a single variable Created Tokens inscription, which is of source address type. Now, when 'TRANSFER' fires, the M-SDU is removed from the transmit buffer and the buffer is marked empty. Hence no duplication can occur.

With the above changes, we have made the assumption that as far as users of the M-Access Service are concerned, the operation of delivery of an M-SDU to a receiving station and the freeing of the transmit buffer can be considered atomic for point-to-point operation.

3.7 Notification Service

The above specifications have included the M-TOG indication primitive as a form of notification service. Of course it is possible not to provide this service by not informing users when the transmit hardware believes that an M-SDU has been lost. This can easily be modelled by deleting the 'M-TOG indication' transition and its associated arcs from the above NPN specifications.

Definitions

S: Set of Source Addresses
D: Set of Destination Addresses
M: Set of Messages ($M - data$)
B: Broadcast Address, B$\in D$
M: Monitor Address, M$\in D$
$D' = D \backslash \{B\}$

Initial Marking

M_0(Transmit-buffers)=M_0(Receive-buffers) =M_0(Broadcast-destinations)=$\emptyset$
M_0(Empty-transmit-buffers)$\subseteq S$
M_0(Empty-receive-buffers)=M_0(Empty-transmit-buffers)$\cup\{M\}$
M_0(Acceptable-sources)$\subseteq S \times D'$

