IEEE P802.11

Wireless Access Method and Physical Layer Specification

Updated MAC State Machines

Michael Fischer
Digital Ocean, Inc.
42424–3 Medical Drive
San Antonio, TX 78229
Telephone: +1–210–614–4096
Facsimile: +1–210–614–8192
email: mfischer@CHILD.com

Abstract

This submission contains a replacement for section 6.7 (MAC state machines) of the D1.2 draft standard. The current section (formerly 5.8) is sufficiently out of date that only complete replacement is appropriate to bring the state machines to a level which matches the remainder of the document. This submission is an attempt to create state machines which properly describe the MAC, not to redefine MAC operation. The author of this submission requests that reviewers inform him directly about errors, omissions, and ambiguities in this set of state machines. If these state machines are adopted for inclusion in the draft, the author of this submission volunteers to work with the editors to correct these state machines and update them to match other changes to the draft. NOTE: In some cases the information needed for or yielded from certain state transitions exposes under– or over–defined aspects of the text of other sections of the draft. This document makes no attempt to supply or change such text. Also, the creation of these state machines began with the D1 draft, and there may be places where update to D1.2 is incomplete — still these state machines are believed to be much superior to those in the D1 through D1.2 drafts, and a much better starting point for further corrections and improvements.

NOTE about the 95/014r1 update: The file containing document 95/014 submitted at the end of the closing session of the July meeting seems to have been severely corrupted, containing some of the correct material, some of the original state machines from D1.0, and some material from other, unrelated files. The 95/014r1 version contains the correct material, and was assembled using cut and paste rather than Microsoft OLE. There still seem to be some problems editing the diagrams from within Word 6 after pasting, so a separate PowerPoint 4 file with the diagrams is provided as document 95/014Ar1. In addition, at the request of the 802.11 Editors, document 95/014r1 includes updates for some, although not all, of the other changes to the MAC–related sections of the draft adopted at the July meeting. It is believed that the appropriate changes are incorporated due to the motions adopted from documents 95/038, 95/139, 95/140, 95/149r1, and 95/150.

Desired Outcome

Adoption of the material from this document as a replacement for section 6.7 of the D1.2 draft.

6.7. MAC State Machines

MAC operation at all stations is described by six communicating state machines. A seventh state machine is used at APs to provide distribution services. All of these state machines may operate concurrently. The functions of these state machines are summarized below and detailed in the remainder of this section. The relationship of and communication among these state machines is illustrated in figure 6–xx(1). In this figure, solid arrows show paths used to transfer frames or fragments, as well as control and status information, whereas shaded arrows show paths used only for control and status transfers.

1. The **MAC Data Service (MD) state machine** provides the MAC data service interface to the LLC sublayer.

2. The **MAC Management Service (MM) state machine** provides the MAC management service interface to the adjacent sublayer management entity and station management entity.

3. The **Distribution Services (DS) state machine** exists only at access points, and provides distribution services and interfaces to both wired and wireless distribution systems. This state machine always operates at an AP, even when no distribution system medium is available.

4. The **MAC Control (C) state machine** provides the distributed coordination function and (optionally) the point coordination function for transfer of frames over the wireless medium provided by the PHY layer. This state machine also provides fragmentation, reassembly, part of power management (detection of frames addressed to stations that might not be awake), part of multirate support (duration calculation), and (optionally) the encryption and decryption functions for the privacy function.

5. The **MAC Management (M) state machine** provides the MAC management functions of the station, including time synchronization, power management, authentication, association/reassociation, and scanning. This state machine also has the primary responsibility for maintaining the MAC management information base.

6. The **Transmitter (T) state machine** handles transfer of frames to the PHY layer for transmission onto the wireless medium, and performs timestamp insertion (when necessary) and CRC generation for those frames.

7. The **Receiver (R) state machine** handles transfer, validation, and duplicate filtering of frames the PHY layer receives from the wireless medium, and updates the NAV using information from validly received frames.

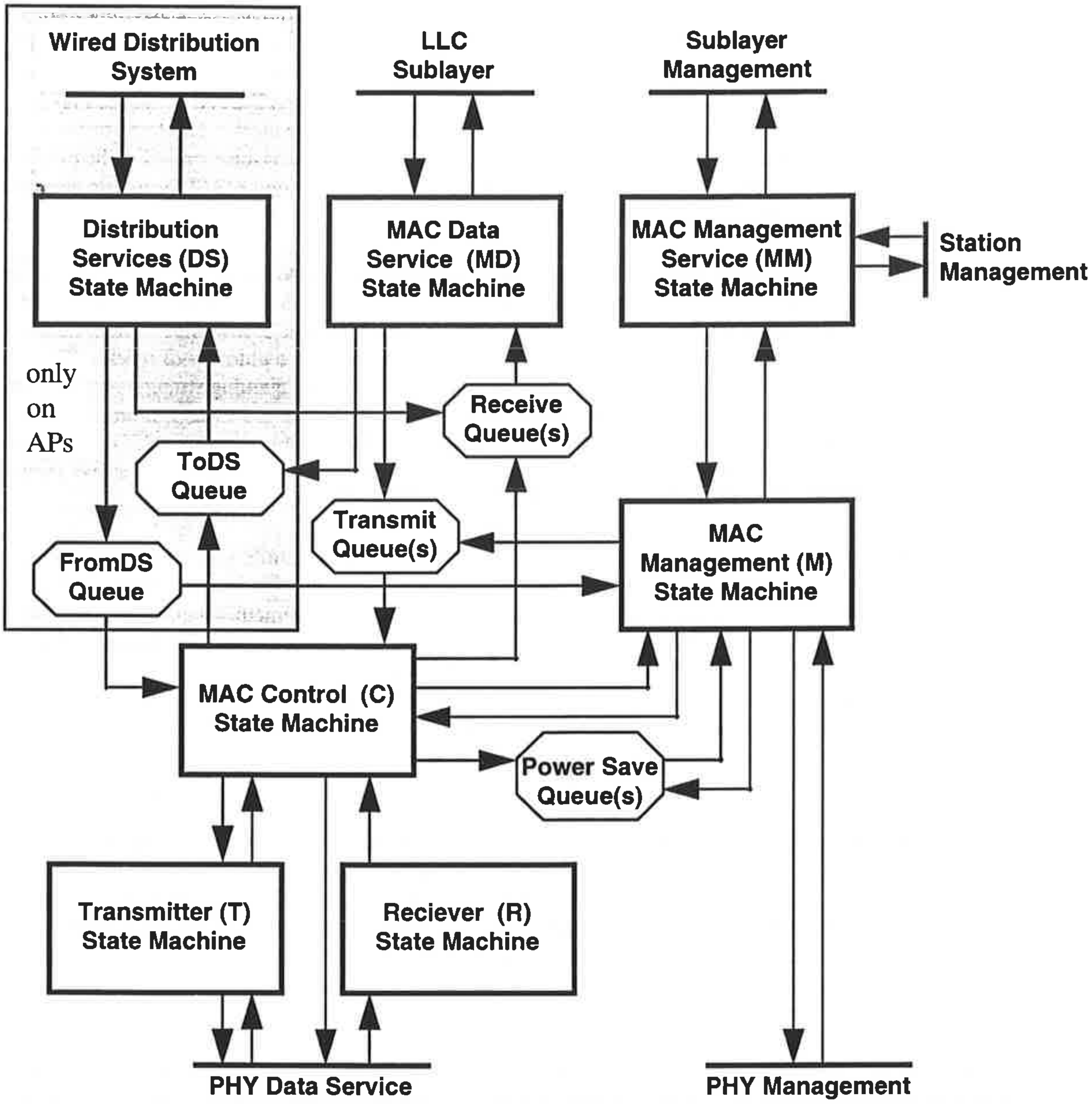
Wired Distribution System
LLC Sublayer
Sublayer Management
Distribution Services (DS) State Machine
MAC Data Service (MD) State Machine
MAC Management Service (MM) State Machine
Station Management
only on APs
Receive Queue(s)
ToDS Queue
Transmit Queue(s)
MAC Management (M) State Machine
FromDS Queue
MAC Control (C) State Machine
Power Save Queue(s)
Transmitter (T) State Machine
Reciever (R) State Machine
PHY Data Service
PHY Management

Figure 6-xx(1): MAC State Machine Overview.

6.7.1. Queues Used Between Certain State Machines

The five queues shown in figure 6–xx(1) are logical constructs that provide holding buffers for MSDUs within the MAC sublayer entity. Some of these queues must include physical storage, such for power save buffers at APs. Other queues are necessary for describing MAC operation in terms of independent, communicating state machines, but may not require physical storage in MAC implementations. These constructs are called "queues" because, in certain cases they need to hold more than one MSDU, while retaining the relative ordering among those MSDUs, and because the state machines are easier to read if the Enqueue() and Dequeue() functions are used for all explicit access to MSDU buffers within the MAC. However, calling these constructs queues does not require the MSDU buffers to be implemented as queues, nor to all use a uniform implementation of all buffers. Further details about the characteristics of each of these queues are presented below:

1. The **Transmit Queue(s) (TXq)** are a set of one or more queues which hold MSDUs awaiting transfer over the wireless medium. Entries may be placed onto these queues by MAC Data Service, for MSDUs received from the LLC sublayer entity, or by MAC Management, for MSDUs removed from the Power Save queues for immediate transmission. Entries are only removed from these queues by MAC Control. There may be a plurality of transmit queues to buffer frames separately for different service classes and/or priorities. The queuing strategy is not mandated by this standard, and these state machines assume only that the priority (and service class if more than one is ever supported) requested by the source of the MSDU is usable as a criterion for selecting buffers for removal from this queue.

2. The **Receive Queue(s) (RXq)** are a set of one or more queues which hold MSDUs addressed to the local LLC sublayer entity. . Entries may be placed onto these queues by MAC Control, for MSDUs received from the wireless medium, or by Distribution Services, for MSDUs received from the distribution system media. Entries are only removed from these queues by MAC Data Service. There may be a plurality of receive queues to buffer frames separately for different service classes and/or priorities. The queuing strategy is not mandated by this standard, and these state machines assume only that the priority (and service class if more than one is ever supported) indicated in the MSDU is available for indication to the LLC sublayer when MSDUs are removed from this queue. The reason that Distribution Services may put MSDUs directly onto the Receive queues is to avoid an AP–specific "own address" test for MSDUs removed from the FromDS queue.

3. The **Power Save Queue(s) (PSq)** are a set of one or more queues which hold MSDUs diverted from the transmit path by MAC Control due to the power save state of the TA and/or RA station. Entries are placed onto this queue by MAC Control and removed from this queue by MAC Management. Depending upon the situation, MAC Management by place MSDUs removed from a PSq onto the transmit queue, or may discard those MSDUs (such as for MSDUs buffered for a power save station which has Disassociated). There may be a plurality of receive queues at APs to buffer frames separately for different stations and/or for contention and contention–free power save operation. The queuing strategy is not mandated by this standard, and these state machines assume only that the station addresses and aging information for each buffered MSDU are usable as selection criteria for removal of entries from this queue.

4. The **ToDS Queue (ToDSq)** is a single queue which holds MSDUs received from the wireless medium with the ToDS frame control bit set (enqueued by MAC Control), as well as MSDUs received from the local LLC sublayer entity at the AP with a multicast DA or a individual DA for a station associated with a different BSS of the ESS (enqueued by MAC Data Service). MSDUs are removed from the ToDS queue by Distribution Services. The reason that MAC Data Service may put MSDUs directly onto the ToDS queue is to avoid needing an AP–specific "ToDS" test for MSDUs removed from the Transmit queue.

5. The **FromDS Queue (FrDSq)** is a single queue which holds MSDUs processed by Distribution Services and awaiting transmission on the wireless medium with the FromDS frame control bit set (also the ToDS frame control bit in the case of frames queued for transmission on a wireless distribution system). These MSDUs are enqueued solely by distribution services and dequeued solely by MAC Control.

The each of these queues has a corresponding flag which is true (=1) if there is at least one MSDU in the queue and is false (=0) if the queue is empty. These flags have names of the form "F_xxx" where the xxx is the name of the queue.

6.7.2. General Note on the MAC State Machines

6.7.2.1. Operational Assumptions

All state machines operate continuously and simultaneously. State transitions require zero time, and occur as soon as the specified conditions are met. If multiple exit conditions are true upon entry to a state, or become true simultaneously while in a state, the transition to be taken shall be the one with the transition bar that originates closest to the top of the vertical bar for that state (unless otherwise specified in the notes for the state). State transitions which return to the same state ("transitions to self") are taken once per occurrence of the specified conditions.

State transitions are not synchronized to any clock or timebase reference except as explicitly listed in the transition conditions. Within the actions listed for a given state transition, the actions are treated as occurring sequentially, so that the new values of variables modified by earlier actions can be used as operands in later actions. However, as viewed from other state machines, all global variables, flags, counters, and queues modified by the actions listed for a given state transition occur simultaneously. This simultaneity is vital in cases where the actions of a single state transition modify the values of variables used in more than one of the transitions from the current state of one of the other state machines.

At initialization of a MAC sublayer entity, each of the state machines starts in state 0. If a particular state machine is independently resetable, that reset is a forced return to state 0.

6.7.2.2. Graphical Representation of State Machines

States are indicated by vertical bars that are labeled above the bar. The state labels are a descriptive title and a state number. Each state number has a prefix letter to identify the state machine. This is illustrated in figure 6-xx(2).

Transitions are indicated by horizontal bars that terminate in an arrowhead. A loop transition that returns to the state it leaves includes a vertical section as part of the transition bar, as do a few inter–state transitions. The conditions that must be met in order to make a given transition are listed above the transition bar. The actions that are taken when a particular transition is made are listed below the transition bar. Transitions are labeled with a string enclosed in brackets. This string includes the letter that identifies the state machine, and two numbers, separated by a colon. These numbers are the state numbers of the originating state and the terminating state of the transition. For example, "[C12:4]" is the transition from state C12 to state C4 in the control state machine. If there is more than one transition between the same pair of states, a lower case letter is appended to those transition labels to render each label unique. For example, "[R2:0a]" and "[R2:0b]" are two, distinct transitions from state R2 to state R0. If it is necessary for a transition bar to cross a vertical bar without connection, a small gap appears in the horizontal bar adjacent to the vertical bar, as is shown in transition [X2:0] of figure 6–xx(2).

In addition to actions taken on transitions, actions may also be taken as part of a state. If this is the case, the actions to be taken in the state are specified in the notes on the particular state machine.

