IEEE 802.11
Wireless Access Method and Physical Layer Specifications

Title: **Proposed Text for Fragmentation/Reassembly at the MAC layer**

Author
Rick White
Motorola, Inc.
Wireless Data Group
50 E. Commerce Drive Suite M-1
Schaumburg, Illinois 60173
Tel: 1-708-576-7878
Fax: 1-708-576-7907
E-Mail: rick_white@wes.mot.com

Abstract: This submission proposes text for fragmentation/reassembly that should be used by the editors for inclusion in the draft 802.11 standard. It is based on the fragmentation proposal presented in [1] and the transmitter priority proposed in [2].

Fragmentation / Reassembly

The process of partitioning a MAC Service Data Unit (MSDU) into smaller MAC level frames, MAC Protocol Data Units (MPDUs), is defined as fragmentation. The process of recombining MPDUs into a single MSDU is defined as reassembly. The MAC Service interface has a maximum MSDU_Size defined in octets. Also each PHY has a Max_Full_MPDU and a Min_Full_MPDU size defined in octets. They define the range of maximum MPDU sizes that the PHY will support and each PHY may have different values. The values for Min_Full_MPDU and Max_Full_MPDU are defined in the standard for each PHY. Max_Full_MPDU defines the maximum full size MPDU that the PHY can transmit. Min_Full_MPDU defines the minimum full size MPDU for a PHY. There is also a Cur_MPDU defined in octets that is a manageable object. Cur_MPDU defines the current maximum size of a MPDU that can be sent to the PHY This allows an entity at the MAC layer or higher to define what the maximum MPDU the PHY will support at any given point in time. The value of Cur_MDPU must be less than or equal to Max_Full_MPDU and greater than or equal to Min_Full_MPDU. This is illustrated in Figure 1.

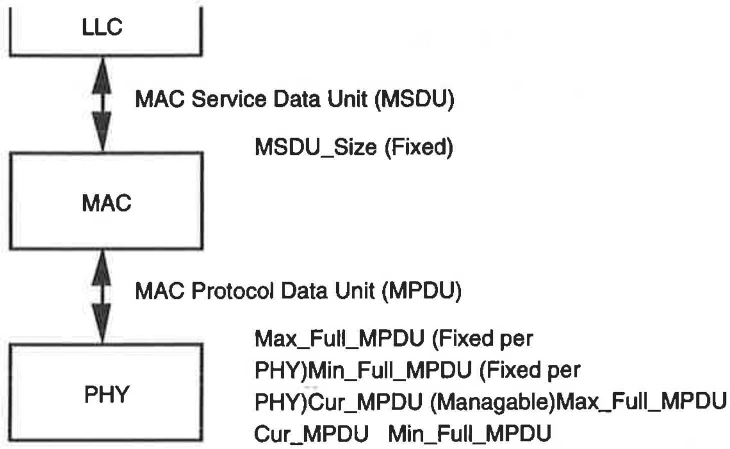

Figure 1: MPDU and MSDU Definitions

When a frame is received from the LLC with a MSDU size greater than the payload of Cur_MPDU, the frame must be fragmented. The MSDU is divided into MPDUs. This is illustrated in Figure 2.