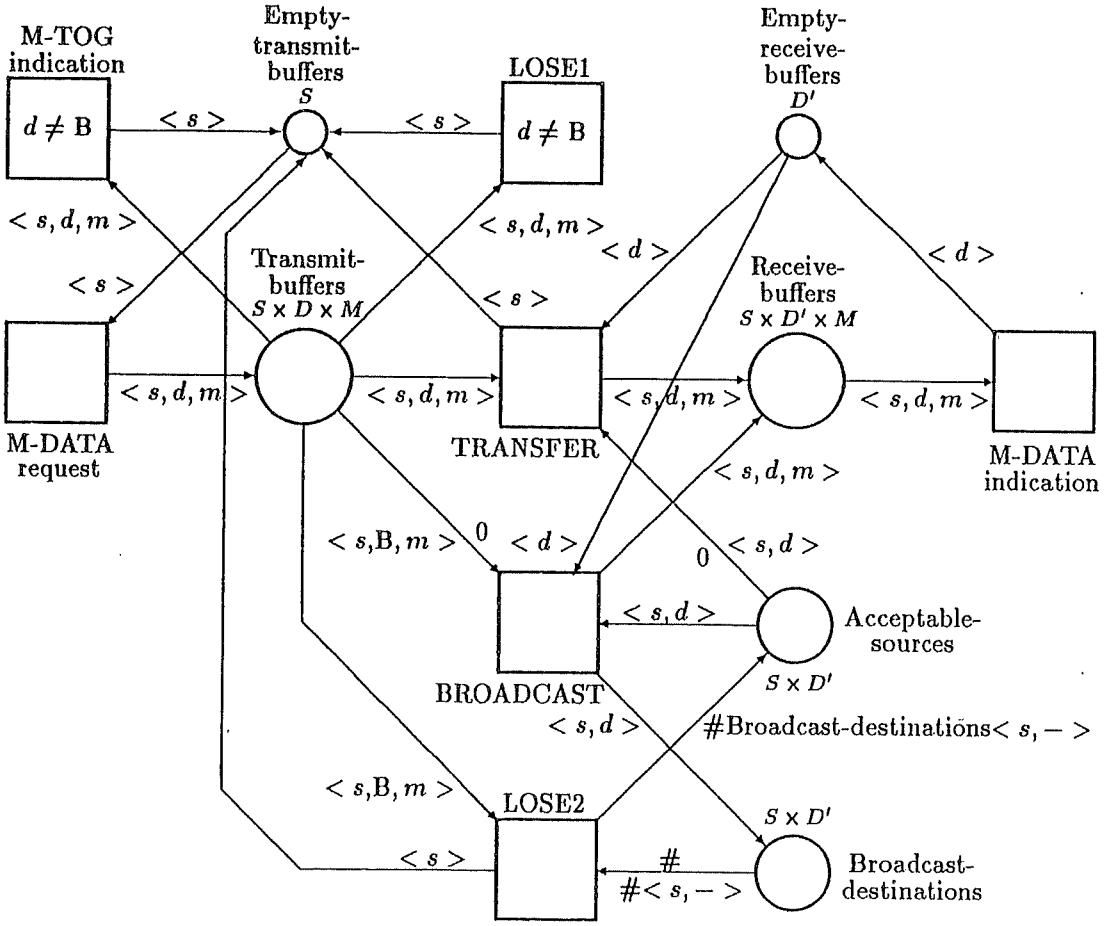


Figure 8: Single CFR M-Access Service: No Duplication

3.8 Comments on Ring Slots

The above NPN specifications model of the M-Access service for a single CFR is more general than that provided by the CFR hardware. It allows there to be simultaneous transmissions (point-to-point or broadcast) by all stations and arbitrary interleaving of all received M-SDUs. The CFR's hardware only permits there to be n simultaneous transmissions, where n is the number of slots on the ring. In most practical CFR installations, the number of stations will be greater than the number of slots.

There would be some point to a further refinement of the NPN specifications to include the slot structure if this was going to be part of a top down design using formal methods. This, however, is not the purpose of this work, which is to provide a formal basis for the development or verification of protocols being designed to operate over CFR systems. We claim that the present level of detail of the M-Access service is sufficient for this purpose.

4 Discussion of NPN Specification

There are a number of matters that require discussion regarding the NPN specification. These include: fairness; finite delay; mandatory sequences versus optional sequences and these are detailed below.

4.1 Finite Delay

The NPN specifications presented above say nothing about the time it takes before a transition fires after it is enabled - the enabling time. This is because nets have abstracted away from time. Hence the enabling time could be anything from zero to infinity. An important property that we would like to preserve in our models is that given a M-DATA request at some point in time we would like to make the temporal assertion that either an associated M-DATA indication or M-TOG indication or a LOSS event occurs some bounded time later. (A temporal logic formulation is being investigated.)

For this to be the case in the net, we must ensure that a transition cannot be enabled indefinitely without firing. This is known as the finite delay property. Another way of looking at this is to consider only those sequences generated by the net where the stop state corresponds to all storage places (eg buffers for M-SDUs) are empty. For example in figures 2 and 3, the stop state is defined by M_0(SP-storage) = $\emptyset$. Hence for a particular M-SDU, the singleton sequence M-DATA request is excluded - it must be followed (at some stage) by one of the three other possible events mentioned above.

4.2 Progress Properties

Another desirable property of the service is that infinite sequences of events must include an M-DATA indication. On the other hand we are quite happy for infinite sequences to exclude the occurrence of either an M-TOG indication or a LOSS event or both. Thus we do not wish the service to be fair to events we would not consider useful.

We would like to guarantee some form of progress property. For example that there exists in every possible sequence, the subsequence "M-DATA request(u,s), M-DATA indica-

tion(u,d)" where u=(s,d,m) is an M-SDU comprising the source address, s, the destination address, d, and the M-data parameter, m; and the second parameter defines the station address at which the primitive occurs.

It appears that the CFR does not support such a progress property. For example, it is possible that every station switches its select register to "receive from nobody". In this case, it is not possible for a M-DATA indication to occur.

We may define a quasi-progress property as follows. 'No infinite sequence will contain an infinite subsequence of LOSS events'. This rules out the possibility of loss of M-SDUs occurring infinitely often.

This is probably true in a single CFR, as loss depends on the probability of a transmission error which is much less than one and hence the probability of an infinite repetition of loss events is zero. However, in ring clusters, M-SDUs may be lost for a number of other reasons. Consider the case when a station on one ring wishes to send M-SDUs to a station on another ring. The receiver may not accept an M-SDU for a number of reasons (eg the source not being selected) and if this is not recovered by retries (which is impossible if the source is not selected) then the M-SDU will be lost as no signal is passed back to the source for a M-TOG indication to occur. Thus an infinite loss sequence is possible.

The above NPN specifications allow the infinite loss case to occur. This appears to be an accurate description in the case of CFR clusters. In the case of a single CFR it may provide too general a model. To overcome this we could do one of two things:

- constrain the model to exclude the offending infinite sequences. This may be done by introducing an extra place to limit the number of LOSS transition occurrences to some finite number. Unfortunately this will increase the state space.
- eliminate the offending sequences when analysing the model.