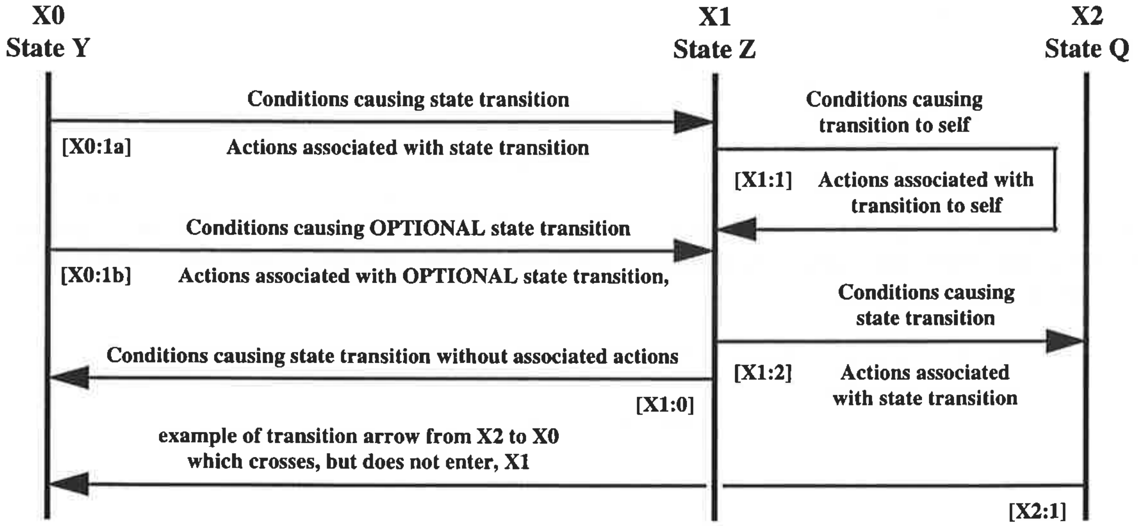
X0
State Y
X1
State Z
X2
State Q
Conditions causing state transition
[X0:1a]
Actions associated with state transition
Conditions causing transition to self
[X1:1]
Actions associated with transition to self
Conditions causing OPTIONAL state transition
[X0:1b]
Actions associated with OPTIONAL state transition,
Conditions causing state transition
[X1:2]
Actions associated with state transition
Conditions causing state transition without associated actions
[X1:0]
example of transition arrow from X2 to X0 which crosses, but does not enter, X1
[X2:1]

Figure 6-xx(2): MAC State Machine Notation

Actions of a given type (e.g. update the NAV) are either associated with state transitions, and are specified below the state transition arrow for the relevant transitions, or are associated with states, and specified in the notes for the relevant states, but not both. All timed activities are based on counters which trigger or enable state transitions, rather than being implicit side effects of the time interval expiring.

6.7.2.3. Notation Conventions

Local variable names begin with the letter(s), in lower case, that identify the state machine to which they are local.

Vector index values are enclosed in brackets "[index]" after the name of the vector variable.

Enumerated sets are enclosed in braces "{ set }" and the "=" operator used to test for membership in such a set.

Flag names begin with **F_** and flag values may be either true (=1) or false (=0). Flag operations are:

Set( )	which sets the flag value to true (=1)
Clear( )	which sets the value to false (=0)

Counter names begin with **T_** , the associated limit variable (if any) begins with **L_** , and the timeout event flag (if needed) begins with **F_** . Timeout flags are set when the counter equals the limit (if counting up) or equals zero (if counting down). Counters with specified limit values continue counting from zero on the clock cycle after reaching their limit values. Counter control functions are:

Start(T_xxx)	starts the counter from its current value
Start(T_xxx, value)	starts the counter with the specified value
Stop( )	stops the counter at its current value
Reset( )	stops the counter and sets the counter value =0
Restart( )	sets the counter value =0 and starts the counter

Operators include:

&	logical AND
|	logical OR
^	logical exclusive OR
!	logical not (monadic)
+	addition
–	subtraction
*	multiplication
/	division
=	equality comparison
:=	direct assignment (left operand value replaces initial value of right operand)
+=	incrementing assignment (left operand value added to initial value of right operand)

Arithmetic operations use signed (2's complement), integer arithmetic unless specified otherwise in the notes for a specific state or transition. Arithmetic precision shall be sufficient to accommodate the allowable ranges of values for all operands in each expression. For entities defined elsewhere in this standard, these value ranges are as specified with the relevant definition. For entities specific to these state machines, 16–bit precision (value range –32768:+32767) shall be assumed unless otherwise specified.

Numeric constants are in decimal unless preceded by **0b** to designate binary or **0h** to designate hexadecimal.

6.7.2.4. Functions Used in State Machine Definitions

Certain specialized operations are represented as functions references in state machine conditions or actions. This is done to improve readability of the state machine diagrams, avoiding the need to list complex operation sequences within the diagram as well as the reducing the number of pages needed for the diagrams themselves. These functions are defined below, listed in alphabetical order.

AccessState(macAddr)

Returns the information known at the local station about the station addressed by the argument for use by MAC Data service (this function is not used by Distribution Services):

0 is returned if the supplied address is a not an individual address,

1 indicates the addressee is associated with this BSS (also returned for the address of this station),
2 indicates the addressee is associated with another BSS of this ESS, and
–1 indicates that nothing is known about the association state of the station.

A non–AP station that is willing to send all frames via distribution services at the AP may implement a this function to always return –1.

Assemble(MSDU, MPDU, length)

Performs the MSDU payload reassembly process on duplicate–filtered or multicast MPDUs. The MSDU onto which the MAC payload of the provided MPDU is to be placed is determined from the MAC header of the MPDU, and the assembled result is available in the MSDU argument when this function assembles a final,. or only, fragment. This MSDU is assumed to be known to this MAC entity because of the required duplicate filtering action prior to the assembly action. The length argument is supplied as the MPDU length, and returned as the current MSDU length after this assembly operation (which may be the same, as for unfragmented MSDUs). This function returns a value of 0 if the MSDU being assembled is not yet complete, or a value of 1 if the MPDU just processed was the last (or only) MPDU of this MSDU.

ChangeNAV(duration)

Changes the value of the NAV to the duration, in microseconds, supplied in the argument. This function is the only means (other than the counting down of the NAV duration) by which the NAV value is ever decreased. This function is used solely to reduce the NAV value at the end of a contention free period.

CRC(accumulator, value)

updates the value of the 4–octet variable in the first argument, based on the 1–octet value in the second argument, using the CRC–32 polynomial specified in section 4.1.2.7.

Decrypt(MPDU, length)

Decrypts the payload of the supplied MPDU, using the station's WEP key (and the IV value at the beginning of the MPDU payload), and removes the IV and ICV fields from the MPDU. The initial MPDU length (in octets) is supplied in the length argument, and is decremented by 8 so the length value upon return reflects the octets removed for the IV and ICV. The WEP bit in the frame control field of the MPDU is cleared by this function. This function returns a value of true (=1) if the ICV check upon decryption was successful, or a value of false (=0) if the ICV check was unsuccessful.

Dequeue(MSDU, queue)

Removes the first MSDU from the specified queue and places this MSDU into the supplied argument.

DupFilter(MPDU)

Returns true (=1) if the sender address, sequence number, and fragment number of the MPDU operand are valid (either the first/sole fragment of a new MSDU, the next sequential fragment of an MSDU being received, or the MPDU contains a control frame); or false (=0) if the MPDU is a duplicate (must also have Retry =1 in the frame control field) or otherwise out of sequence. When a valid MPDU is processed, the relevant address, sequence control, and aging information about the related MSDU is recorded (for first/sole fragments) or updated (for intermediate/final fragments). The amount of storage for duplicate filtering tuples is assumed to, at least, meet the minimum requirements defined in section 6. A mechanism for selective replacement of filtering tuples is necessary, but is not defined herein.

Duration(length {, "A"} {, "R"} {, "M"})

Returns the duration value in microseconds (suitable for use in the Duration/ID field of an MPDU) of an MPDU with the specified length. This duration includes time for an SIFS plus an acknowledgment if the "A" argument is present, includes time for an RTS/CTS exchange if the "R" argument is present, and includes time for an SIFS plus a frame of length aMax_Frame_Length if the "M" argument is present.

Encrypt(MPDU, length)

Encrypts the payload of the supplied MPDU, using the station's WEP key (and an IV value generated within, or obtained by, this function), and inserts the IV and ICV fields into the MPDU, before and after the MPDU payload. The initial MPDU length (in octets) is supplied in the length argument, and is incremented by 8 so the length value upon return includes the octets added for the IV and ICV. The WEP bit in the frame control field of the MPDU is set by this function. From the point of view of these state machines, the IV update strategy (which is not specified in this standard) is fully encapsulated in this function.

Enqueue(MSDU, queue {, "head"})
Adds the supplied MSDU to the specified queue. The position at which the MSDU is added to the queue is unspecified (but is typically the tail of the queue) unless the optional "head" designation is present to indicate that the MSDU is to be placed at the head of the specified queue.

Fragment(MSDU, MPDU, FragNum, FragLength, FragLast)
Fragments the supplied MSDU, returning the fragment designated by FragNum in MPDU and the length of this fragment in FragLength. The MPDU includes a MAC header with appropriate Duration value for the Data/ACK exchange. The FragLast argument is set =1 if FragNum is the sole or last fragment. If the fragmentation rules in use at this station do not require fragmenting the MSDU, this function may return MPDU as an unfragmented copy of MSDU, and FragLast set =1, when called with FragNum =0.

GenACK(MPDU) or GenACK(macAddr, duration)
The first form generates and returns an MPDU containing an ACK control frame with a destination address equal to the source address of the MPDU (presumably a management or data frame) supplied as an argument and a duration value appropriate (for the active PHY) for the duration of the MPDU supplied as an argument minus the duration of the ACK frame and 1 SIFS interval. When the Duration/ID field of the supplied MPDU contains a value >32767, a value of 32768 is used in the Duration field of the ACK frame. The second form generates and returns an MPDU containing an ACK control frame with the specified destination address and duration value.

GenCFend({ack})
Generates and returns an MPDU containing a CF–End control frame, or a CF–End+Ack frame if the optional argument is present and !=0.

GenCFpoll(macAddr {, ack})
Generates and returns an MPDU containing a CF–Poll (no data) control frame addressed to the specified station. If the optional second argument is present and !=0, the frame subtype is CF–Poll+Ack.

GenCTS(MPDU)
Generates and returns an MPDU containing a CTS control frame with a destination address equal to the source address field of the MPDU (presumably an RTS control frame) supplied as an argument and a duration value appropriate (for the active PHY) for the duration of the MPDU supplied as an argument minus the duration of the CTS frame and 1 SIFS interval.

GenRTS(MPDU)
Generates and returns an MPDU containing an RTS control frame with the source and destination addresses equal to the Address1 and Address 2 fields of the MAC header of the MPDU (presumably a management or data frame) supplied as an argument, and the duration value appropriate (for the active PHY) for the duration of the MPDU supplied as an argument, plus the duration of the CTS frame, ACK frame, and 3 SIFS intervals.

Int(value)
Returns the integer part of the argument value.

Length(MxDU)
Returns the length, in octets, of the frame (MSDU or MPDU) stored in the buffer designated by the argument.

Min(value1, value2)
Returns the arithmetically smaller of the two argument values as real numbers (e.g. Min(–3, –2) = –3).

PollList(macAddr) or PollList("next")

If called with a MAC address argument, returns the value 2 (0b0010) if the specified station is on the contention free polling list and appropriate to be polled at this time, or the value 0 if either the addressed station is not on the polling list or should not be polled at this time. If called with an argument of "next," returns the MAC address of the next station to poll, or 0 if there are no more stations to poll during this CFP.

PsMode(macAddr)

Returns the power save mode (0=active mode, 1=power save or unknown mode) of the designated station, as known at the local station. At an AP, this function shall return 0 for a station using power save mode between the time a PS–Poll is received from that station and the time an another frame is received from that station indicating a resumption of power save mode. A non–AP station that is willing to send all frames via distribution services at the AP may implement a this function to always return 1.

Random()

Returns a random fractional value >0 and <=1.

Respond(FrameType, macAddr)

Generates an MPDU for a control frame of the specified frame type (generally CTS or ACK), with the specified destination address, and sends this MPDU to the transmitter state machine, using TxRequest(), at the end of an SIFS interval (determined as T_IFS=aSIFS).

ServiceState(macAddr, FrameType)

Tests whether the frames of the type specified may be validly sent by the designated station under the authentication & association state currently existing between this station and the station designated by the address argument. This function returns true (=1) if the frame type is permitted and false (=0) if the frame type is not permitted.

SetServiceState(macAddr, state)

Creates or updates the service state recorded for the designated station, based on the state argument. The same encoding of state values is as defined in section 2.5.

Tally()

Updates one or more MIB counters. The items being tallied and the counters being updated are listed in the notes for transitions that use this function.

ThisBSS(MPDU)

Returns true (=1) if the MAC header of the supplied MPDU contains a frame with any subtype of data or management types, or a CF–End or CF–End+Ack control frame, from a sender within the current BSS; and returns false (=0) otherwise. For data and management frames, the BSSID test equation is:

((FromDS=0) & (Address3=aCurrent_BSS_ID)) | ((FromDS=1) & (Address2=aCurrent_BSS_ID))

UpdNAV(duration)

updates the NAV based on the argument value (in microseconds). The NAV update operation can increase, but can never reduce, the current NAV value. The NAV update equation is:

```
IF (NAV < duration)  THEN  (NAV := duration)  ENDIF
```

6.7.2.5. Global Variables Used in State Machine Definitions

All MIB elements are treated as being global variables. In addition, the global variables defined below are used within the state machines (listed in alphabetical order):

F_AP	=1 if this station is operating as an AP (whether or not attached to a distribution system medium)
F_Awake	=1 if this station is currently active for transfers on the wireless medium
F_BCN	=1 when T_BCN wraps around from 0 to L_BCN

F_CFack =1 when the receiver state machine detects a frame with any CF–Ack subtype from this BSS

F_CFaware =1 if this station is CF–Aware

F_CFavail =1 if this station is CF–Aware and either is associated with a BSS with a polling PC or F_PC =1

F_CFP =1 if a contention free period is in progress in this BSS (stations with F_PC = 1), or =1 if a contention free period is believed to be in progress in this BSS (stations with F_PC = 0)

F_CFpoll =1 when a CF–Aware station receives a directed frame with any CF–Poll subtype from this BSS

F_DTIM =1 at the nominal time that a beacon containing a DTIM is to be transmitted, set concurrently with F_BCN and cleared no earlier than the processing of the beacon frame at power saving stations.

F_Mbusy =1 when the receiver state machine believes the medium to be busy (physical carrier sense only)

F_PC =1 if this station is operating as both an AP and a point coordinator

F_TxDone =1 when the transmitter state machine is idle

F_WEP =1 if the privacy function is active at this station

L_BCN the limit value for T_BCN (actually a reload value because T_BCN counts down). This variable is usually set to aBeacon_Interval, but may be set to (aBeacon_Interval * aDTIM_Count) at power saving stations that only wake up to receive beacons which contain DTIMs.

NAV current NAV value for zero/non–zero testing. The NAV counts down (by microseconds), and can only be modified using UpdNAV() or ChangeNAV().

SeqNum the value used for the sequence number field in the MAC headers of frames generated by this station.

TxSt Records the status of the MSDU transfer in progress within the MAC Control state machine, and is accessible from other state machines (particularly MAC Management). The value of TxSt is 0 when no transfer is in progress, =1 when a DCF transfer is in progress, =2 when a DCF transfer needs to execute a backoff, and =–1 when a CF–Poll response is in progress, =–2 when a CF–Poll response needs to be retired (awaiting another CF–Poll), =–3 when the point coordinator is in control of transmissions at an AP, and =–4 when the point coordinator is ending a contention free period.

T_Backoff the number of slot times remaining in the current coordination function backoff. This counter counts down by 1 each time T_Slot wraps around and the medium is non–busy (!F_Mbusy & (NAV=0)), and stops (setting F_Backoff) upon reaching zero.

T_BCN the number of microseconds remaining in the current beacon interval (stations with F_AP = 1), or the number of microseconds until the next TBTT (stations with F_AP = 0). This counter counts down, and wraps around from zero to the value of L_BCN.

T_Busy the number of microseconds that the medium has been indicated to be busy (CCA) by the PHY

T_CFDR the number of microseconds remaining in the current contention free period (stations with F_PC = 1), or the maximum number of microseconds remaining in the current contention free period (stations with F_PC = 0). This counter counts down, and stops upon reaching zero.

T_Dwell the number of microseconds remaining in the current dwell (FH PHY) or other medium occupancy limit (if any). This counter operates continuously when an FH PHY is enabled, counts down, and wraps around from zero to aDwell_Time. When using a PHY without dwell boundaries, T_Dwell is a constant with a value greater than the duration of a maximum–length MPDU with RTS/CTS/ACK.

