1. Bridging

1.1 MAC Relevance to Bridging

The proposed MAC reference model does not prohibit transparent bridging. Any other 802.17 MAC specific bridging method shall:

a) Not prevent communication between stations in a bridged LAN

b) Preserve the MAC service

c) Preserve the characteristics of each bridging method within its own domain

d) Provide for the ability of both bridging techniques to coexist simultaneously on a LAN without adverse interaction.

1.1.1 Preservation of the MAC service

a) A bridge is not directly addressed by communicating end stations, except as an end station for management purposes: frames transmitted between end stations carry the MAC Address of the peer-end-station in their destination address field, not the MAC Address, if any of the Bridge.

b) All MAC Addresses must be unique within a Bridged LAN

c) The MAC Addresses of end stations are not restricted by the topology and configuration of the Bridged LAN

1.1.2 Host MAC Address

One MAC Address per station on the ring is considered. The number of ringlets that this MAC services is not individually addressable in the case of absence of link aggregation. The primitives necessary for resiliency and maintenance of the MAC are part of the MAC Reference model and the MAC layer Management.

1.2 Transparent Bridging

This section extends the ISO/IEC 15802-3 specification to incorporate the 802.17 MAC.

1.2.1 Internal Sublayer Service provided within the MAC Bridge

As specified in ISO/IEC 15802-3.

1.2.2 Support of the Internal Sub layer Service by specific MAC procedures

As specified in ISO/IEC 15802-3.

1.2.3 Support by IEEE Std 802.17 (RPR)

The RPR access method is specified in IEEE Std 802.17. Refer to appropriate sections for frame structure and MAC method.

On receipt of a M_UNITDATA.request primitive, the local MAC Entity performs Transmit Data Encapsulation, assembling a frame using the parameters supplied as specified below. It prepends a preamble and a Start Frame Delimiter before handing the frame to the Transmit Media Access Management component in the MAC Sublayer for transmission.

On receipt of a MAC frame by Receive Media Access Management, the MAC frame is passed to Receive Data Decapsulation, which validates the FCS and disassembles the frame, as specified below, into parameters that are supplied with the M_UNITDATA.indication primitive.

The **frame_type** parameter takes only the value user_data_frame and is not explicitly encoded in MAC frames.

The **mac_action** parameter takes only the value request_with_no_response and is not explicitly encoded in MAC frames.

The source_address parameter is encoded in the source address field of the MAC frame.

The number of octets in the mac_service_data_unit parameter is encoded in the length field of the MAC frame, and the octets of data are encoded in the data filed.

The **user_priority** parameter provided in a data request primitive is not encode din MAC frames. The user_priority parameter provided in a data indication primitive takes the value of the Default User Priority parameter for the Port through which the MAC frame was received.

The **frame_check_sequence** parameter is encoded in the FCS field of the MAC frame. The FCS is computed as a function of the destination address, source address, length, data, and PAD fields. If an M_UNITDATA.request primitive is not accompanied by this parameter, it is calculated in accordance with IEEE Std 802.17, Section X.

1.2.4 Mapping the MAC Service to/from the 802.17 RPR MAC Protocol

This section/clause specifies the mapping of the MAC Service to/from the ISO/IEC 8802-3 MAC Protocol.

1.2.5 Primitive/Parameter and MAC Frame Field Relationships

Figure 1 below shows the mapping of the MA-UNITDATA.request primitive and parameters to the 802.17 MAC frame fields, and the mapping of the 802.17 MAC frame fields to the MA-UNITDATA.indication.