4.3 Fairness

In the above section we have mentioned that we are quite happy for the service not to be fair to 'LOSS' events and 'M-TOG indications'. We would be delighted if these events never occurred. Another form of fairness that would probably be desirable is that the service should be fair to each of the stations. By this we mean that we want to disallow the behaviour where a set of stations can be locked out of communication with another station indefinitely by yet another station constantly gaining access to it.

The CFR allows a receiver to select a set of stations (the 'hate list') from which it will not accept M-SDUs, so in general it is not fair. However, given that a source station is not on the hate list, we would like to guarantee that eventually it will succeed. The problem is identical to that described in the previous section. We wish to guarantee that the subsequence 'M-DATA request((s,d,m),s), M-DATA indication((s,d,m),d)' occurs in a infinite sequence containing an infinite subsequence of 'M-DATA request((s,d,m),s)'s. Although allowing for this possibility, the NPN specification does not guarantee this behaviour.

4.4 Mandatory Sequences

When defining the service it is important to be able to state which sequences of service primitives are essential and which others are optional. More generally one needs to specify that one or more of a set of sequences is mandatory. Thus in order to conform to the M-Access Service, it is necessary that, there exists a sequence in which "M-DATA request((s,d,m),s), M-DATA indication((s,d,m),d)" is a subsequence. It is now debatable whether or not this should be universally quantified over all source-destination pairs. This is probably too strong, as there will be some destinations which do not want every other source to be able to send them data (cf the "hate list" in the CFR). However, it does seem reasonable to quantify over source addresses. Thus at the very least, each source must be able to send one M-SDU to one other station. On the other hand, it is obviously not mandatory for the service to include sequences which contain LOSS events. It is also necessary that the language of service primitives of the realisation of the service is a sublanguage of that defined in the service specification.

We could therefore consider figures of merit of conformance to a service specification. For example factors in a figure of merit would be the number of sequences that contained LOSS events, and the proportion of LOSS events in the sequence.

5 Conclusions

The service provided by clusters of Cambridge Fast Rings, known as the M-Access Service, has been characterised using a high-level Petri Net. The specification has been divided into a set of 'senders' (one for each station) and 'receivers' (one for each station and the monitor), communicating via a queue in the service provider.

An attempt has been made to clarify the present M-Access service definition and care has been taken to itemise the modelling assumptions.

The specification is presented at various levels of detail. In its most general form, the M-Access Service provider can re-order, duplicate or lose M-SDUs which can be transmitted either to a single destination or broadcast to all stations. This allows a very simple model of the behaviour using a high-level net consisting of just one place (representing a queue of arbitrary size and service discipline) and four transitions (3 representing service primitive occurrences and the forth representing loss of M-SDUs). This specification does not indicate which destinations receive broadcast M-SDUs, only that some broadcast M-SDUs may have been received. In this sense it is incomplete.

At the next level of detail, interaction with users is made explicit. In particular, the list of sources, with which each destination is prepared to communicate (realised in the 'hate list' and 'select register' of the CFR), is specified and this allows the destinations to which broadcasts are received to be detailed. This further detail comes at the expense of 5 extra places and associated arcs.

If the broadcast service is restricted to being duplicate free, then this can be modelled with the addition of one place and 2 transitions and associated arcs. The addition of a further transition allows the complete service to be duplicate free. This is also at the expense of a relatively complex inscription to describe sums of tokens.

A further refinement is presented where the service provided by just a single Cambridge

Fast Ring is modelled. In this specification, the sequence of M-SDUs is preserved and single buffers are modelled for transmitting and receiving M-SDUs (for each station). Duplication and loss are still possible. The list of acceptable sources is included in the specification. The service provider is conveniently modularised into service primitive actions and those associated with its internal operation on M-SDUs: loss; transference (originals or duplicates); and broadcast. Further refinements placing restrictions on the amount of duplication are also presented.

It is claimed that the models are relatively simple (half to one A4 sheet) and allow flow of data to be visualised by executing the net. Sequences of service primitives may also be generated. This allows considerable confidence to be gained in the veracity of the specification. The specification is also very general. The particular model developed here could easily be modified to represent an electronic mail service or connectionless network service for example. It also provides an adequate model for many local area networks of varying topologies (rings, buses, broadcast star networks (eg Hubnet)).