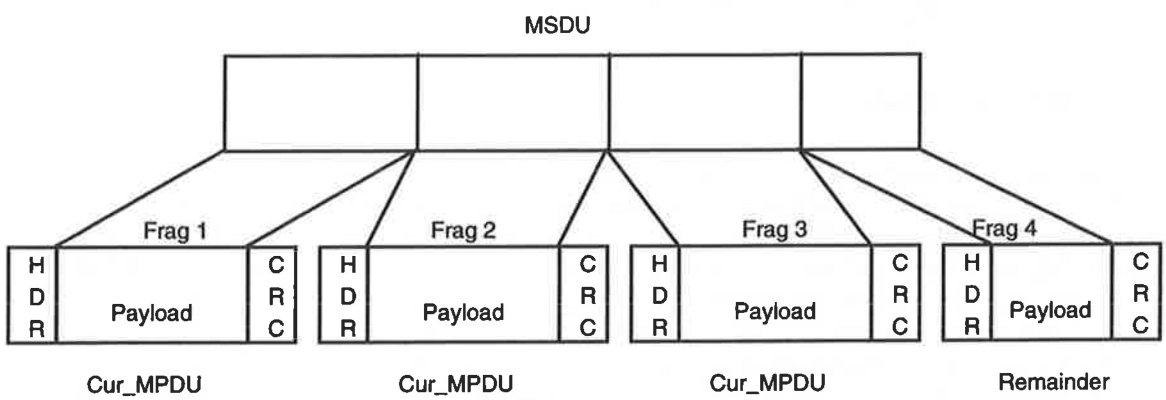

Figure 2: Fragmentation

Fragmentation

The MAC will fragment and reassemble MSDUs. The fragmentation and reassembly mechanisms allows for fragments to be retransmitted. For the purposes of this description a 'dwell time' will refer to the duration of time spent on a single frequency in a FH system. Therefore in a FH PHY the PHY will hop to the next frequency in the hop sequence at the end of the current dwell time. For other systems a 'dwell time' will refer to the period of time spanning from the start of transmission of a TIM until just before the start of transmission of the next TIM.

Whenever possible, the size of the payload of a fragment shall be some fixed number of bytes. Let this number of bytes be denoted by Cur_MPDU_Payload. Cur_MPDU_Payload equals Cur_MPDU minus MAC Header minus CRC. The payload of a fragment shall never be larger than Cur_MPDU_Payload. However, the size of the payload may be less than Cur_MPDU_Payload.

When data needs to be transmitted, the number of bytes in the payload of the fragment shall be determined based on the time at which the fragment is to be transmitted for the first time. Once the fragment is transmitted for the first time, its contents shall be fixed until it is successfully delivered to the destination station.

The number of data bytes in the payload of a fragment shall depend on the values of the following two variables at the instant the fragment is to be transmitted for the first time:

1. The time remaining in the current dwell time.
2. The number of bytes in the MSDU that have not yet been transmitted for the first time.

Since the control of the channel will be lost at a dwell time boundary and the station will have to contend for the channel after the dwell boundary, it is required that the acknowledgment of a fragment be transmitted before the stations cross the dwell time boundary. Hence, if there is not enough time remaining in the dwell time to transmit a fragment with an Cur_MPDU_Payload byte payload, the number of bytes in the payload shall be reduced to the maximum number of bytes that will allow the fragment plus the MAC acknowledgment to fit within the time remaining in the dwell time. This strategy will (on the first transmission attempt) prevent the time near the end of the dwell time from being wasted. This is shown in Figure 3.

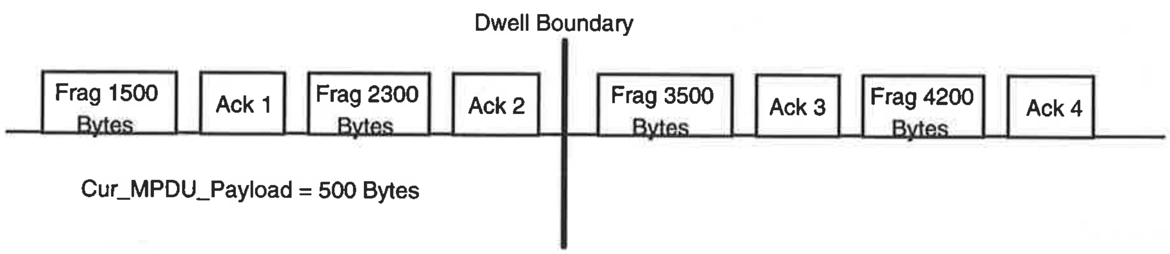

Figure 3: Fragmentation Near a Dwell Boundary

Referring to Figure 3, a 1500 byte MSDU is fragmented into four fragments with Cur_MPDU_Payload set at 500 bytes. There is enough time left in the dwell to send two fragments, one of 500 bytes and a second of 300 bytes. After the dwell boundary, the rest of the MSDU is sent, one 500 byte fragment and one 200 byte fragment.

However, the number of bytes in the payload of a fragment shall be greater than some (configurable) minimum quantity, denoted Min_MPDU_Payload. If there is not enough time remaining in the dwell time to fit a fragment with a Min_MPDU_Payload byte payload plus acknowledgment, the station shall not create another fragment for transmission in the current dwell time. The station shall wait until the next dwell time to create and transmit a fragment with a Cur_MPDU_Payload byte payload (provided there are at least Cur_MPDU_Payload more bytes remaining in the packet). Stations can violate this minimum payload size rule only if the remaining number of bytes in the packet is less than Min_MPDU_Payload. If the number of bytes that have not yet been transmitted is less than Min_MPDU_Payload, the last fragment of the packet will have to contain less than Min_MPDU_Payload bytes. (I.e., the payload of the fragment will not be padded to bring its length up to Min_MPDU_Payload.)