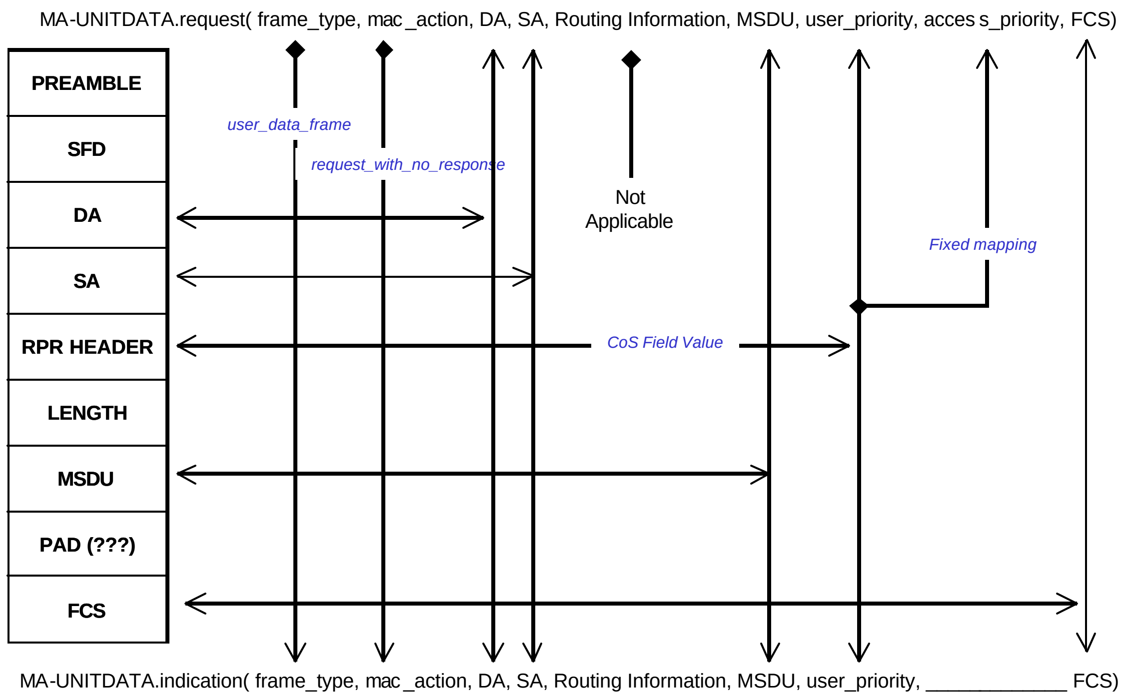

Figure 1: Mapping of MAC Service primitiveds to/from 802.m MAC frames

The MAC frame structure is defined in Section outlining 802.17 MAC.

The 802.17 MAC Protocol provides a single priority service regardless of the priority requested.

The 802.17 MAC Protocol does not provide the capability of transferring the requested priority to the remote MAC Service user.

1.2.6 The Forwarding Process

See IEO/IEC 15802-3 Std, section 7.7.

1.2.6.1 Transparent Bridging using VLAN tags

Compliant to IEEE Std 802.1Q.

1.2.7 Mapping Priority

The user-priority parameter in an M_UNITDATA.request primitive shall be equal to the user_priority parameter in the corresponding data indication.

The mapping of user_priority to outbound access_priority is achieved via fixed, MAC method-specific mappings. The access_priority parameter in an M_UNITDATA.request primitive shall be determined from the user_priority in accordance with the values shown in Table 1 for the MAC method that will receive the data request. The values shown in Table 1 are not modifiable by management or other means.

Table 1: Outbound access priorities

user_priority	Outbound Access Priority per MAC method									
	802.3	802.17	8802-4	8802-5 (default)	8802-5 (alternate)	8802-6	802.9a	8802.11	8802-12	FDDI
0	0	0	0	0	4	0	0	0	0	0
1	0	1	1	1	4	1	0	0	0	1
2	0	2	2	2	4	2	0	0	0	2
3	0	3	3	3	4	3	0	0	0	3
4	0	4	4	4	4	4	0	0	4	4
5	0	5	5	5	5	5	0	0	4	5
6	0	6	6	6	6	6	0	0	4	6
7	0	6	7	6	6	7	0	0	4	6

1.3 The Spanning Tree Algorithm and Protocol

Compliant to ISO/IEC 15802-3 Std, Section 8.

1.4 Encapsulation Bridging

1.4.1 Using the RPR header for encapsulation bridging

Indication of encapsulation

1.5 Bridging and Link Aggregation

One Aggregation MAC address for the aggregate link

1.6 802.17 MAC compatibility with 802.1 standards

1.6.1 802.1w over 802.17

Annex B(informative)

Encapsulated Bridging

This technique utilizes the 802.17 MAC frame header **type field** to indicate that the payload contains an encapsulated 802.3 frame. The technique in essence makes the 802.17 ring appear conceptually as virtual multi port bridge, where each node on the ring is a port.

For the purposes of the following example a bridge is defined as a device that supports both 802.3 interfaces, and 802.17 interfaces. Such a bridge is referred to as an 802.17 Encapsulating Bridge.

The examples that follow are based on the following network diagram.