T_IFS the number of microseconds since the end of the last reception, used to generate inter–frame spaces

T_Resp the number of microseconds since the transmitter state machine started on a frame requiring a response

T_Slot the number of microseconds remaining in the current slot. This counter operates continuously when the PHY is enabled, counts down, and wraps around from zero to aSlot_Time.

T_TSF the 8–octet counter, incremented every microsecond, used by the time synchronization function

6.7.3. MAC Data Service (MD) State Machine

The MAC Data Service state machine provides the MAC data service interface to the LLC sublayer. The MAC Data Service state transition diagram appears in Figure 6–xx(3).

6.7.3.1. Local Variables in the MAC Data Service State Machine

mdAddr1 the contents of the Address1 (recipient address) field of mdMSDU

mdAddr2 the contents of the Address2 (acknowledgement address) field of mdMSDU

mdAddr3 the contents of the Address3 (source address or BSSID) field of mdMSDU

mdDA	the destination address passed to/from LLC
mdData	the data vector passed to/from LLC
mdFromDS	the contents of the FromDS bit in the frame control field of mdMSDU
mdMSDU	a vector used to hold the outgoing MSDU generated from the LLC request or the incoming MSDU whose receipt is being indicated to LLC
mdPri	the transfer priority requested by LLC
mdRoute	the source routing information provided by LLC
mdSC	the service class requested by LLC
mdToDS	the contents of the ToDS bit in the frame control field of mdMSDU
mdType	the contents of the Frame Type and Subtype fields of the frame control field of mdMSDU

6.7.3.2. MAC Data Service State Machine Definition

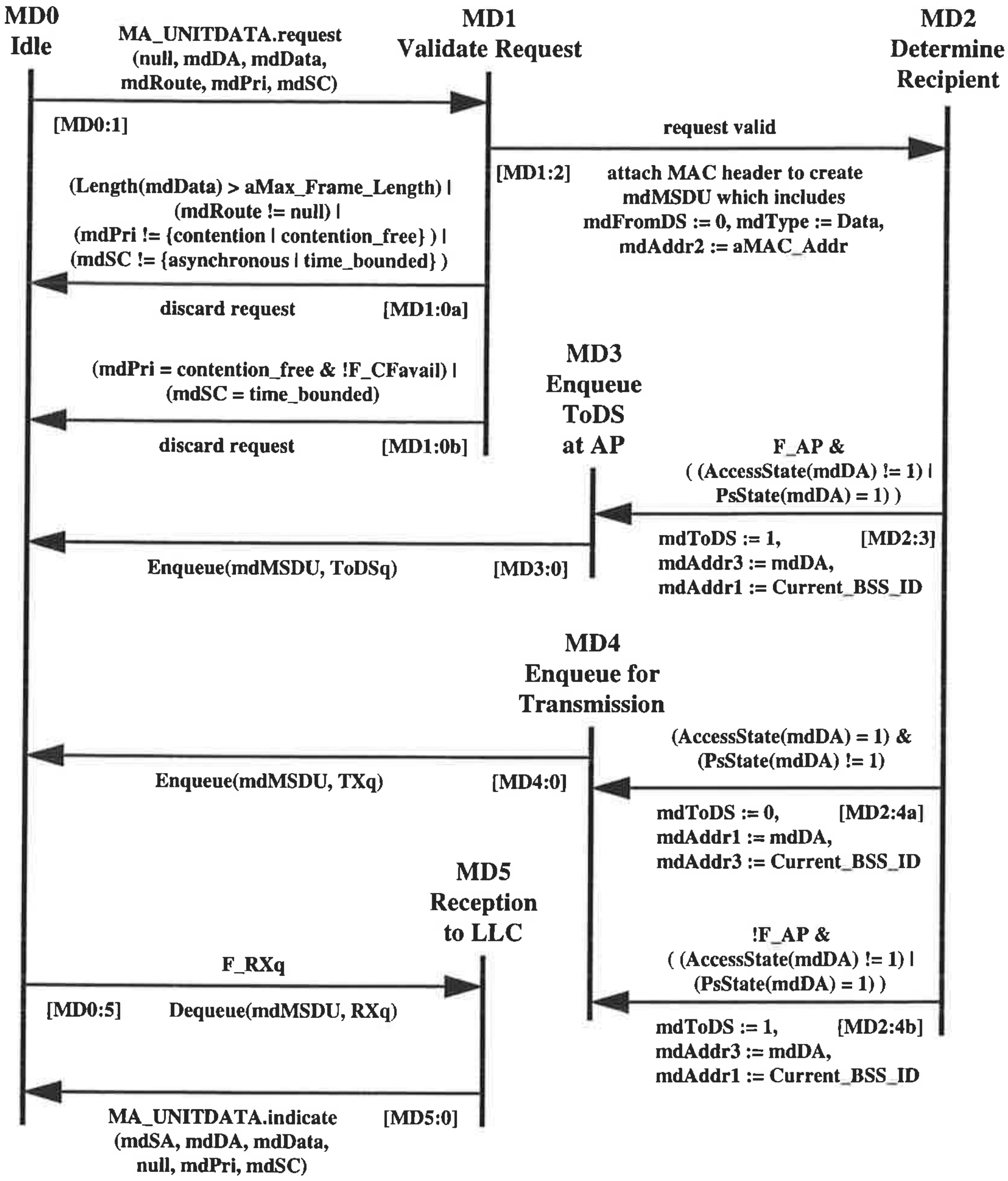

Figure 6-xx(3): MAC Data Service State Machine

6.7.3.3. Notes to the MAC Data Service State Machine

State MD0, Idle: MAC data service enters this state upon initialization and completion of transfers to or from LLC.

MD0:1, Transmit request from LLC: When a MA_UNITDATA.request from LLC is received, this transition is taken to validate the request before forwarding the request for transmission.

MD0:5, Reception buffered in receive queue: When a non–empty receive queue is detected, this transition is taken to dequeue the MSDU and indicate the reception to LLC..

State MD1, Validate Request: In this state the parameters of the MA_UNITDATA.request are checked to ensure that the requested transfer is within the capabilities of the MAC.

MD1:0a, Invalid parameter values: If the MA_UNITDATA.request specifies an excessive data length, non–null source routing information, an unavailable priority, or an unavailable service class, this transition back to Idle state is taken to discard the MSDU without transmission.

MD1:0b, Unsupported services requested (optional): The specification of contention free priority, when no point coordinator is available, and/or the specification of time bounded service class (the support for which is currently undefined) cannot be handled, although they are legal values in the MA_UNITDATA.request. This transition returns to idle state to discard these requests. This transition is distinct from transition MD1:0a because these services are requested by valid parameter values and, at least in the case of contention free priority, the service may be available at certain times.

MD1:2, Request valid: Whenever MAC data service is in state MD1 and conditions do not enable any of the invalid or unsupported service transitions, the MA_UNITDATA.request is valid and this transition is taken to attach the MAC header and default header field contents to the MSDU and to determine where to enqueue the MSDU. The source address (Address2 field contents) is always set to the MAC address of this station, overriding the source address provided by LLC.

State MD2, Determine Recipient: In this state the proper queue is selected for transmission of this MSDU.

MD2:3, Recipient requires delivery via distribution (at AP): If the MSDU is addressed to anything except a non–power–save station associated with this AP, this transition is taken to mark the frame to be delivered by Distribution Services (ToDS=1) and to enter state MD3 to enqueue this MSDU on the ToDS queue.

MD2:4a, Recipient active in same BSS: If the MSDU is addressed to a non–power managed station associated with the current BSS, this transition is taken to mark the frame for direct delivery and enter state MD4 to enqueue this frame on the transmit queue.

MD2:4b, Recipient requires delivery via distribution (non–AP): If the MSDU is addressed to anything except a non–power–save station in the same BSS, this transition is taken to mark the frame to be delivered by Distribution Services (ToDS=1) at the AP, and to enter state MD4 to send the MSDU for transmission.

State MD3, Enqueue ToDS at AP: Once the fields in the MAC header have been filled in by MAC data service at the AP, this transition is taken to send the MSDU to Distribution Services.

MD3:0, Add to end of ToDS queue: Enqueuing an MSDU from the local LLC entity at the AP directly onto the ToDS queue is equivalent to transmitting the MSDU from an associated station to the AP with the ToDS bit set in the frame control field, and is done for the same reasons — communicating with a station attached elsewhere on the distribution system, communicating with a station that uses power save mode, and distributing multicasts to all relevant stations.

State MD4, Enqueue for Transmission: Once the fields in the MAC header have been filled by MAC data service, this transition is taken to send the MSDU on the wireless medium.

MD4:0, Add to end of Transmit queue: When the MAC header has been prepared, either for direct transfer to another station in the BSS or a transfer to distribution services at the AP, this transition is taken to enqueue the MSDU for transmission.

State MD5, Reception to LLC: In this state MSDUs received from the wireless medium are reported to LLC.

MD5:0, Indicate reception to LLC: When the received MSDU payload has been extracted from the MAC framing, this transition is taken to generate MA_UNITDATA.indicate to the local LLC sublayer entity.

6.7.3.4. Known Limitations of the MAC Data Service State Machine

The MAC data service interface has no MA_UNITDATA.confirmation primitive, so there is no means by which to inform the LLC sublayer that an MSDU has been discarded due to excessive length, unavailable service class, unsupported priority, or non–null route information.

The lack of a defined mechanism for time bounded frame delivery means that MAC data service discards frames that request the time bounded service class, even though this service class is defined as available in the service specification.

6.7.4. MAC Management Service (MM) State Machine

The MAC Management Service state machine provides the MAC management service interface to the adjacent sublayer management entity and station management entity. The MAC Management Service state transition diagram appears in Figure 6–xx(4).

6.7.4.1. Local Variables Used in the MAC Management Service State Machine

<< to be supplied when further definition is available for the MAC management service interface >>

6.7.4.2. MAC Management Service State Machine Definition

<< to be supplied when further definition is available for the MAC management service interface >>

<< placeholder for diagram >>

Figure 6-xx(4): MAC Management Service State Machine

6.7.4.3. Notes to the MAC Management Service State Machine

<< to be supplied when further definition is available for the MAC management service interface >>

6.7.4.4. Known Limitations of the MAC Management Service State Machine

<< the relevant service interface was not defined when this submission was written >>

6.7.5. Distribution Services (DS) State Machine

The Distribution Services state machine exists only at access points, where it provides distribution services and interfaces to both wired and wireless distribution systems. This state machine operates at all APs, even those which are attached to neither wired nor wireless distribution system media. The Distribution Services state transition diagram appearsin figure 6–xx(5).

All MSDUs processed by distribution services may be assumed to have 802.11 MAC headers, because the only entities which enqueue MSDUs on the ToDS queue are MAC data service, which attaches a MAC header before enqueuing the MSDU, and the receive function of MAC control, which would have discarded a reception long before detecting ToDS=1 if there was not a valid MAC header. Since MSDUs reach the distribution system through other instances of distribution services at APs, or through equivalent functionality at portals, all MSDUs received from the distribution system will also have MAC headers.

6.7.5.1. Local Variables Used in the Distribution Services State Machine

The only local variable explicitly used in the Distribution Services state machine is "dsMSDU," which holds the MSDU being processed for distribution. There are implicit references to the source address, destination address, sender address, receiver address, and ToDS/FromDS frame control field bits.

6.7.5.2. Distribution Services State Machine Definition

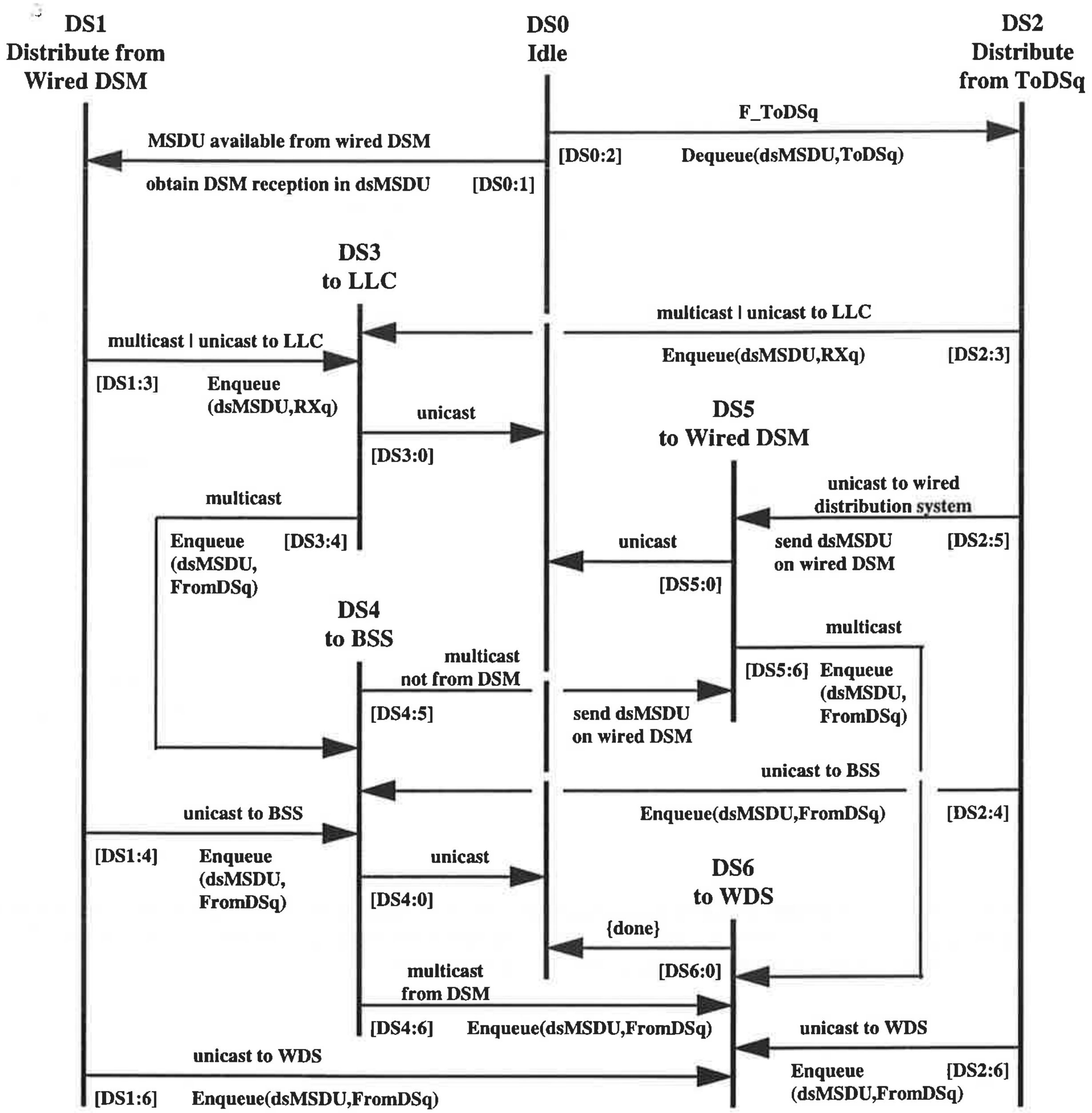

Figure 6-xx(5): Distribution Services State Machine

6.7.5.3. Notes to the Distribution Services State Machine

State DS0, Idle: Distribution services enters this state upon initialization or after processing of an MSDU is concluded. There is no "sleep" state in this state machine because distribution services exists only at APs, which do not sleep.

DS0:1, Obtain MSDU from (wired) DSM: When an MSDU is available from the interface to the wired distribution system medium, this transition is taken to obtain that MSDU for distribution processing.

DS0:2, Obtain MSDU from ToDS Queue: When an MSDU is available in the ToDS queue, this transition is taken to obtain that MSDU for distribution processing.