If a fragment requires retransmission, its contents and length shall remain fixed for the lifetime of the MSDU at that station. In other words, after a fragment is transmitted once, contents or length of that fragment are not allowed to fluctuate to accommodate the dwell time boundaries. By fixing a fragments size for the lifetime of the packet the overhead associated with the fragmentation process is minimized and the fragment reassembly process is simplified. Let the *fragmentation set* refer to the contents and length of each of the fragments that make up the data packet. The fragmentation set is created at a station as soon as the fragments are attempted for the first time. The fragmentation set remains fixed for the lifetime of the packet at the transmitting station. This is shown in Figure 4.

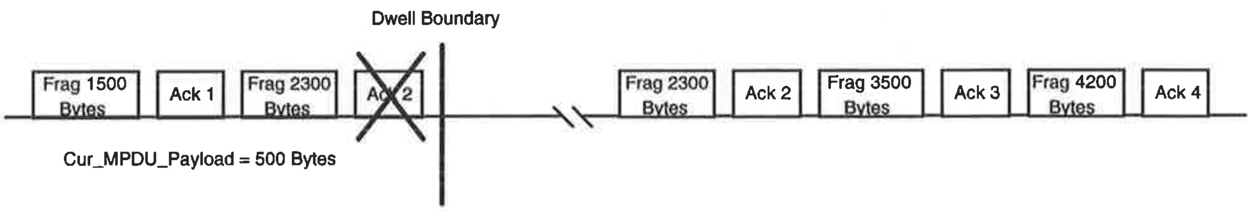

Figure 4: Fragmented MSDU with missed ACK Near a Dwell Boundary

In the example shown in Figure 4, the same 1500 byte MSDU is fragmented at the same point in the dwell time as in Figure 3 but the ACK for the second fragment is missed. After the dwell boundary, the fragment is retransmitted and the fragment size remains 300 bytes.

Each fragment will contain a fragment ID number. When a station is transmitting a MSDU, the fragments will be in order of lowest ID to highest ID. The fragment ID also contains a bit that indicates the last fragment of the MSDU.

If, when retransmitting a fragment, there is not enough time remaining in the dwell time to allow transmission of the fragment plus the acknowledgment, the station shall wait until the start of the next dwell time before retransmitting that fragment.

Reassembly

Each data fragment must contain enough information to allow the complete data packet to be reassembled from its constituent fragments. The header of each data fragment contains the following information that is used by the destination station to correctly reassemble the complete data packet:

- Frame type (data, acknowledgment, etc.).
- Source address
- Destination address
- MPDUID. This field allows the receiver to check that all incoming fragments belong to the same data packet.
- A fragment number. Fragments of an MSDU are numbered sequentially, 1, 2, 3, etc.
- A last fragment indicator to inform the receiver that the fragment ID of the fragment corresponds to the last fragment of the MSDU. Only the last fragment of the MSDU will have this bit set to one. All other fragments of the MSDU will have this bit set to zero.

The destination station can reconstruct the packet by piecing together fragments in order of increasing fragment ID number. If the fragment with the last fragment bit set to one has not yet been received, then the destination station knows that the data packet is not yet complete. As soon as the station receives the fragment with the last fragment bit set to one, the station knows that no more fragments will be received for the MSDU.

To properly reassemble packets, a destination station must discard any duplicated fragments received over the RF. If a station receives a fragment with the same source, destination, sequence number, and fragment number as a previous fragment, then the station must discard the duplicate fragment.

Control of the Channel

Transmitter priority is used to provide an efficient MSDU delivery mechanism. Once a station has contended for the channel, it will maintain control of the channel until it has sent all of the data frames of a MSDU. After all frames have been transmitted, the station will relinquish control of the channel.

Once the station has contended for the channel, it will continue to send frames until either all frames of a fragmented MSDU have been sent, an acknowledgment is not received, or the station can not send any additional frames due to a dwell time boundary.

Figure 5 illustrates the transmission of a multiple fragment MSDU using transmitter priority.