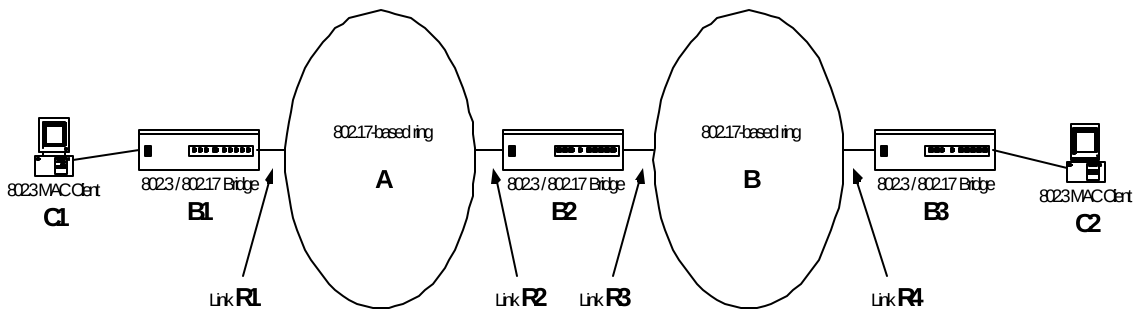

Figure B.1 – Model for Encapsulated Bridging

Device C1 is a client that can emit and receive 802.3 frames. It is able to send and receive such frames to and from bridge B1.

Bridge B1 performs the Encapsulation function between the C1 LAN and the 802.17 Ring A.

Bridge B1 performs the De-encapsulation function between the 802.17 Ring A and the C1 LAN.

Link R1 is an 802.17 connection. Frames traveling over the 801.17 ring are in the standard 802.17 MAC Frame Format.

Link R2 is an 802.17 MAC interface to Bridge B2 from the 802.17 Ring A.

Link R3 is an 802.17 MAC interface from Bridge B2 to the 802.17 Ring B.

Link R4 is an 802.17 MAC interface from the 802.17 Ring B to Bridge B3.

Device C2 is a client that can emit and receive 802.3 frames. It is also able to send and receive such frames to and from bridge B3.

Frame formats and the Encapsulation Process from Client C1 to Client C2

Device C1 will emit an 802.3 compliant frame as follows:

802.3 MAC DA	802.3 MAC SA	802.3 Type	Payload	FCS

Figure B.2 – Client 802.3 MAC Frame format

This frame will be received by bridge B1. Assuming Bridge B1 has an empty filtering database, the frame will be placed onto the 802.17 ring A with a broadcast destination address. Thus, the frame will be as follows:

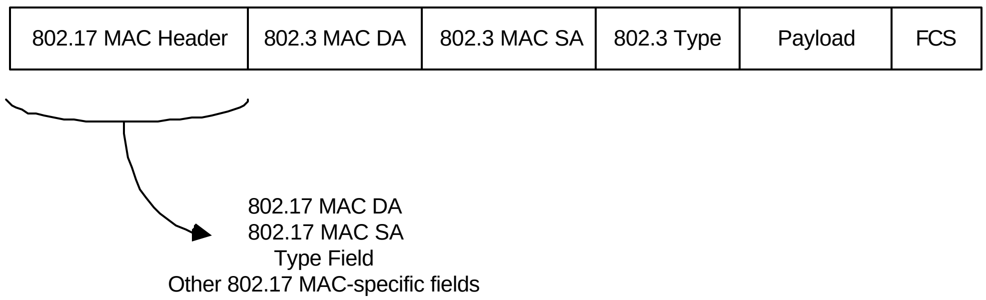

Figure B.3 – Ring A Encapsulated Frame (802.17 Encapsulating 802.3)

The values for the 802.17 fields will be as follows:

802.17 MAC DA	Broadcast, FF FF FF FF FF FF
802.17 MAC SA	The MAC address assigned to the 802.17 MAC interface shown in figure B.1 as link R1.
Type Field	A value to indicate 802.3 encapsulation.

Table B.2 – Field Values for the 802.17 Frame placed onto Ring A in Figure B.1

When Bridge B2 receives the 802.17 MAC Frame from Ring A, the frame is de-encapsulated on ingress to the bridging function.

In the example, the only egress port is connected to the 802.17 MAC on Ring B. Therefore the frame is re-encapsulated for transport over Ring B. The contents of the 802.17 header will be as follows:

802.17 MAC DA	Broadcast, FF FF FF FF FF FF
802.17 MAC SA	The MAC address assigned to the 802.17 MAC interface shown in figure B.1 as link R3.
Type Field	A value to indicate 802.3 encapsulation.