State DS1, Distribute from (wired) DSM: In this state the address information in dsMSDU, and related association information recorded at the AP are analyzed to determine where dsMSDU needs to be sent. The Address1 field in the MAC header of dsMSDU is updated to designate the appropriate destination(s), the Address2 field is updated to identify this AP (BSSID), and the FromDS bit is set in the frame control field.

DS1:3, Unicast to LLC or Multicast: If dsMSDU is addressed to the individual address of the local LLC entity at the AP, or if dsMSDU is addressed to any type of group address (in this state machine, "multicast" should be understood to mean both multicast and broadcast addresses), this transition is taken to enqueue dsMSDU on the receive queue where MAC data service can inform the local LLC entity, and to enter DS3 state where further distribution of multicasts can take place.

DS1:4, Unicast to BSS: If dsMSDU is addressed to an individual address of a station associated with this AP, this transition is taken to enqueue dsMSDU on the FromDS queue for transfer on the wireless medium.

DS1:6, Unicast to WDS: If dsMSDU is addressed to an individual address of a station associated with another BSS of this ESS that must be reached via wireless distribution system, this transition is taken to build a WDS header (add Address4 field, set both ToDS and FromDS in the frame control field) and to enqueue dsMSDU for transfer as a WDS frame on the wireless medium.

State DS2, Distribute from ToDS Queue: In this state the address information in dsMSDU, and related association information recorded at the AP are analyzed to determine where dsMSDU needs to be sent. The Address1 field in the MAC header of dsMSDU is updated to designate the appropriate destination(s), the Address2 field is updated to identify this AP (BSSID), and the FromDS bit is set in the frame control field.

DS2:3, Unicast to LLC or Multicast: If dsMSDU is addressed to the individual address of the local LLC entity at the AP, or if dsMSDU is addressed to any type of group address, this transition is taken to enqueue dsMSDU on the receive queue where MAC data service can inform the local LLC entity, and to enter state DS3 where further distribution of multicasts can occur.

DS2:4, Unicast to BSS: If dsMSDU is addressed to an individual address of a station associated with this AP, this transition is taken to enqueue dsMSDU on the FromDS queue for transfer on the wireless medium.

DS2:5, Unicast to DSM: If dsMSDU is addressed to an individual address of a station associated with another BSS of this ESS, or otherwise accessible via the wired distribution system medium, this transition is taken to send dsMSDU to the wired distribution system medium interface.

DS2:6, Unicast to WDS: If dsMSDU is addressed to an individual address of a station associated with another BSS of this ESS that must be reached via wireless distribution system, this transition is taken to build a WDS header (add Address4 field, set both ToDS and FromDS in the frame control field) and to enqueue dsMSDU for transfer as a WDS frame on the wireless medium.

State DS3, Distribute to LLC: In this state the address information in dsMSDU is checked. If the destination of dsMSDU is an individual address, distribution processing for this MSDU is complete. If the destination address is a multicast, the MAC header information is updated and another copy of dsMSDU is sent to stations in the BSS.

DS3:0, Unicast (to LLC) completed: If the destination of dsMSDU is an individual address, distribution of this MSDU is complete and this transition is taken to return to Idle state.

DS3:4, Multicast not from BSS: If the destination of dsMSDU is a multicast address, this transition is taken to enqueue this MSDU for transfer to stations in the BSS via the wireless medium. Multicasts are sent to the stations in the BSS even if the source is a station in the BSS to allow all associated stations, including those using power save mode, to have a chance to receive the multicast.

State DS4, Distribute to BSS: In this state the address information in dsMSDU is checked. If the destination of dsMSDU is an individual address, distribution processing for this MSDU is complete. If the destination address is a multicast, the source address is checked. If the source is not a station accessed via the wired distribution system medium, the MAC header information is updated and another copy of dsMSDU is sent to the wired DSM (if any). If the destination address is a station accessed via the wired DSM, the MAC header information is updated and another copy of dsMSDU is sent to the wireless distribution system (if any).

DS4:0, Unicast (to BSS) completed: If the destination of dsMSDU is an individual address, distribution of this MSDU is complete and this transition is taken to return to Idle state.

DS4:5, Multicast not from (wired) DSM: If the destination of dsMSDU is a multicast address, and the source address is not a station accessed via the wired DSM, this is taken to send dsMSDU to the wired distribution system medium interface.

DS4:6, Multicast from (wired) DSM: If the destination of dsMSDU is a multicast address, and the source address is a station accessed via the wired DSM, this transition is taken to build a WDS header (add Address4 field, set both ToDS and FromDS in the frame control field) and to enqueue dsMSDU for transfer as a WDS frame on the wireless medium.

State DS5, Distribute to (wired) DSM: In this state the address information in dsMSDU is checked. If the destination of dsMSDU is an individual address, distribution processing for this MSDU is complete. If the destination address is a multicast, the MAC header information is updated and another copy of dsMSDU is sent to the wireless distribution system (if any). If there is not a wired DSM active at this AP, operations which send MSDUs to the wired DSM interface are null, but the state transitions still take place so that multicast distribution works properly.

DS5:0, Unicast (to BSS) completed: If the destination of dsMSDU is an individual address, distribution of this MSDU is complete and this transition is taken to return to Idle state.

DS5:6, Multicast: If the destination of dsMSDU is a multicast address, this transition is taken to build a WDS header (add Address4 field, set both ToDS and FromDS in the frame control field) and to enqueue dsMSDU for transfer as a WDS frame on the wireless medium. This transition is not conditional on the source of the multicast because there may be multiple WDS destinations, and the multicast needs to reach all of them.

State DS6, Distribute to WDS: In this state the dsMSDU is sent to the wireless distribution system. If there is not a wireless distribution system active at this AP, operations which enqueue MSDUs for the wireless distribution system are null, and this state causes immediate drop through to the Idle state..

DS6:0, (distribution completed): After enqueuing dsMSDU for transfer on the wireless distribution system, distribution of this MSDU is complete and this transition is taken to return to Idle state.

6.7.5.4. Known Limitations of the Distribution Services State Machine

There is a high probability that the wired DSM will have a considerably higher data rate than the wireless medium, and MSDUs might accumulate from the wired DSM faster than they can be delivered over the wireless medium. This state machine does not address the potential need for selectively discarding MSDUs from the wired DSM when a large delivery backlog threatens to exhaust buffer space at the AP.

A wireless distribution system may connect a plurality of APs. If so, a mechanism, not defined herein, may be needed to prevent "loops" in the wireless distribution of multicasts among these APs.

6.7.6. MAC Control (C) State Machine

The MAC Control state machine provides the distributed coordination function and (optionally) the point coordination function for transfer of frames over the wireless medium. This state machine also provides fragmentation, reassembly, and part of power management (detection of frames addressed to stations that might not be awake). The MAC Control state transition diagram appears in figures 6–xx(6) through 6–xx(8). Figure 6–xx(6) contains the top level of this state machine and the states used for reception control. Figure 6–xx(7) contains the states used for DCF transmission control and CF–Poll responses. Figure 6–xx(8) contains the optional states for operation of a point coordinator at an AP.

6.7.6.1. Local Variables Used in the MAC Control State Machine

cAddr1, crAddr1 refers to the contents of the Addr1 field of cMPDU or crMPDU

cAddr2, crAddr2 refers to the contents of the Addr2 field of cMPDU or crMPDU

cCFnext =1 when the point coordinator is ready for the next outgoing MSDU to be dequeued (into cMSDU) and fragmented (into cMPDU)

cCW current contention window value

crDir =1 when the last frame provided by receiver state machine (crMPDU) was addressed to the individual address of this station

cFragLast, crFragLast contents of the Last fields of the frame control fields of cMPDU or crMPDU, also used within the state machine to indicate the final fragment being handled

cFragLng the length of the current transmit fragment (cMPDU)

cFragNum the fragment number of the current transmit fragment (cMPDU)

cLength, crLength the length of cMPDU or crMPDU

cMore refers to the contents of the More field of the frame control field of cMPDU

cMPDU, crMPDU vectors used to store MPDUs being transmitted (cMPDU) or received (crMPDU)

cMSDU, crMSDU vectors used to store MSDUs being fragmented (cMSDU) or assembled (crMSDU)

cRetry contents of the Retry fields of the frame control fields of cMPDU or crMPDU

cRetryCnt the number of retries for the current transmission attempt (cMPDU)

cToDS contents of the ToDS fields of the frame control fields of cMPDU or crMPDU

crTs the timestamp value returned by the receiver state machine with crMPDU

cType, crType contents of Type and Subtype fields of frame control fields of cMPDU or crMPDU

6.7.6.2. MAC Control State Machine Definition

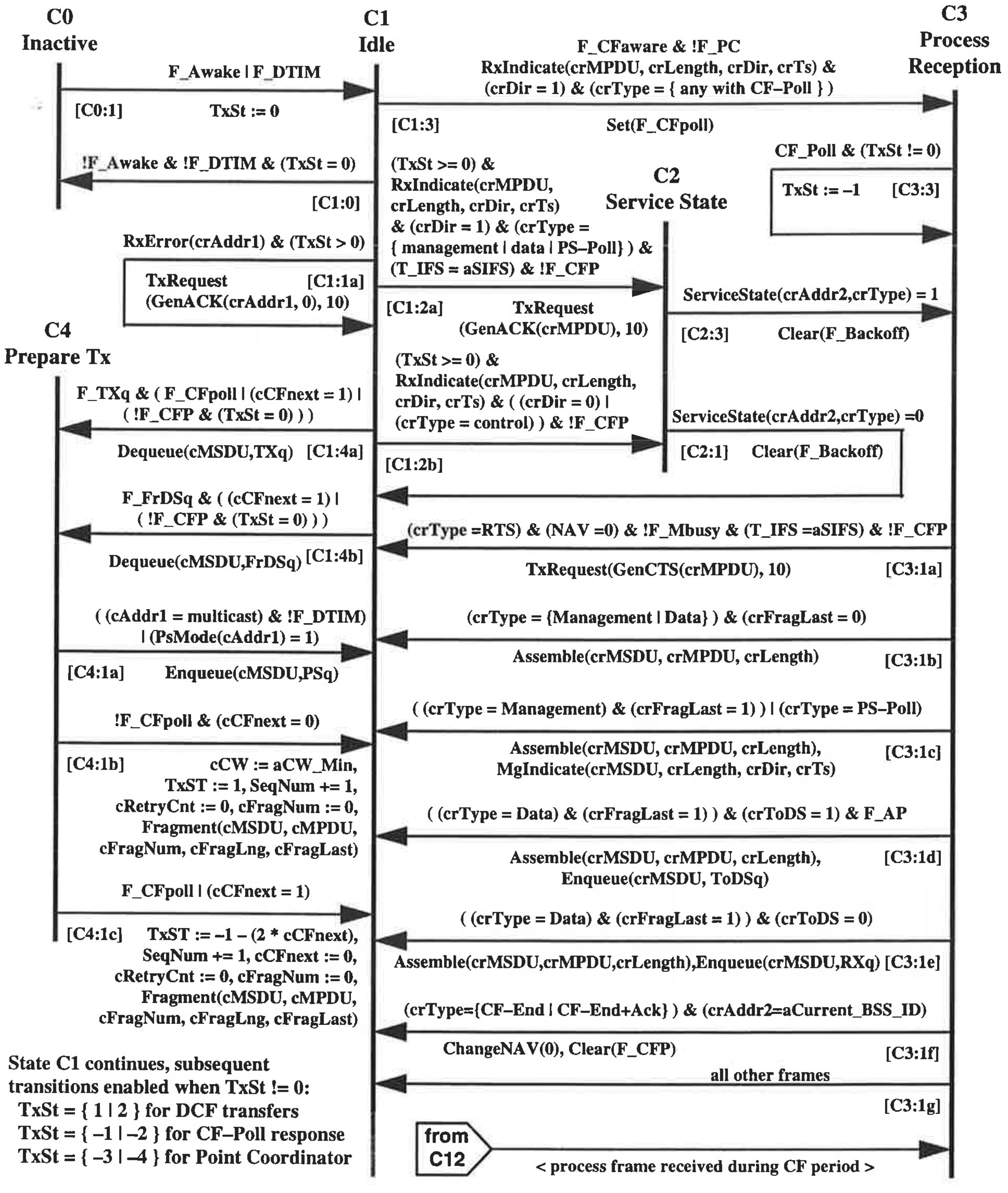
C0
Inactive
C1
Idle
C2
Service State
C3
Process
Reception
C4
Prepare Tx
F_Awake | F_DTIM
[C0:1]
TxSt := 0
!F_Awake & !F_DTIM & (TxSt = 0)
[C1:0]
RxError(crAddr1) & (TxSt > 0)
TxRequest
[C1:1a]
(GenACK(crAddr1, 0), 10)
F_CFaware & !F_PC
RxIndicate(crMPDU, crLength, crDir, crTs) &
(crDir = 1) & (crType = { any with CF–Poll })
[C1:3]
Set(F_CFpoll)
CF_Poll & (TxSt != 0)
TxSt := –1
[C3:3]
(TxSt >= 0) &
RxIndicate(crMPDU,
crLength, crDir, crTs)
& (crDir = 1) & (crType =
{ management | data | PS–Poll}) &
(T_IFS = aSIFS) & !F_CFP
[C1:2a]
TxRequest
(GenACK(crMPDU), 10)
ServiceState(crAddr2,crType) = 1
[C2:3]
Clear(F_Backoff)
(TxSt >= 0) &
RxIndicate(crMPDU, crLength,
crDir, crTs) & ((crDir = 0) |
(crType = control)) & !F_CFP
[C1:2b]
ServiceState(crAddr2,crType) =0
[C2:1]
Clear(F_Backoff)
F_TXq & (F_CFpoll | (cCFnext = 1) |
(!F_CFP & (TxSt = 0)))
Dequeue(cMSDU,TXq)
[C1:4a]
F_FrDSq & ((cCFnext = 1) |
(!F_CFP & (TxSt = 0)))
Dequeue(cMSDU,FrDSq)
[C1:4b]
(crType =RTS) & (NAV =0) & !F_Mbusy & (T_IFS =aSIFS) & !F_CFP
TxRequest(GenCTS(crMPDU), 10)
[C3:1a]
((cAddr1 = multicast) & !F_DTIM)
| (PsMode(cAddr1) = 1)
[C4:1a]
Enqueue(cMSDU,PSq)
(crType = {Management | Data}) & (crFragLast = 0)
Assemble(crMSDU, crMPDU, crLength)
[C3:1b]
!F_CFpoll & (cCFnext = 0)
[C4:1b]
cCW := aCW_Min,
TxST := 1, SeqNum += 1,
cRetryCnt := 0, cFragNum := 0,
Fragment(cMSDU, cMPDU,
cFragNum, cFragLng, cFragLast)
((crType = Management) & (crFragLast = 1)) | (crType = PS–Poll)
Assemble(crMSDU, crMPDU, crLength),
MgIndicate(crMSDU, crLength, crDir, crTs)
[C3:1c]
((crType = Data) & (crFragLast = 1)) & (crToDS = 1) & F_AP
Assemble(crMSDU, crMPDU, crLength),
Enqueue(crMSDU, ToDSq)
[C3:1d]
F_CFpoll | (cCFnext = 1)
[C4:1c]
TxST := –1 – (2 * cCFnext),
SeqNum += 1, cCFnext := 0,
cRetryCnt := 0, cFragNum := 0,
Fragment(cMSDU, cMPDU,
cFragNum, cFragLng, cFragLast)
((crType = Data) & (crFragLast = 1)) & (crToDS = 0)
Assemble(crMSDU,crMPDU,crLength),Enqueue(crMSDU,RXq) [C3:1e]
(crType={CF–End | CF–End+Ack}) & (crAddr2=aCurrent_BSS_ID)
ChangeNAV(0), Clear(F_CFP)
[C3:1f]
all other frames
[C3:1g]
State C1 continues, subsequent
transitions enabled when TxSt != 0:
TxSt = { 1 | 2 } for DCF transfers
TxSt = { –1 | –2 } for CF–Poll response
TxSt = { –3 | –4 } for Point Coordinator
from
C12
< process frame received during CF period >