High-level nets allow very general specifications to be modelled quite simply. As these are restricted the models become more complex. The greater the degree of non-determinism and concurrency the simpler the net representation. This facilitates stepwise refinement from general specifications to more specific situations.

Limitations of the approach have been indicated. These concern the need to exclude unwanted infinite sequences and involve notions of fairness. This issue is a subject of research within the net community and elsewhere. It has also been stressed that conformance to a service specification needs to be specified in order to allow systems to be verified.

Although only a particular application is modelled, it is claimed that the technique can be generally applied to the specification of protocol services, including connection-oriented, connectionless and N-way data transmission.

6 Acknowledgements

David Tennenhouse suggested to me that it would be interesting to verify the CFR UDL protocol. This has led to the investigation of the M-Access Service described in this report. Detailed discussions on the operation of the CFR with David and John Porter have been most useful.

The permission of the Director Research, Telecom Australia, to publish this paper is hereby acknowledged.

References

[1] Andy Hopper and Roger M. Needham. *The Cambridge Fast Ring Networking System (CFR).* Technical Report 90, University of Cambridge Computer Laboratory, New Museums Site, Pembroke Street, Cambridge CB2 3QG, England, June 1986.

[2] A. M. Chambers and D. L. Tennenhouse. Communications architectures for the Cambridge Fast Ring. October 1986. Draft Unison Project Document, Ref: UA004.

[3] A. M. Chambers. CFR M-Access service definition. November 1986. Draft Unison

Project Document, Ref: UA008.

[4] D. L. Tennenhouse. The Unison Data Link protocol specification. September 1986. Draft Unison Project Document, Ref: UC022.

[5] G. R. Wheeler. *Numerical Petri Nets - A Definition.* Research Laboratories Report 7780, Telecom Australia, May 1985.

[6] Jonathan Billington, Geoffrey Wheeler, and Michael Wilbur-Ham. PROTEAN: a high-level Petri net tool for the specification and verification of communication protocols. August 1987. to be published in IEEE Transactions on Software Engineering, Special Issue on Computer Communication Systems.

[7] James L. Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice Hall, Englewood Cliffs, N.J., 1981.

[8] Wolfgang Reisig. *Petri Nets, An Introduction.* Volume 4 of *EATCS Monographs on Theoretical Computer Science*, Springer-Verlag, Berlin, 1985.

[9] E. Best and C. Fernandez. *Notations and Terminology on Petri Net Theory.* Arbeitspapiere 195, GMD, January 1986.

[10] W. Reisig. Place/Transition Systems. In W. Brauer, W. Reisig, and G. Rozenberg, editors, *Petri Nets: Central Models and Their Properties*, pages 117 – 141, Springer-Verlag, Berlin, February 1987. Lecture Notes in Computer Science, Vol. 254.

[11] Hartmann J. Genrich and Kurt Lautenbach. System modelling with high-level Petri nets. *Theoretical Computer Science*, 13:109–136, 1981.

[12] G.R. Wheeler. July 1987. Private Communication.

A Appendix: Numerical Petri Nets

This appendix provides an introduction to NPNs, and a subset of notation sufficient for understanding the specification of the CFR. The complete definition of NPNs can be found in [5]. The reader is assumed to have a knowledge of Petri Nets [7,8] or Place/Transition Systems as defined in [9,10].

A.1 Extensions

Numerical Petri Nets are Place/Transition (P/T) Systems with the following extensions.

- Tokens have been generalised to tuples of variables (cf Predicate/Transition (PrT) Nets [11]). This allows the convenient modelling of the parameters associated with service primitives, and in particular M-SDUs.
- A set of data variables is associated with the net. Only very simple types (integer, modulo, boolean, enumerated, strings) are presently considered. A data variable can always be represented by a place, with appropriate input and output arcs and a token carrying its present value. It is introduced purely for modelling convenience for objects such as counters. (This feature is not used in the CFR specification).

- There are two different types of place capacity. The first, K, sets a bound on the number of tokens of a particular value that can be resident in a place (this is the same as for PrT systems). The second, K*, sets a bound on the total number of tokens allowed in a place. (This feature is not used and needs to be generalised to be more useful).