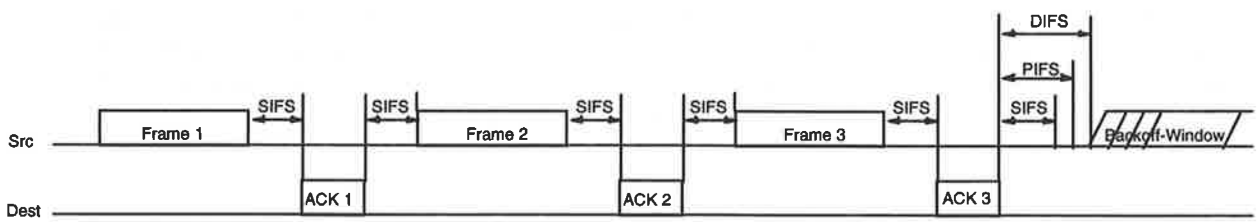

Figure 5 : Transmission of a Multiple Fragment MSDU using Transmitter Priority

The source station transmits a data frame then releases the channel and waits for an acknowledgment. When the source station releases the channel following its data frame, it will immediately turn its radio around and monitor the channel for an acknowledgment frame from the destination station.

When the destination station has finished sending the acknowledgment, the SIFS following the acknowledgment is then reserved for the source station to continue (if necessary) with another data frame. The station sending the acknowledgment does not have permission to transmit on the channel immediately following the acknowledgment.

If the source station receives an acknowledgment but there is not enough time to transmit the next frame and receive an acknowledgment, it will contend for the channel at the beginning of the next dwell time.

If the source station does not receive an acknowledgment frame, it will attempt to retransmit the data frame at a later time (according to the backoff algorithm). When the time arrives to retransmit the data frame, the source station will contend for access in the contention window.

After a station contends for the channel to retransmit a data frame of a multiple fragment MSDU, it will start with the last data frame that was not acknowledged. The destination station will receive the fragments in order since the source sends them one at a time, in order. It is possible however, that the destination station may receive duplicate packets. This will happen if the destination station sends an acknowledgment and the source does not receive it. The source will resend the same data frame after executing the backoff algorithm and contending for the channel.

A station will transmit after the SIFS only under the following conditions (for transmitter priority):

- The station has just received a frame that requires acknowledging.
- The source station has received an acknowledgment to a previous frame, has more data frame(s) for the same MSDU to transmit, and there is enough time left in the dwell time to send the next frame & receive an acknowledgment.

The following guidelines also apply.

- When a station has transmitted a frame other than a data frame, it does not have priority to transmit on the channel following the acknowledgment for that frame.
- When a MSDU has been successfully delivered, the station does not have priority to transmit on the channel following the last acknowledgment of the last data frame.
- Only unacknowledged data frames need to be retransmitted.

If a multiple frame MSDU does not require an acknowledgment (for example, a broadcast/multicast packet transmitted by the Access Point), the source station will transmit all frames of the MSDU without releasing the channel as long as there is enough time left in the dwell time. If there is not, the station will transmit as many frames as possible and recontend for the channel during the next dwell time.

RTS/CTS usage with Fragmentation

The following is a description of using RTS/CTS for a fragmented MSDU. The RTS/CTS defines the duration of the first frame and acknowledgment. A duration field is defined in the data and acknowledgment frames that specifies the duration of the next frame and acknowledgment. This is illustrated in Figure 6.

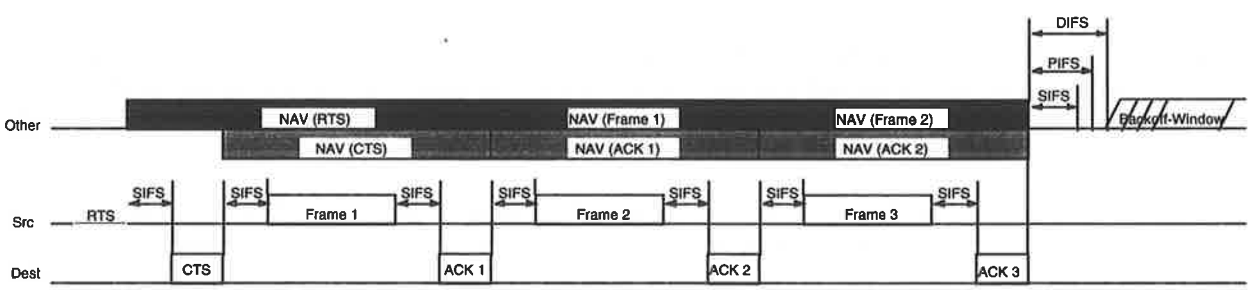

Figure 6 : RTS / CTS with Fragmented MSDU

Each frame contains information that defines the duration of the next transmission. The RTS will update the NAV to indicate busy until the end of ACK 1. The CTS will also update the NAV to indicate busy until the end of ACK 1. Both Frame 1 and ACK 1 will update the NAV to indicate busy until the end of ACK 2. This is done by using the duration field in the DATA and ACK frames. This will continue until the last Frame and ACK which will have the duration set to zero. Each Frame and ACK acts as a virtual RTS and CTS. The Frames and ACKs do not need to be short like RTS and CTS because the channel has already been marked busy.