Figure 6-xx(6): MAC Control State Machine, Top Level & Reception Control

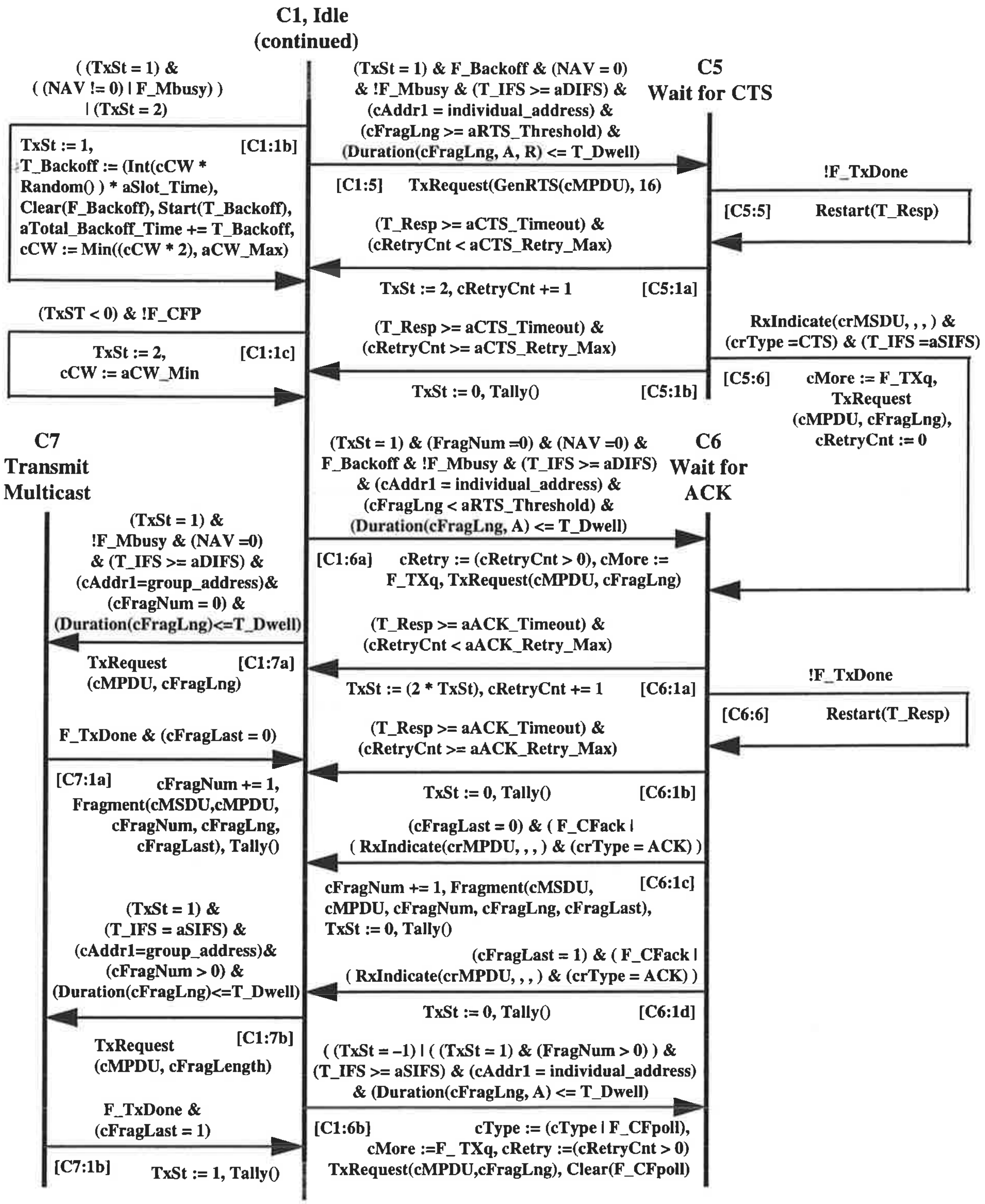
C1, Idle
(continued)
((TxSt = 1) &
((NAV != 0) | F_Mbusy))
| (TxSt = 2)
TxSt := 1, [C1:1b]
T_Backoff := (Int(cCW *
Random()) * aSlot_Time),
Clear(F_Backoff), Start(T_Backoff),
aTotal_Backoff_Time += T_Backoff,
cCW := Min((cCW * 2), aCW_Max)
(TxST < 0) & !F_CFP
TxSt := 2, [C1:1c]
cCW := aCW_Min
(TxSt = 1) & F_Backoff & (NAV = 0)
& !F_Mbusy & (T_IFS >= aDIFS) &
(cAddr1 = individual_address) &
(cFragLng >= aRTS_Threshold) &
(Duration(cFragLng, A, R) <= T_Dwell)
[C1:5] TxRequest(GenRTS(cMPDU), 16)
C5
Wait for CTS
!F_TxDone
[C5:5] Restart(T_Resp)
(T_Resp >= aCTS_Timeout) &
(cRetryCnt < aCTS_Retry_Max)
TxSt := 2, cRetryCnt += 1 [C5:1a]
(T_Resp >= aCTS_Timeout) &
(cRetryCnt >= aCTS_Retry_Max)
TxSt := 0, Tally() [C5:1b]
RxIndicate(crMSDU, , ,) &
(crType =CTS) & (T_IFS =aSIFS)
[C5:6] cMore := F_TXq,
TxRequest
(cMPDU, cFragLng),
cRetryCnt := 0
C7
Transmit
Multicast
(TxSt = 1) & (FragNum =0) & (NAV =0) &
F_Backoff & !F_Mbusy & (T_IFS >= aDIFS)
& (cAddr1 = individual_address) &
(cFragLng < aRTS_Threshold) &
(Duration(cFragLng, A) <= T_Dwell)
C6
Wait for
ACK
[C1:6a] cRetry := (cRetryCnt > 0), cMore :=
F_TXq, TxRequest(cMPDU, cFragLng)
(TxSt = 1) &
!F_Mbusy & (NAV =0)
& (T_IFS >= aDIFS) &
(cAddr1=group_address)&
(cFragNum = 0) &
(Duration(cFragLng)<=T_Dwell)
TxRequest [C1:7a]
(cMPDU, cFragLng)
(T_Resp >= aACK_Timeout) &
(cRetryCnt < aACK_Retry_Max)
TxSt := (2 * TxSt), cRetryCnt += 1 [C6:1a]
!F_TxDone
[C6:6] Restart(T_Resp)
F_TxDone & (cFragLast = 0)
(T_Resp >= aACK_Timeout) &
(cRetryCnt >= aACK_Retry_Max)
[C7:1a] cFragNum += 1,
Fragment(cMSDU,cMPDU,
cFragNum, cFragLng,
cFragLast), Tally()
TxSt := 0, Tally() [C6:1b]
(cFragLast = 0) & (F_CFack |
(RxIndicate(crMPDU, , ,) & (crType = ACK))
cFragNum += 1, Fragment(cMSDU, [C6:1c]
cMPDU, cFragNum, cFragLng, cFragLast),
TxSt := 0, Tally()
(TxSt = 1) &
(T_IFS = aSIFS) &
(cAddr1=group_address)&
(cFragNum > 0) &
(Duration(cFragLng)<=T_Dwell)
(cFragLast = 1) & (F_CFack |
(RxIndicate(crMPDU, , ,) & (crType = ACK))
TxSt := 0, Tally() [C6:1d]
TxRequest [C1:7b]
(cMPDU, cFragLength)
((TxSt = –1) | ((TxSt = 1) & (FragNum > 0)) &
(T_IFS >= aSIFS) & (cAddr1 = individual_address)
& (Duration(cFragLng, A) <= T_Dwell)
F_TxDone &
(cFragLast = 1)
[C1:6b] cType := (cType | F_CFpoll),
cMore :=F_ TXq, cRetry :=(cRetryCnt > 0)
TxRequest(cMPDU,cFragLng), Clear(F_CFpoll)
[C7:1b] TxSt := 1, Tally()

Figure 6-xx(7): MAC Control State Machine, Transmission Control

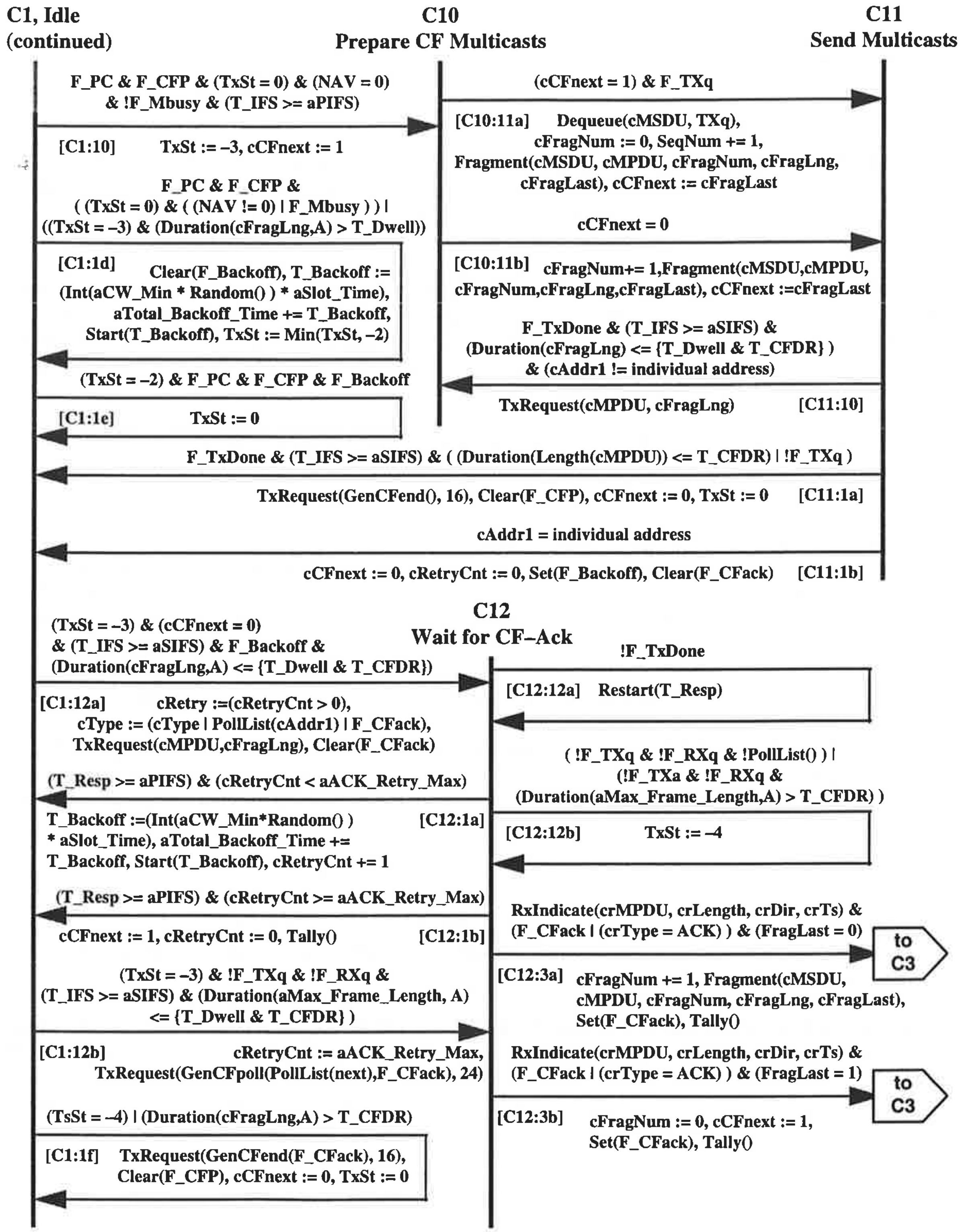
C1, Idle
(continued)
C10
Prepare CF Multicasts
C11
Send Multicasts
F_PC & F_CFP & (TxSt = 0) & (NAV = 0)
& !F_Mbusy & (T_IFS >= aPIFS)
[C1:10] TxSt := –3, cCFnext := 1
F_PC & F_CFP &
((TxSt = 0) & ((NAV != 0) | F_Mbusy)) |
((TxSt = –3) & (Duration(cFragLng,A) > T_Dwell))
[C1:1d] Clear(F_Backoff), T_Backoff :=
(Int(aCW_Min * Random()) * aSlot_Time),
aTotal_Backoff_Time += T_Backoff,
Start(T_Backoff), TxSt := Min(TxSt, –2)
(TxSt = –2) & F_PC & F_CFP & F_Backoff
[C1:1e] TxSt := 0
(cCFnext = 1) & F_TXq
[C10:11a] Dequeue(cMSDU, TXq),
cFragNum := 0, SeqNum += 1,
Fragment(cMSDU, cMPDU, cFragNum, cFragLng,
cFragLast), cCFnext := cFragLast
cCFnext = 0
[C10:11b] cFragNum+= 1,Fragment(cMSDU,cMPDU,
cFragNum,cFragLng,cFragLast), cCFnext :=cFragLast
F_TxDone & (T_IFS >= aSIFS) &
(Duration(cFragLng) <= {T_Dwell & T_CFDR})
& (cAddr1 != individual address)
TxRequest(cMPDU, cFragLng) [C11:10]
F_TxDone & (T_IFS >= aSIFS) & ((Duration(Length(cMPDU)) <= T_CFDR) | !F_TXq)
TxRequest(GenCFend(), 16), Clear(F_CFP), cCFnext := 0, TxSt := 0 [C11:1a]
cAddr1 = individual address
cCFnext := 0, cRetryCnt := 0, Set(F_Backoff), Clear(F_CFack) [C11:1b]
C12
Wait for CF–Ack
(TxSt = –3) & (cCFnext = 0)
& (T_IFS >= aSIFS) & F_Backoff &
(Duration(cFragLng,A) <= {T_Dwell & T_CFDR})
[C1:12a] cRetry :=(cRetryCnt > 0),
cType := (cType | PollList(cAddr1) | F_CFack),
TxRequest(cMPDU,cFragLng), Clear(F_CFack)
(T_Resp >= aPIFS) & (cRetryCnt < aACK_Retry_Max)
T_Backoff :=(Int(aCW_Min*Random()) [C12:1a]
* aSlot_Time), aTotal_Backoff_Time +=
T_Backoff, Start(T_Backoff), cRetryCnt += 1
(T_Resp >= aPIFS) & (cRetryCnt >= aACK_Retry_Max)
cCFnext := 1, cRetryCnt := 0, Tally() [C12:1b]
(TxSt = –3) & !F_TXq & !F_RXq &
(T_IFS >= aSIFS) & (Duration(aMax_Frame_Length, A)
<= {T_Dwell & T_CFDR})
[C1:12b] cRetryCnt := aACK_Retry_Max,
TxRequest(GenCFpoll(PollList(next),F_CFack), 24)
(TsSt = –4) | (Duration(cFragLng,A) > T_CFDR)
[C1:1f] TxRequest(GenCFend(F_CFack), 16),
Clear(F_CFP), cCFnext := 0, TxSt := 0
!F_TxDone
[C12:12a] Restart(T_Resp)
(!F_TXq & !F_RXq & !PollList()) |
(!F_TXa & !F_RXq &
(Duration(aMax_Frame_Length,A) > T_CFDR))
[C12:12b] TxSt := –4
RxIndicate(crMPDU, crLength, crDir, crTs) &
(F_CFack | (crType = ACK)) & (FragLast = 0)
to
C3
[C12:3a] cFragNum += 1, Fragment(cMSDU,
cMPDU, cFragNum, cFragLng, cFragLast),
Set(F_CFack), Tally()
RxIndicate(crMPDU, crLength, crDir, crTs) &
(F_CFack | (crType = ACK)) & (FragLast = 1)
to
C3
[C12:3b] cFragNum := 0, cCFnext := 1,
Set(F_CFack), Tally()

Figure 6-xx(8): MAC Control State Machine, Point Coordinator

6.7.6.3. Notes to the MAC Control State Machine

State C0, Inactive: The MAC controller shall enter this state upon initialization or when idle and told to "sleep" by the clearing of F_Awake. If told to sleep during transfer activities the MAC control state machine completes the transmission or reception in progress before entering inactive state.