- Three inscriptions are associated with the arcs of the underlying net.

  1. An Input Condition (IC) is inscribed to the left of a transition's input arc, as seen by an observer at the transition. It defines a condition which may be satisfied by a collection of tokens in the associated input place.
  2. The Destroyed Tokens (DT) are inscribed to the right of each input arc, as seen by our observer. It defines a bag (multiset) of tokens, which is removed from the associated input place (by bag subtraction), when the transition fires.
  3. The Created Tokens (CT) are inscribed next to each output arc of a transition. It defines a bag of tokens, which is deposited into the associated output place, when the transition fires.

- There are two optional inscriptions associated with each transition.

  1. A Transition Condition (TC), written next to or inside the associated transition. It defines a condition on net data variables and the variables associated with tokens residing in the transition's input places.
  2. A Transition Operation (TO), written next to or inside its transition. It defines an operation on the data variables. (This feature is not used in the CFR specification).

- An initial Marking (Mo), defining the initial allocation of tokens to each of the places in the net, and the initial value of each of the data variables.

A.2 A Generic Example

An example which illustrates the extensions is shown in figure 9. Places and transitions are given names which are strings of alphanumeric characters commencing with a letter. Places are represented by ellipses (usually circles) and transitions by rectangles or bars. Place capacities are represented by integers written next to the place (eg n or n*). Note that the underlying transitions, places and arcs constitute a directed net.

A.3 Marking

An NPN marking is the net marking (the bags of tokens associated with each place) together with a vector consisting of the value of each of the data variables. The net marking is restricted by the capacities of the places, as is the vector by the type of each variable. The NPN marking may be thought of as the global state of a distributed system.

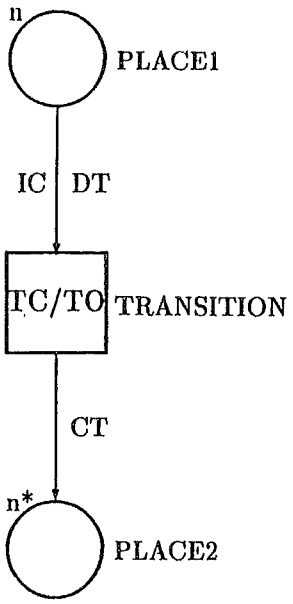


Figure 9: NPN illustrating graphical net elements and generic inscriptions

A.4 Enabling

A transition is enabled when

- the Input Condition is true for each of its input places
- its Transition Condition is true
- if fired, the capacities of its output places will not be exceeded.

A.5 Transition Firing Rule

When a transition fires (occurs), the net marking is changed by the following actions which occur indivisibly and concurrently.

- for each input place, its Destroyed Tokens are removed (bag subtraction)
- for each output place, its Created Tokens are added (bag addition)
- the Transition Operation is performed if it exists.

A.6 Net Execution

An NPN is defined with an initial marking. For this initial marking a set (possibly empty) of transitions will be enabled. (Note that a single transition may be enabled a number of times, if a number of different values of variables satisfy the ICs and TC.) An arbitrary choice is made as to which transition occurs (and which set of allowed variable values are bound). This occurrence generates a new marking, with a new set of enabled transitions. The net can be executed further, generating a set of reachable markings. The complete

set of markings that can be generated this way is known as the Reachability Set. These markings are related to one another by a set of transition firing (occurrence) sequences. A directed graph which relates the set of markings (nodes of the graph) to transition occurrences (edges of the graph) is called a Reachability Graph.

A.6.1 Binding Variables

Free variables (those associated with tokens) may be part of the ICs, DTs, CTs, TC and TO specifications associated with a transition. The scope of these variables is restricted to the transition concerned. When the transition fires, the variables are bound to a particular value via consistent substitution.

A.7 Notation

The following presents the subset of NPN notation used in specifying the CFR M-Access Service. An extension to the notation is defined for bag sums.

In the following: *tok* is a token; $M(p)$ is the bag of tokens in place p; and $mult(x, M(p))$ is the multiplicity of token x in the bag of tokens $M(p)$ and $\emptyset$ denotes the empty bag.