In the case where an acknowledgment is not received by the source station, the NAV will be marked busy for next frame exchange. This is the worst case situation. This is shown in Figure 7. If the acknowledgment is not sent by the destination station, stations that can only hear the destination station will not update their NAV and be free to access the channel. All stations that hear the source will be free to access the channel after the NAV from Frame 1 has expired.

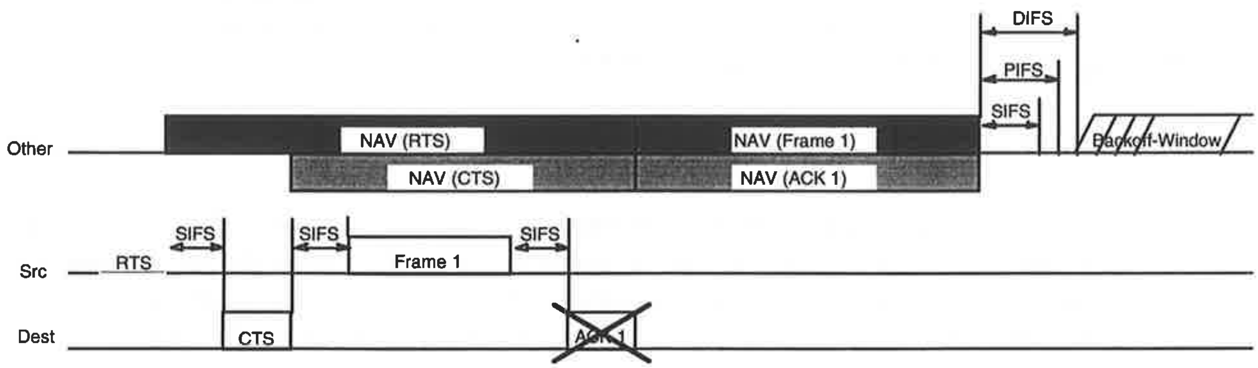

Figure 7: RTS / CTS with Transmitter Priority with missed Acknowledgment

Frame Formats

Data Frames

The frame format for a Data frame is shown in Figure 8. The Fragment Number is always required in the data frame even if the MSDU is a single data frame.

Fixed Hdr	NID	DEST	SRC	Duration	Fragment Number	Payload	CRC32
5	3	6	6	2	1	1 - Cur_MPDU_Payload	4

Figure 8: Data Frame

The Fixed Header consists of the Type, Control, and MPDUID fields. The NID field is a 3 octet field that defines the Network ID. The DEST field is the 48 bit IEEE address of the destination. The SRC field is the 48 bit IEEE address of the source. The Duration field defines the medium occupancy time from the end of the Data frame to the end of the ACK frame for the next MSDU fragment Data frame. The Fragment Number defines the fragment number within a given MPDU. More detailed information of the Fragment Number field is given below. The payload is all or part of the MSDU. It ranges in size from 1 to Cur_MPDU_Payload octets. The CRC32 field is 32 bit FCS generated over the entire frame, from beginning of the Fixed Header.

The Fragment Number field of a Data frame in shown in Figure 9.

Last Frag	Fragment Number
MSB	LSB

Figure 9: Fragment Number Field for a Data Frame

The Last Fragment field is a single bit and indicates if the current fragment is the last fragment of the MSDU. The Fragment Number field is a binary representation of the fragment number of the MSDU (fragment 1 - 0000001, fragment 2 - 0000010 . . .)

Acknowledgment Frame

The frame format for the acknowledgment frame is shown in Figure 10. The Duration field defines the medium occupancy time from the end of the ACK to the end of the ACK for the next MSDU fragment Data frame. The fragment number field indicates the fragment number that is being acknowledged. It has the same format as the Fragment Number field in the Data frame.

Fixed Hdr	Duration	Fragment #	CRC 8
5	2	1	1

Figure 10: Acknowledgment Frame

References

[1] Mark Demange - Motorola, et al, "Fragmentation/Reassembly at the MAC Layer", Doc IEEE P802.11-94/37 March. 1994.

[2] Rick White - Motorola, "Transmitter Priority in the MAC Layer", Doc IEEE P802.11-94/106, May 1994.