C0:1, Wakeup: When F_Awake is set or the start of the next beacon interval (TBTT) for a beacon with DTIM occurs while the receiver is inactive, this transition is taken to prepare for possible reception activities. TxSt is set to zero to ensure that transfers are not initiated until appropriate sequences for DCF for PCF have occurred since this wakeup.

State C1, Idle: The MAC controller enters this state upon wakeup and between most transfer activities. By having this "global" idle state, with additional state information maintained in variable TxSt, many portions of the MAC controller, including MSDU dequeuing and fragmentation, and acknowledgment and processing of received MPDUs, can be shared between DCF and PCF, significantly reducing the number of states and transitions needed in this state machine. State C1 continues across all three of the state transition diagrams.

C1:0, Go to sleep: When the F_Awake flag is cleared, and there is not a DTIM expected nor in progress (!F_DTIM), and there is no DCF nor PCF activity unfinished (TxSt =0), an idle MAC controller takes this transition to become inactive.

C1:1a, Acknowledge erroneous directed frame: When the receiver state machine reports an erroneous directed MPDU (such as a duplicate reception, out–of–sequence fragment, or non–decryptable WEP frame) this transition is taken to acknowledge the erroneous frame without further processing of the reception. By acknowledging this frame the sending station does not waste time on the medium, nor delay other, potentially valid, frames while retrying this frame.

C1:1b, Backoff for DCF: This transition starts the backoff timer, and doubles the contention window value for use on the next backoff. This transition is taken whenever a DCF transmission is pending (TxSt=1) with the medium busy (NAV or CCA), as well as when a backoff is required prior to a retry of back–to–back transmission attempt (TxSt=2, which is returned to TxSt=1 by this transition). The transitions which initiate DCF transfers require the backoff timer to have expired (F_Backoff set) as well as the medium to be free. T_Backoff only decrements when the medium is free (!F_Mbusy & (NAV=0)).

C1:1c, Cleanup for CF–Poll response at end of CFP: If a frame has been partially transmitted in response to CF–Polls at the time the CF period ends, TxSt will be <0 when F_CFP is cleared. When this situation occurs, this transition is taken to allow the remainder of the MSDU to be delivered during the contention period. TxSt is set =2 so that a backoff is executed, because otherwise the DCF would attempt to send this non–initial fragment immediately.

C1:1d, Backoff for start of CFP or new dwell during CFP (PC only): If the nominal start of a CF period is delayed due to a medium bush condition this transition is taken to generate a backoff in the range 0..aCW_Min (this contention window does not increase). This reduces the risk of destructive collisions between overlapping point–coordinated BSSes on the same channel. In addition, at the beginning of each dwell period during the contention free period (or medium occupancy period for non–FH PHYs), this transition is taken to reduce the risk of collisions when the channel must be relinquished within a CFP. In the former case, this transition occurs when TxSt=0, and TxSt is set =–2 to enable transition C1:1e upon expiration of the backoff timer. In the later case this transition occurs when TsSt=–3, and the value of TxSt is not affected by this transition.

C1:1e, Cleanup for PCF backoff (PC only): To avoid interaction between DCF backoffs and the start of CF periods, transition C1:10 does not refer to the backoff timer among its enabling conditions. Therefore, of a backoff at the beginning of a CF period is necessary, this transition is taken upon expiration of the backoff timer (F_Backoff set) to change TxSt=2 (meaning PCF backoff in progress) to TxSt=0, re–enabling transition C1:10.

C1:1f, End CFP: When there are no further transmissions pending during a CF period (TxSt=–4), or there is insufficient time remaining in the CF period for the pending transmission, this transition is taken to generate and send the CF–End frame to end the CF period. If an acknowledgment is pending for a frame received at the PC (F_CFack set), a CF–End+Ack frame is generated.

C1:2a, Obtain & acknowledge directed reception: When the receiver state machine indicates receipt of a directed data frame, management frame, or PS–Poll frame (which should only occur at an AP), while operating under DCF control (TxSt>=0), this transition is taken to obtain the MPDU and generate an acknowledgment. This is done prior to checking services state (State C2) because the MPDU was received without error, so an acknowledgment is needed whether or not the frame is able to be processed at this station.

C1:2b, Obtain non–directed reception: When the receiver state machine indicates receipt of a directed data frame, management frame, or a control frame, while operating under DCF control (TxSt>=0), this transition is taken to obtain the MPDU without generating an acknowledgment.

C1:3, Recognize CF–Poll (CF–Aware only): When the receiver state machine at a CF–Aware station indicates receipt of any directed frame that includes a CF–Poll function (Data+CF–Poll, Data+CF–Poll+Ack, CF–Poll(no data), or CF–Poll+Ack(no data)), this transition is taken to obtain the MPDU and set F_CF–Poll to enable generation of a CF–Poll response. There is no need to traverse state C2 in this situation, because in order to be polled by a PC, this station must already have associated with the AP at which the PC is operating, so this station can be assumed to be in service state 3. There is no need to generate an ACK response before processing the reception because this CF–Aware station can acknowledge the reception as part of its response to the CF–Poll.

C1:4a, Dequeue from TXq: When the next outgoing frame is required, by the DCF (TxSt=0), PC (cCFnext=1), or station responding to a CF–Poll (F_CFpoll set), and the transmit queue is non–empty, this transition is taken to dequeue the MSDU at the head of the transmit queue and to enter state C4 to prepare this MSDU for transfer.

C1:4b, Dequeue from FrDSq (AP only): When the next outgoing frame is required, by the DCF (TxSt=0) or PC (cCFnext=1), and the transmit queue is empty but the FromDS queue is non–empty, this transition is taken to dequeue the MSDU at the head of the FromDS queue and to enter state C4 to prepare this MSDU for transfer. This transition is only required at APs, because there is not a FromDS queue at non–AP stations. The transmit queue is processed before the FromDS queue because the power–save processing in MAC management places the buffered multicasts (and directed frames indicated in the DTIM when a point coordinator is active) at the head of the transmit queue immediately after generating and enqueuing each beacon frame.

C1:5, Send RTS: When the DCF has a pending transmission of a directed Data or Management frame at least as long as aRTS_Threshold, and a free medium (CCA and NAV and backoff), this transition is taken to generate and send the RTS frame, and to enter state C5 to await the CTS response.

C1:6a, Send first fragment without RTS: When the DCF has a pending transmission of a directed Data or Management frame shorter than aRTS_Threshold, and a free medium (CCA and NAV and backoff), this transition is taken to send the Data or Management frame, and to enter state C6 to await the ACK response. The More frame control bit is set if there are additional MSDUs remaining on the TXq and the Retry frame control bit is set if this transition is a retry of an earlier, unacknowledged transmission attempt.

C1:6b, Send subsequent fragment or CF–Poll response: When the DCF has a pending transmission of a non–initial fragment of a directed Data or Management frame (TxSt=1), or a station responding to a CF–Poll has any fragment (TxSt=–1), this transition is taken to send the Data or Management frame, and to enter state C6 to await the ACK response.

C1:7a, Send first fragment of multicast: When the DCF has a pending transmission of the initial fragment of a broadcast or multicast Data or Management frame, this transition is taken to send the Data or Management frame.

C1:7b, Send subsequent fragment of multicast: When the DCF has a pending transmission of a non–initial fragment of a broadcast or multicast Data or Management frame, this transition is taken to send the Data or Management fragment.

C1:10, Start CFP (PC only): When the CF period is scheduled to begin, and the medium is free, this transition is taken to indicate the PC is in control of the transmission process (TxSt=3) and request a frame be dequeued and fragmented (cCFnext=1).

C1:12a, Send CF fragment (PC only): When the PC is active and a fragment is available for transmission (TxSt=3 & cCFnext=0), this transition is taken to send the fragment. The retry frame control bit is set if this transmission is a retry of the fragment, and the appropriate combination of CF–Poll and CF–Ack subtype bits are set in the frame subtype field.

C1:12b, Send CF–Poll (PC only): When the PC is active and no more MSDUs are queued for transmission, but there is still enough time in the CF period for an aMax_Frame_Length MPDU plus acknowledgment, this transition is taken to send a CF–Poll, possibly with CF–Ack, to the next station on the polling list.

State C2, Check Service State: In this state a received frame is validated for reception processing by checking whether the service state that exists between the sender and this station permits the sending of a frame of this class (1, 2, or 3). Frame types in higher classes than the service state are ignored by the recipient. The definitions in the security section state that these classes are what may be transmitted, but the enforcement must be at the receiving end because where security is an issue (such as the requirement that authentication occur before association), a station cannot be relied upon not to transmit frames which are of a higher class than the service state allows.

C2:1, Discard unauthorized frame: If the frame class is greater than the service state, the frame is discarded.

C2:3, Process authorized frame: If the frame class is less than or equal to the service state, the frame is processed.

State C3, Process Reception: In this state the valid receptions are decoded and processed.

C3:1a, Respond with CTS: If the reception was an RTS frame in contention period, and the medium is not busy at this station (CCA and NAV), this transition is taken to respond with a CTS frame after an SIFS interval.

C3:1b, Assemble non–final fragment: If the reception was a non–final fragment of a data or management frame, this transition is taken to reassemble the partially–complete frame.

C3:1c, Assemble final management fragment: If the reception was the final (or sole) fragment of a management frame, this transition is taken to assemble the completed frame and to indicate the reception and pass the management MSDU to the MAC management state machine.

C3:1d, Assemble final "ToDS" data fragment (AP only): If the reception was the final (or sole) fragment of a data frame with the ToDS frame control bit =1, this transition is taken to assemble the completed frame and enqueue the frame on the ToDSq.

C3:1e, Assemble final data fragment: If the reception was the final (or sole) fragment of a data frame with the ToDS frame control bit =0, this transition is taken to assemble the completed frame and enqueue the frame on the RXq.

C3:1f, Process CF–End: If the reception was a CF–End or CF–End+Ack, this transition is taken to reset the NAV and mark the CF period as over. There is no need for separate interpretation of the CF–End+Ack because the receiver state machine detects the CF–Ack indication and sets F_CFack prior to generating the RxIndicate.

C3:1g, Ignore all other received frames: If none of the other C3:1–transitions are taken, this transition is taken to ignore the received MPDU.

C3:3, CF–Poll with partial transmission (CF–Aware only): If a frame has been partially transmitted under DCF control at the time a CF–Aware station is polled during the CF period, TxSt will be !=0 when F_CFpoll is set. When this situation occurs, this transition is taken to allow the next MPDU of the MSDU to be delivered in the CF–Poll response by changing TxSt to =–1.

State C4, Prepare Transmission: In this state a queued MSDU is prepared for transmission, or transferred from the TXq or FrDSq to the PSq if dequeued at a time the destination (power saving) station is not likely to be awake.

C4:1a, Move frame to PSq: If the dequeued MSDU contains a recipient address of broadcast, multicast, or the individual address of a station not known to be in active mode, this transition is taken to place the MSDU onto the PSq. This MSDU may be returned to the TXq by MAC management at a time when multicasts are permitted (after a DTIM) or when the addressed station is known to be active.

C4:1b, Prepare frame for DCF transmission: If the MSDU is being prepared for transmission under DCF control, this transition is taken to generate the first fragment and to initialize for DCF transfer. This initialization includes setting cCW to the minimum contention window value (aCW_Min), clearing the retry counter and fragment number, incrementing the sequence number, and setting TxSt=1 to enable DCF transfer.

C4:1c, Prepare frame for PCF transmission: If the MSDU is being prepared for transmission under PCF control, this transition is taken to generate the first fragment and to initialize for PCF transfer. This initialization includes clearing the retry counter and fragment number, incrementing the sequence number, and setting TxSt=–1 for CF–Poll response (cCFnext=0) or TxSt=–3 for PC (cCFnext=1).

State C5, Wait for CTS: After sending an RTS frame, the DCF enters this state to await a response.

C5:1a, Set up to retry RTS: If the CTS response timeout expires without a valid response, when the retry limit has not yet been reached, this transition is taken to increment the retry count and return to state C1 with TxSt=2 to force a backoff, and subsequent retry of the RTS.

C5:1b, Abandon RTS retries: If the CTS response timeout expires without a valid response, when the retry limit has been reached, this transition is taken to cease attempting to send this MSDU and to update statistics counters including aCollision_Count, aMultiple_Collision_Count, and aFailed_Count.

C5:5, Start CTS response timer: When the RTS frame has been passed to the transmitter state machine, this transition is taken to reset and start the response timer.

C5:6, Start transfer on CTS response: If a CTS frame is received within the response timeout period, this transition is taken to send the first (or sole) fragment of the MSDU after an SIFS interval. The More bit in the frame control field of this fragment is set (and remains set for subsequent fragments) if the TXq is non–empty.

State C6, Wait for ACK: After sending a directed data, management, or PS–Poll frame under DCF control or in response to a CF–Poll, this state is entered to wait for an acknowledgment.

C6:1a, Set up to retry fragment: If the ACK response timeout expires without a valid response, when the retry limit has not yet been reached, this transition is taken to increment the retry count and return to state C1 with TxSt=2 to force a backoff, and subsequent retry of the frame transmission.

C6:1b, Abandon fragment retries: If the ACK response timeout expires without a valid response, when the retry limit has been reached, this transition is taken to cease attempting to send this MSDU and to update statistics counters including aCollision_Count, aMultiple_Collision_Count, and aFailed_Count.

C6:1c, Prepare next fragment on ACK response: If an ACK frame or CF–Ack indication is received within the response timeout period, and the frame begin acknowledged is not the last (or sole) fragment, this transition is taken to generate the next fragment of the MSDU and return to state C1 to allow the frame to be transferred.

C6:1d, Detect ACK response to final fragment: If an ACK frame or CF–Ack indication is received within the response timeout period, and the frame begin acknowledged is the last (or sole) fragment, this transition is taken to end DCF or CF–Poll response processing for this MSDU (TxSt=0).

C6:6, Start ACK response timer: When the fragment has been passed to the transmitter state machine, this transition is taken to reset and start the response timer.

State C7, Transmit Multicast: After sending a broadcast or multicast data or management frame under DCF control, this state is entered to process any remaining fragments of the MSDU. This state is not used when processing CF–Poll responses because multicast transfers from such a station are sent to the AP as directed frames with the ToDS bit set.

C7:1a, Prepare next multicast fragment: If the frame just sent is not the last (or sole) fragment of the MSDU, this transition is taken to generate the next fragment of the MSDU and return to state C1 to allow the fragment to be transferred.

C7:1b, Detect transmission of final multicast fragment: If the frame just sent is the last (or sole) fragment of the MSDU, this transition is taken to end DCF processing for this MSDU (TxSt=0).

States C8, C9: These states do not currently exist. Their numbers are reserved so that if additional DCF states are required, the point coordinator states, which start at state C10, will not require renumbering.

State C10, Prepare CF Multicasts (PC only): This state is where the beacon with DTIM that starts each CF period, and any queued multicast frames are prepared for transmission by the point coordinator. The first of these multicast frames will always be the beacon because MAC management places the beacon MSDU at the head of the TXq at the same time as F_CFP is set to start the CF period.