A.7.1 Tokens

In general tokens are tuples of variables/constants separated by commas and enclosed in angular brackets. Variable names are strings of alphanumeric characters commencing with a letter. Some examples are $< 7 >$, $< red, yellow >$ and $< x, y, z >$. Alphanumeric strings may be variables or constants, the context making it clear.

A.7.2 Input Conditions

IC(p,t)	Condition on Input Place Marking, $M(p)$
tok	$tok \in M(p)$
0	$M(p) = \emptyset$
#	$\top$ (always true)

A.7.3 Destroyed Tokens

DT(p,t)	Destroyed Tokens Bag $D(p)$
tok	$\{tok\}$
0	$\emptyset$
#	$M(p)$
'blank'	the "enabling" tokens, $E(p)$

The "enabling" tokens are defined as follows for each input condition.

IC(p,t)	Enabling Tokens Bag $E(p)$
tok	$\{tok\}$
0	$\emptyset$
#	$M(p)$

A.7.4 Created Tokens

CT(t,p)	CREATED TOKENS BAG $C(p)$
tok	$\{tok\}$

A.7.5 Transition Condition

Transition Conditions comprise boolean expressions with the usual logical connectives ('&' for **and**; '|' for **inclusive or**), negation operator ('$\sim$' for **not**) and two place predicates ($<, \leq, =, \geq, >$ and $<>$ for $\neq$). They may involve variables, natural number expressions and string expressions.

A.7.6 Additional Notation

Bag Sums

With broadcast protocols and services we need a convenient notation to describe bag sums over the domains of the token variables. We have two cases of interest:

- A partition of the domain of the marking of a place, which is a set of sets; and
- A partition of the place marking (in general a set of bags).

In the first case a notation has been defined in [5]. We modify it to use $*$ (instead of *underscore*) to indicate variables over which sums are made.

A notation for the second case above is not described in [5]. The need for the notation was raised by the author and the present form of the syntax is a modified and generalised version of that suggested by Geoff Wheeler [12].

Consider a place, p, with arity n_p, a set of domains of variables $\{V_1, \ldots, V_{n_p}\}$ and Marking $M(p)$ given by the multirelation

$$M(p) : V_1(p) \times \cdots \times V_{n_p}(p) \longrightarrow N$$

Let $v_i \in V_i(p)\ \forall i \in \{1, 2, \ldots, n_p\}$, then the notation N$(p)$ and its meaning (the bag $B(p)$) is defined in the following table.

N(p)	BAG $B(p)$
$< v_1, \ldots, v_{i-1}, *, v_{i+1}, \ldots, v_{n_p} >$	$\bigcup_{v_i \in V_i(p)} \{< v_1, \ldots, v_i, \ldots, v_{n_p} >\}$
$\#p < v_1, \ldots, v_{i-1}, -, v_{i+1}, \ldots, v_{n_p} >$	$\sum_{v_i \in V_i(p)} \{m_{v_i} < v_1, \ldots, v_i, \ldots, v_{n_p} >\}$

where $m_{v_i} = mult(< v_1 \ldots, v_i, \ldots, v_{n_p} >, M(p))$. When $M(p)$ is a relation rather than a multirelation, the bag sum reduces to set union.

These may be used in DT and CT inscriptions, for example DT(s,t) =N(p), but when $s \neq p$, we must ensure that the domains of $M(p)$ and $M(s)$ are the same. An abbreviated form of the second notation can be used when the bag sum desired is associated with the same place to which the DT or CT refers, ie when $s = p$. In this case we may drop the p from the start of the notation.

$$\# < v_1, \ldots, v_{i-1}, -, v_{i+1}, \ldots, v_{n_p} > \equiv \#p < v_1, \ldots, v_{i-1}, -, v_{i+1}, \ldots, v_{n_p} >$$

Examples of the notation can be found in the CFR specification. An example of the abbreviated notation and its meaning is as follows.

DT(p,t)	DESTROYED TOKENS BAG $D(p)$
$\# < s, - >$	$\bigcup_{i \in D'} \{< s, i >\} \cap M(Broadcast - destinations)$, with $s \in S$

where $M(Broadcast - destinations) \subseteq S \times D'$.

Optional Place Inscription

In this report we have included for the first time a place inscription which explicitly states the domains of the token variables ie the domain of the marking of the place. The appropriate cartesian product is written next to the place.