Table B.3 – 802.17 MAC Header Contents for transport on Ring B to Bridge B3

The frame will be received by Bridge B3 and de-encapsulated.

The 802.3 frame (that was the payload for the 802.17 frame) is then broadcast on all eligible ports on Bridge B3. This includes the 802.3 MAC Client C2.

A port is eligible to broadcast the 802.3 frame based on 802.1D and 802.1Q forwarding rules.

Frame formats and the Encapsulation Process from Client C2 to Client C1

A process defined in 802.1D called Learning allows each Bridge between 802.3 MAC Clients C1 and C2 to be more intelligent about the way a frame is transported on the return path. This is because each bridge learns the source MAC addresses for the 802.17 nodes on both rings A and B. Therefore, there is no need to perform the broadcast action that was required from C1 to C2 since now the node 802.17 MAC addresses are known.

From Client C2 to Bridge B3, the frame will be formed as follows:

802.3 MAC DA C1	802.3 MAC SA C2	802.3 Type	Payload	FCS

Figure B.4 – Format of 802.3 Frame from C2 to Bridge B3

Bridge B3 will then perform a filtering database lookup, compliant with 802.1D; the result of which indicates the destination port on bridge B3 is for the 802.17 MAC on ring B. Furthermore the result returns the 802.17 MAC Node address with which the frame will be encapsulated for transport over ring B.

Thus, the frame which is transported from link R4 to link R3 will be formatted as follows:

802.17 MAC DA R3	802.17 MAC DA R4	Encap Type	Misc 802.17 Fields	802.3 MAC DA C1	802.3 MAC SA C2	802.3 Type	Payload	FCS

802.17 MAC Header

Figure B.5 – Format of 802.17 Frame from R4 to R3

Bridge B2 receives the frame. The frame is de-encapsulated and the 802.3 MAC DA is looked up according to 802.1D. Learning also takes place on the 802.3 MAC source address. The result (by process of learning on the C1 – C2 frame) returns a destination port of R2, which is an 802.17 MAC. Thus the frame will be transported across Ring A using 802.17 MAC Encapsulation and the frame will appear as follows:

802.17 MAC DA R1	802.17 MAC DA R2	Encap Type	Misc 802.17 Fields	802.3 MAC DA C1	802.3 MAC SA C2	802.3 Type	Payload	FCS

802.17 MAC Header

Figure B.6 – 802.17 MAC Frame transported across Ring A from R2 to R1.

When the frame is received at bridge B1, the frame is de-encapsulated. The 802.3 Header is then processed according to rules defined in 802.1D. The result of the 802.1D lookup is an action to forward the frame to port attached to 802.3 MAC Client C1.

802.1D Learning and Destination Address Lookups

The learning process that occurs on Bridge B1, B2 and B3 is used traditionally in a bridge to determine where a specific 802.3 MAC address lives. Therefore, when an 802.3 frame arrives on a bridge port, a lookup is performed on the destination 802.3 MAC address to determine the destination port, and an additional lookup is performed on the 802.3 MAC Source address to determine if the frame is either 1) already learned against that same bridge port, 2) the 802.3 MAC Source address is new and must be added to the filtering database or 3) the 802.3 MAC source address is present in the filtering database, but is present for a different bridge port, therefore indicating that the device associated with this MAC address has moved and that the filtering database must be updated.

The associated data (the data returned as the result of a lookup) must be enhanced since the 802.17 MAC bridge port needs an extra level of indirection; that being the 802.17 Ring Node MAC destination address.

Thus, an 802.3 MAC Address destination or source lookup performed on a bridge supporting 802.17 will return associated data as follows:

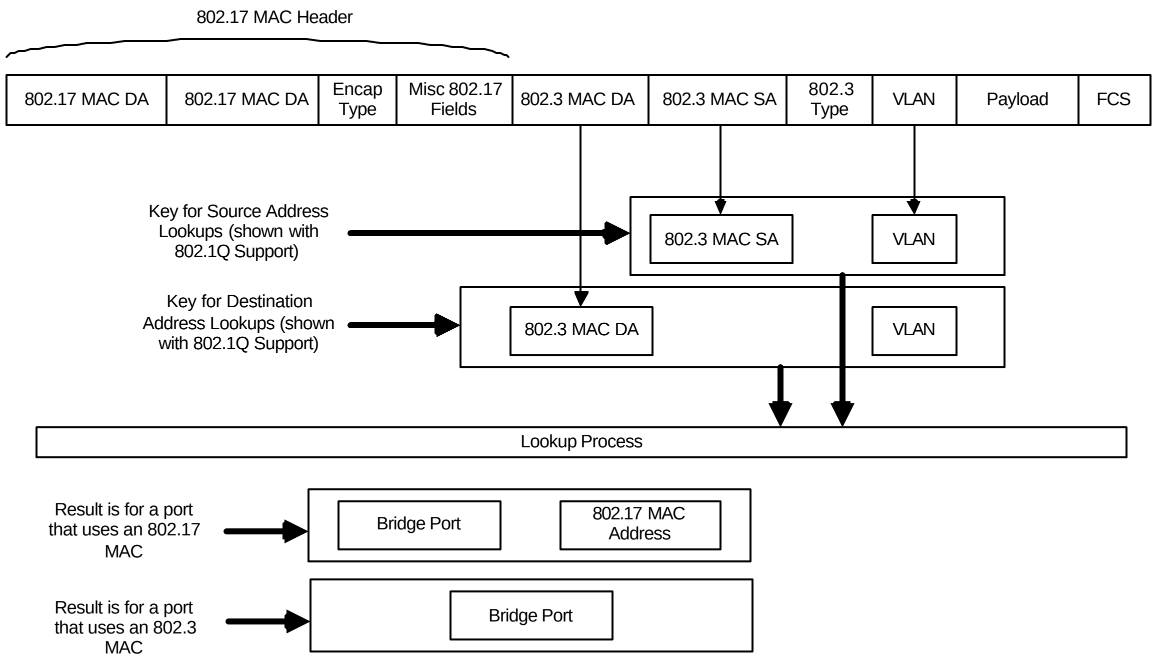

Figure B.7 – 802.3 MAC Source and Destination Address Lookup Keys

In the case of the example, the lookups for the 802.3 MAC Destination address works as follows:

Bridge B1 receives an 802.3 frame with destination address of C2. However, as stated earlier, the filtering databases are empty. Therefore an 802.3 MAC Destination Address lookup is performed which fails. Therefore the frame must be broadcast on the 802.17 Ring A, hence the destination address of FF FF FF FF FF FF. Bridge B1 also looks up the 802.3 MAC Source Address and does not find an entry. It therefore adds an entry as follows:

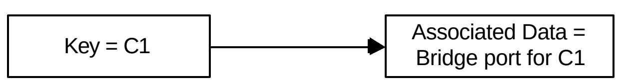

Figure B.8 – Bridge B1 Filtering Database after learn from C1

Bridge B2 then receives the 802.17 frame and performs a lookup for the 802.3 MAC Destination Address. This fails. It then performs a lookup on the 802.3 MAC Source Address. This fails also. It therefore adds an entry to the filtering database as follows:

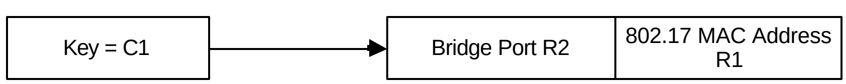

Figure B.9 – Bridge B2 Filtering Database after learn from C1

The frame is sent over ring B using the 802.17 MAC Destination address of FF FF FF FF FF FF. When the frame arrives at Bridge B3 a lookup is performed on the 802.3 MAC Destination Address. This fails. A lookup is performed on the 802.3 MAC Source address. This fails. The following entry is then added to the filtering database:

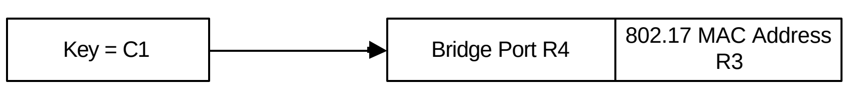

Figure B.10– Bridge B3 Filtering Database after learn from C1

The frame is the forwarded to C2 (by way of a unknown destination broadcast). Should C2 then respond to C1, it would build a frame with C1 as the destination address in its 802.3 frame.

Lookups performed in the other direction would succeed for all of the 802.3 destination address lookups, returning appropriate 802.17 MAC Destination addresses such that frames in said direction will no longer be broadcast on ring A and ring B. Such frames on ring A and ring B would have explicit 802.17 Node MAC addresses in their destination address fields.

Learning would also occur in the return direction (C2-C1) such that any future communication between C1 and C2 would also yield successful 802.3 MAC destination address lookups making communication in this direction more efficient since it would not have to resort to 802.1D's unknown destination broadcast facility.