C10:11a, Dequeue multicast frame (PC only): When there are no fragments left from the previous MSDU, or when state C10 is newly entered at the beginning of free medium to start a CF period, this transition is taken to dequeue the next MSDU, increment the sequence number, generate the first (or sole) fragment, and enter state C11 to send this fragment. cCFnext is set to the last fragment indicator value to either enable transition C10:11b, if there are more fragments of this MSDU, or to reenable this transition (C10:11a) if the next transfer is of a new MSDU.

C10:11b, Prepare next multicast fragment (PC only): When there are fragments remaining in the current MSDU, this transition is taken to generate the next fragment, and enter state C11 to send this fragment. cCFnext is set to the last fragment indicator value to either enable transition C10:11a, if there are no more fragments of this MSDU, or to reenable this transition (C10:11b) if the next transfer is of another MPDU of the same MSDU.

State C11, Send CF Multicasts (PC only): This state is where the beacon and other multicast frames at the beginning of the CF period are transmitted, as a burst of traffic, ending when the queue of multicasts is exhausted or the available time in the CF period is exhausted.

C11:10, End of multicast fragment (PC only): When an SIFS interval has elapsed since the previous transmission, and there is sufficient time left in the CF period to send the pending MPDU, this transition is taken to pass the MPDU to the transmitter state machine and return to state C10 to prepare the next multicast. The duration test ensures that there is sufficient time in both the CF period and the current dwell, but other transitions from state C11 only deal with the case where there is insufficient time in the CF period. This is because the dwell timer wraps around, so if there is enough time in the CF period, this transition will ultimately be able to occur, even if delayed near the end of a dwell.

C11:1a, End of CFP reached during multicast transmissions (PC only): If the CF period ends while in state C11, this transition is taken to send the CF–End frame and shut down the point coordinator (cCFnext=TxSt=0). The CF period could end because the TXq becomes empty before an MSDU directed to an individual address is encountered, or because the time available in the CF period is exhausted.

C11:1b, End of available multicasts (PC only): When an MSDU directed to an individual address is dequeued (transition C10:11a), this transition is taken to commence CF transfers which require acknowledgments and may include CF–Polls. Initialization for this activity includes setting cCFnext=0, to prevent another dequeue by state C4 before the current MSDU is transmitted, clearing cRetryCnt and F_CFack, because the multicasts are unacknowledged, so these might not have the correct values upon leaving state C11, and setting F_Backoff to avoid an extra backoff at the start of the acknowledged portion of CF period transfers.

State C12, Wait for CF–Ack (PC only): After sending a directed data or management frame the point coordinator enters this state to wait for an acknowledgment.

C12:1a, Retry PC transmission (PC only): If a PIFS period elapses without a valid response, when the retry limit has not yet been reached, this transition is taken to increment the retry count, and initiate a backoff (between 0 and aCW_Min) and return to state C1 for retry of the frame transmission after the backoff.

C12:1b, Abandon PC retries (PC only): If a PIFS period elapses without a valid response, when the retry limit has been reached, this transition is taken to cease attempting to send this MSDU and to update statistics counters including aCollision_Count, aMultiple_Collision_Count, and aFailed_Count.

C12:3a, Process reception at PC when Tx fragments remain (PC only): If an ACK frame or CF–Ack indication is received within the PIFS period, and the frame begin acknowledged is not the last (or sole) fragment, this transition is taken to generate the next fragment of the MSDU and to enter state C3 to process the reception with which the acknowledgment was indicated. If the acknowledgment was an ACK frame, state C3 will return immediately to state C1 via transition C3:1g.

C12:3b, Process reception at PC when Tx at end of frame (PC only): If an ACK frame or CF–Ack indication is received within the PIFS period, and the frame begin acknowledged is the last (or sole) fragment, this transition is taken to set up to dequeue the next MSDU (cCFnext=1), and to enter state C3 to process the reception with which the acknowledgment was indicated. If the acknowledgment was an ACK frame, state C3 will return immediately to state C1 via transition C3:1g.

C12:12a, Start PC response timer (PC only): When the fragment has been passed to the transmitter state machine, this transition is taken to reset and start the response timer.

C12:12b, Detect impending end of CFP (PC only): When there is nothing further to do during this CF period because the TXq, FrDSq, and PollList are empty, or because both the TXq and FrDSq are empty and there is insufficient time remaining in the CF period for a frame of aMax_Frame_Length plus acknowledgment (hence the next station on the polling list cannot safely be polled), this transition is taken to initiate ending the CF period by setting TxSt=–4. The actual transmission of the CF–End frame and shutdown of the point coordinator until the next CF period occurs in transition C1:1f after state C1 is entered following the acknowledgement (or lack thereof) to the transmission which caused entry to state C12.

6.7.6.4. Known Limitations of the MAC Control State Machine

This control state machine, running at an AP, always acknowledges a PS–Poll and signals the management state machine to move frames for the station sending the Ps–Poll from the PSq to the TXq. This control state machine never responds to a PS–Poll with a buffered data frame after an SIFS interval (which is also a less safe response because the NAV setting at stations which detect the PS–Poll only protects an ACK response, not a Data response).

The control state machine does not explicitly support dynamic data rate switching by providing a data rate argument on transmit requests. However, to the extent that rate switching is transparent in PHY receivers that support the optional data rates, the control state machine implicitly supports dynamic rate switching. To use dynamic rate switching the data rate capabilities of stations need to be recorded along with access state, and the Fragment() function would utilize this information when building the MPDUs and calculating the durations with individual destination addresses of stations that can receive at the higher rate. This rate selection would be an implicit attribute of the MPDU transferred to the transmit state machine by TxRequest() so that the PHY–specific rate selection code could be put in the TXVECTOR.

Neither the control state machine nor the management state machine implements an aging mechanism for buffered frames which have remained undelivered for an excessive amount of time.

6.7.7. MAC Management (M) State Machine

The MAC Management state machine provides the MAC management functions of the station, including time synchronization, power management, authentication, association/reassociation, and scanning. This state machine also has the primary responsibility for maintaining the MAC management information base. The MAC Management state transition diagram appears in figure 6–xx(9).

6.7.7.1. Local Variables Used in the MAC Management State Machine

6.7.7.2. MAC Management State Machine Definition

Figure 6-xx(9): MAC Management State Machine, Station

Figure 6-xx(10): MAC Management State Machine, Access Point

6.7.7.3. Notes to the MAC Management State Machine

6.7.7.4. Known Limitations of the MAC Management State Machine

6.7.8. Transmitter (T) State Machine

The transmitter state machine is a simple "data pump" which transfers the MPDU octets to the PHY while calculating the CRC value, transfers the CRC value after the end of the MPDU payload is reached, then ends the transmission and returns to its idle state awaiting the next transmit request from the MAC control state machine. CRC generation must be done in the transmitter state machine in order to accommodate Beacon and Probe Response frames, where the contents of the timestamp field in the MPDU are not known until immediately prior to transmission. If WEP is active at this station, the transmitter state machine handles encryption of the MPDU payloads of outgoing data frames. The Transmitter state transition diagram appears in figure 6–xx(11).

6.7.8.1. Local Variables in the Transmitter State Machine

tCrc a 4–octet vector used to accumulate the CRC
tFrameType the contents of the Type and Subtype fields of the Frame Control field of tMPDU
tMPDU a vector containing the MPDU from the control state machine, which includes the MAC header and payload, but not the CRC field
tLength the number of octets in the MPDU provided by the control state machine
tOctet counts octets transferred to PHY and indexes tMPDU vector to select next octet for transfer
tPLCPlength the value used for the length element in the TxVector, which is the value of tLength plus 4 to account for the addition of the CRC–32 field
tTs an 8–octet vector used to capture the value of T_TSF for use generating the Timestamp field

tOffset is a PHY–dependent constant, not a local variable. See note for transition T2:3.

6.7.8.2. Transmitter State Machine Definition

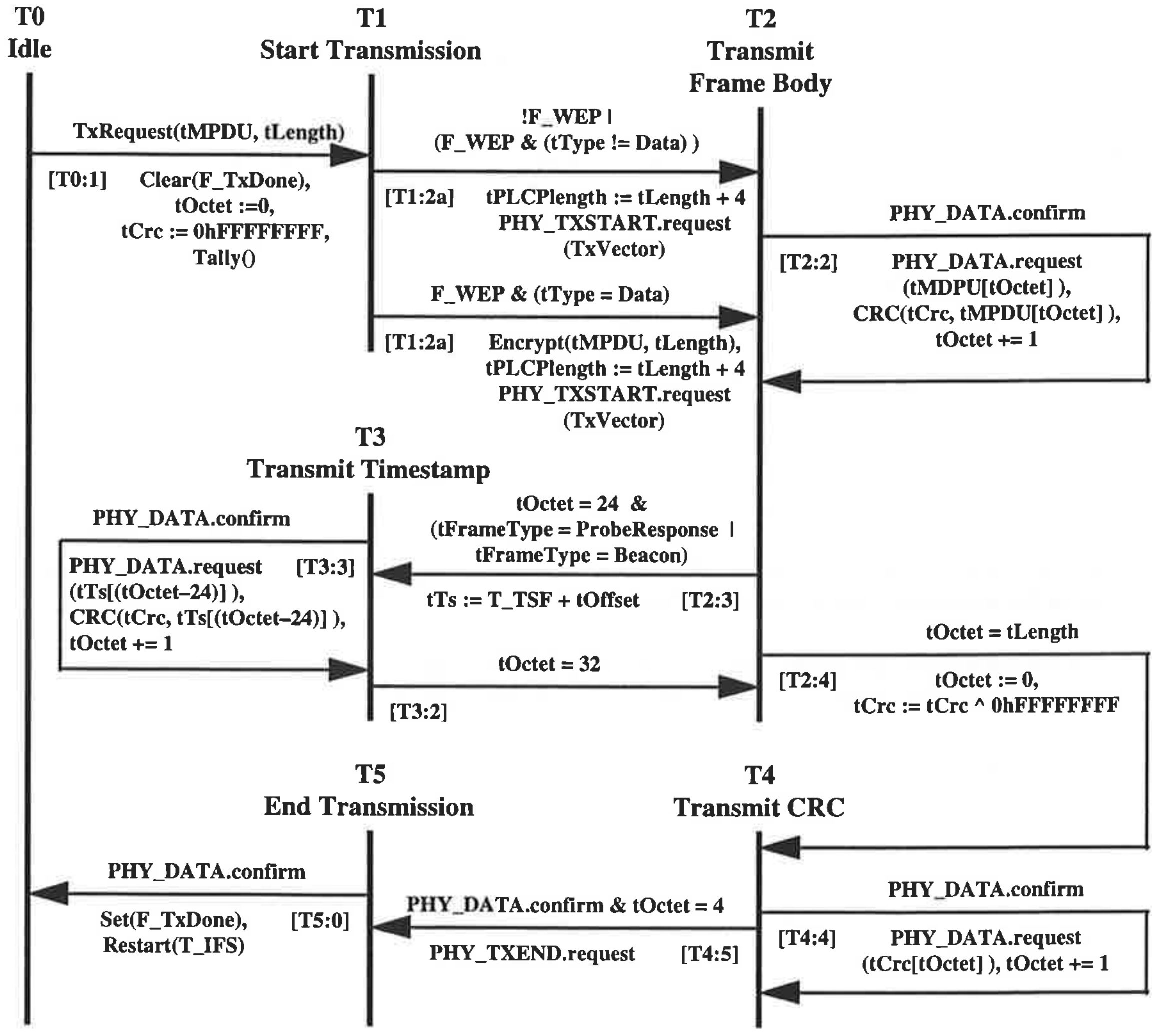

Figure 6-xx(11): Transmitter State Machine

6.7.8.3. Notes to the Transmitter State Machine

State T0, Idle: The MAC transmitter shall enter this state upon initialization or after a transmission is concluded. There is no separate "sleep" state in this state machine because there is no functional difference (to the MAC) between an idle transmitter and a sleeping transmitter.

T0:1, Initialize for transmission: When a transmit request is received from the control state machine, this transition is taken to indicate that a transmission is in progress, to initialize the octet counter, and to initialize the CRC generator. The Tally() call updates the transmission–related counters appropriate for the addressing of the current tMPDU. Relevant counters include aTransmitted_Frame_Count, aOctets_Transmitted_Count, a_Multicast_Transmitted_Frame_Count, and aBroadcast_Transmitted_Frame_Count.

State T1, Start Transmission: Starts the PHY transmission sequence, and encrypts the MPDU payload of Data MPDUs when WEP is active.

T1:2a, Start transmitter without encryption: When no encryption is required, as well as when the encryption option is not supported, this transition is taken to start the PHY transmitter. The length value in the TxVector is 4 octets greater than the tLength value to allow for the addition of the CRC–32. Other elements of the TxVector required by the active PHY shall be generated based on the values of appropriate MIB or global variables.

T1:2b, Start transmitter with encryption (optional): When the WEP option is supported and the MPDU contains a data frame or fragment, this transition is taken to encrypt the MPDU payload. Encryption also inserts the IV and ICV into the MPDU, sets the WEP bit in the frame control field, and increments the tLength value by 8 to allow for the inserted fields. The PHY transmitter is started after MPDU is encrypted and the IV and ICV fields added. The length value in the TxVector is 4 octets greater than the (incremented) tLength value to allow for the addition of the CRC–32. Other elements of the TxVector required by the active PHY shall be generated based on the values of appropriate MIB or global variables.

State T2, Transmit Frame Body: The transmit state machine enters this state after the PHY transmitter has been started, and stays in this state while sending the remainder of the MPDU passed from the control state machine, except for timestamp values required in certain management frames.

T2:2, Send next octet: This transition to self is taken after the PHY confirms the previous request. This transition sends the next octet to the PHY, increments the octet counter, and updates the CRC value.

T2:3, Capture Timestamp Value: This transition is taken after sending the MAC header of Beacon or Probe Response frames, where the first field of the data area is an 8–octet timestamp. The timestamp value is calculated by adding an appropriate (PHY–dependent) tOffset value to the T_TSF value at the time the first octet of the timestamp field is sent to the PHY.

T2:4, End of MPDU payload: This transition is taken when the last octet of the MPDU payload has been transferred to the PHY. The accumulated CRC value is one's complemented in preparation for transmission.

State T3, Transmit Timestamp: Sends the 8 octets of the Timestamp field.

T3:3, Send next timestamp octet: This transition to self is taken after the PHY confirms the transfer of the previous octet, and sends the next octet of the timestamp to the PHY and updates the CRC value.

T3:2, Resume sending payload: This transition is taken after sending the 8th octet of the timestamp.

State T4, Transmit CRC: Sends the 4 octets of the CRC field after the MPDU payload.

T4:4, Send next CRC octet: This transition to self is taken after the PHY confirms the transfer of the previous octet, and sends the next octet of the CRC to the PHY.

T4:5, Stop PHY transmitter: When the PHY confirms the last octet of the CRC field, this transition is taken to indicate the end of the MPDU and initiate the PHY transmitter shutdown sequence.

State T5: End Transmission: Waits for the PHY to confirm the end of data.

T5:0, Return to idle: When the PHY confirms the end of data, this transition is taken to indicate completion of the transmission to the MAC control state machine, start the IFS timer, and suspend operation of the transmitter state machine.

6.7.8.4. Known Limitations of the Transmitter State Machine

The transmitter state machine assumes that the PHY always returns a confirmation for each request, and provides no recovery mechanism for a case where the PHY fails to return a confirmation.

The WEP support in the transmitter state machine only encrypts the MPDU payloads if WEP is active and the frame type is Data. The authentication frames which require an encrypted payload must be encrypted in the management state machine and passed for transmission already encrypted.

6.7.9. Receiver (R) State Machine

The Receiver state machine handles transfer, validation, and duplicate filtering of frames passed from the PHY layer pursuant to reception from the wireless medium. The receiver state machine performs all NAV updates (which increase the NAV value or leave the NAV unchanged), while other state machines perform all NAV resets (which reduce the NAV value, generally to zero). If WEP is active at this station, the receiver state machine handles decryption of the MPDU payloads on incoming frames with the WEP bit set in the frame control field. The Receiver state transition diagram appears in figure 6–xx(12).

6.7.9.1. Local Variables Used in the Receiver State Machine

rAddr1	refers to the contents of the Addr1 field of rMPDU
rAddr2	refers to the contents of the Addr2 field of rMPDU
rAddr3	refers to the contents of the Addr3 field of rMPDU
rCF_Dur_Rem	refers to the contents of the CF_Duration_Remaining entry in the CF parameters element of a beacon frame (or zero if the beacon has no CF parameters element)
rCrc	a 4–octet vector used to accumulate the CRC
rDir	=1 when a valid frame is received addressed to the individual address of this station, =0 when a valid frame is received, from a source within the current BSS, addressed to the broadcast address or to a group address accepted by this station
rDuration	refers to the contents of the Duration/ID field of rMPDU
rType	the contents of the Type and Subtype fields of the frame control field of rMPDU
rFromDS	refers to the contents of the FromDS bit in the frame control field of rMPDU
rMPDU	a vector used to store the MPDU being received from the PHY
rLength	the number of octets in the MPDU provided by the control state machine
rOctet	counts octets received from PHY and indexes rMPDU vector to select next octet for reception
rPLCPlength	used to refer to the value of the length element in the RxVector
rToDS	refers to the contents of the ToDS bit in the frame control field of rMPDU
rTs	an 8–octet vector used to capture the value of T_TSF at the time the first octet of a Timestamp field might be received
rWEP	refers to the contents of the WEP bit in the frame control field of rMPDU

rDurPsPoll is a PHY–dependent constant, not a local variable. See note for transition R5:6e.

6.7.9.2. Receiver State Machine Definition

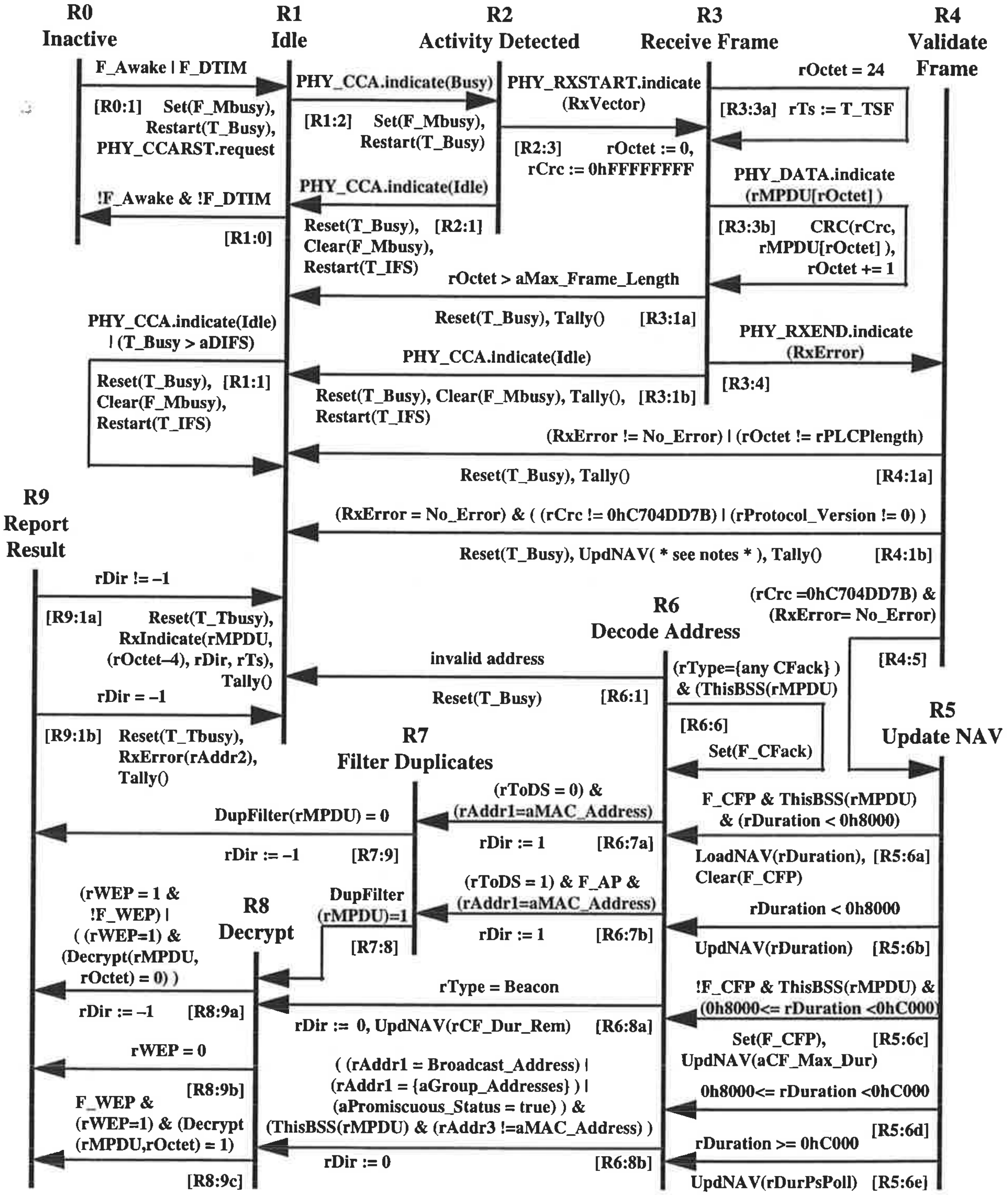

Figure 6-xx(12): Receiver State Machine

6.7.9.3. Notes to the Receiver State Machine

State R0, Inactive: The MAC receiver shall enter this state upon initialization or when idle and told to "sleep" by the clearing of F_Awake. If told to sleep during reception activities the receiver state machine completes the reception before entering inactive state.

R0:1, Wakeup: When F_Awake is set or the start of the next beacon interval (TBTT) for a beacon with DTIM occurs while the receiver is inactive, this transition is taken to prepare for possible reception activities. F_Mbusy is set to avoid starting a transmission before the PHY's CCA function can properly report on the state of the medium. T_Busy is restarted to time the interval after which the continued absence of a CCA assertion indicates a non–busy medium, and the PHY's CCA state machine is reset to initiate assessment of the medium.

State R1, Idle: The MAC receiver enters this state upon wakeup or after reception–related processing is concluded. In this state the MAC waits for the medium to become non–busy after a reception, and continues waiting until new activity is detected on the medium.

R1:0, Go to sleep: Then the F_Awake flag is cleared, and there is not a DTIM expected nor in progress (!F_DTIM), an idle receiver takes this transition to become inactive.

R1:1, Determine medium to be free: At most entries to Idle state the medium is indicated to be busy (F_Mbusy set). This transition clears F_Mbusy, restarts the IFS timer, and resets the busy duration timer when a CCA idle condition is indicated by the PHY, or when T_Busy has measured a DIFS interval without a CCA busy indication from the PHY. This transition is thereby able to serve two purposes. When Idle state is entered from Inactive state, F_Mbusy has been set as a precaution and T_Busy has just been restarted, so this transition will clear F_Mbusy if an appropriate interval elapses without confirmation of the CCA busy condition from the PHY. When Idle state is entered from various other states due to completion of or an error during frame reception, T_Busy has been reset (hence stopped) so the only possible source of this transition is the CCA idle indication which implements the wait for medium free at the end of a reception, and starts the IFS timer at the correct time for proper inter–frame spacing.

R1:2, Medium activity detected: When a medium busy (CCA) condition is indicated by the PHY, this transition is taken to set F_Mbusy and prepare for the possible recognition of a PLCP header by the PHY.

State R2, Activity Detected: The MAC receiver enters this state at the beginning of a CCA busy condition, to await the possible start of a frame reception.

R2:1, Idle medium detected: When the PHY indicates the medium is idle before detecting the start of a frame, this transition is taken to return to idle state, clear F_Mbusy, and restart the IFS timer.

R2:3, PLCP header detected: When the PHY indicates the start of a frame this transition is taken to initialize the octet counter and CRC generator for receiving an MPDU.

State R3, Receive Frame: In this state the MAC receiver assembles the MPDU contents and accumulates the CRC check value as octets are supplied by the PHY. Errors that prevent completion of frame reception are detected in this state, whereas errors that apply to the contents of the frame are checked after leaving this state.

R3:1a, Frame too long: If the number of octets received exceeds the maximum MPDU size, this transition is taken to cease MPDU reception activities and update counters including aError_Count and aFrame_Too_Long_Count. By resetting T_Busy but not clearing F_Mbusy, this transition causes the remainder of the frame to be ignored by the receiver state machine, which is in Idle state awaiting a CCA idle indication.

R3:1b, Medium idle before end of frame: If the PHY indicates an idle medium before indicating the end of the frame, this transition is taken to clear F_Mbusy, restart the IFS timer, and return to Idle state, and update counters including aError_Count and aFrame_With_Protocol_Error_Count.

R3:3a, Capture timestamp reference: When the received octet counter reaches 24, which is the first octet of the MPDU payload for management frames, this transition is taken to capture a copy of the TSF timer, which will be needed if the reception in progress is an error–free beacon or probe response frame.

R3:3b, Receive Octet: When the PHY indicates reception of each of the octets of the MPDU, this transition is taken to place the octet into rMPDU, increment the count of octets received, and update the CRC value.

R3:4, Handle end of frame: When the PHY indicates the end of a reception, this reception is taken to validate the received frame.

State R4, Validate Frame: This state ensures that reception status is error–free, and consistent between PHY and MAC before allowing any interpretation of the contents of the received MPDU.

R4:1a, Frame format or length error: If the PHY reported a receive error, or the number of octets indicated by the PHY differs from the length information obtained from a supposedly correct PLCP header, this transition is taken to discard the MPDU and update counters including aError_Count and aLength_Mismatch_Count.

R4:1b, CRC or protocol error: If the reported no errors, but the CRC–32 remainder is incorrect or the frame control field indicates an unsupported protocol version, this transition is taken to discard the frame and update counters including aError_Count and aFCS_Error_Count. The conditions which enable this transition are unique because the length information from the PLCP header is valid but the contents of the MPDU are not usable. The rPLCPlength provides partial information on which to base a NAV update: If rPLCPlength =14, the frame was CTS or ACK, so the NAV update duration should be (Max_Frame_Length+ACK+2*SIFS) under the assumption that this was a CTS or an ACK for a successor fragment. If rPLCPlength =20, the frame was RTS, CF–End or PS–Poll, so the NAV update duration should be (CTS+Max_Frame_Length+ACK+3*SIFS) because the most probable frame was an RTS. For any other rPLCPlength value, the frame was data or management, in which case the NAV update duration should be (CTS+Max_Frame_Length+ACK+3*SIFS) unless aFragmentation_Threshold is greater than aMax_Frame_Length, in which case the update duration can safely be (SIFS+ACK). This NAV update based on information from a valid PLCP header is of even greater importance in BSAs with stations of mixed data rate capability, since the stations which are unable to receive at the higher data rate cannot interpret MPDU contents. However, to apply this technique to mixed rate environments, the PHYs need to report an "Unsupported_Rate" RxError value, which would be another condition that enabled this state transition.

R4:5, No error: If neither the PHY nor the CRC check detected an error, this transition is taken to update the NAV based on information in the MAC header of the MPDU.

State R5: Update NAV: This state performs the analysis of the MAC header to determine the proper NAV update value. NAV update is performed before address decoding because the NAV is updated based on all valid frames the station received, not just frames addressed to the station.

R5:6a, Update NAV and detect missed CF–End: If the Duration/ID field value is less than 0h8000 on a frame sent in this BSS while the station's F_CFP is set, this station did not receive the CF–End (or CF–End+Ack) which marked the end of the contention free period, so this transition is taken to clear F_CFP and to load the NAV using this frame's duration value, overriding remaining NAV time from the ended CF period.

R5:6b, Update NAV by duration: If the Duration/ID field value is less than 0h8000, and transition R5:6a was not taken, this transition is taken to update the NAV using this duration value.

R5:6c, Updated NAV due to missed CF period start: If the Duration/ID field value is between 0h8000 and 0hBFFF, while the station's F_CFP bit is clear, this station did not properly detect the start of the CF period, so this transition is taken to set F_CFP and to update the NAV to the maximum possible CF period duration.

R5:6d, No NAV update during CF period: If the Duration/ID field value is between 0h8000 and 0hBFFF, and transition R5:6c was not taken, this transition is taken because no NAV update is necessary during the contention free period.

R5:6e, Update NAV for PS–Poll: If the Duration/ID field value is greater than or equal to 0hC000, the field contains a StationID, hence the frame is a PS–Poll. This transition is taken to update the NAV using PsPollDur, which is a PHY–dependent constant of the time required (SIFS+ACK) for the AP to acknowledge the PS–Poll.

State R6: Decode Address: This state decodes the addresses and frame control information in the MAC header to determine whether this station needs to process additional material from the received frame. For stations which attempt to maintain tables on the association and power save mode of other stations in the BSS, this state is an excellent place to gather such information, because all transitions into this state occur with valid MPDUs, and no subsequent states are traversed by all valid MPDUs because of the address filtering done in this state.

R6:1, Invalid address: If the conditions for no other transition from this state (to states R7 or R8) are met, after all allowed transitions to self (state R6) have been taken, this transition is taken to discard the MPDU and wait for the medium to become idle. This "invalid" MPDU address is invalid from the point of view of this station, but may be valid for other stations, possibly in other BSSes.

R6:6, Record CF–Ack: If the MPDU has any of the frame type/subtype combinations which include a contention free acknowledgment, and the MPDU was sent within this BSS, this transition is taken to indicate the acknowledgment by setting F_CFack. The CF–Ack must be detected here because the acknowledgment relevant to a transmission in response to a poll by the point coordinator is contained in the MAC header of the next transmission from the point coordinator, which may not be addressed to the same station as sent the frame being acknowledged. The relevant frame type/subtype combinations are Data+CF–Ack, CF–Ack (no data), Data+CF–Poll+CF–Ack, CF–Poll+CF–Ack (no data), and CF–End+Ack.

R6:7a, Directed frame to station: If the Address1 field of the MPDU contains the individual address of this station, and the MPDU is not directed to Distribution Services, this transition is taken to mark the MPDU as a directed reception (which may require acknowledgment) and perform duplicate filtering.

R6:7b, Directed frame to DS: If the Address1 field of the MPDU contains the individual address of an AP and the MPDU is directed to Distribution Services, this transition is taken to make the MPDU as a directed reception (which may require acknowledgment) and to perform duplicate filtering.

R6:8a, Beacon frame: If the received MPDU contains a Beacon frame, this transition is taken, independent of the addressing of the Beacon frame, to update the NAV based on the CF_Duration_Remaining entry in the CF Parameters element of the Beacon, and to pass the Beacon frame to MAC management for further processing. If there is no CF Parameters element in this Beacon frame the NAV update is null.

R6:8b, Multicast frame from this BSS: If the Address1 field of the MPDU contains the broadcast address or a group address recognized by this station, the appropriate other address field (Address2 or Address3) contains the correct BSSID, and the multicast source is not this station, this transition is taken to mark the MPDU as non–directed (therefore not to be acknowledged) and to skip the duplicate filtering step because non–acknowledged MPDUs cannot be duplicated due to MAC operation. Placing the station in promiscuous receive mode overrides the Address1 filtering, but not the BSSID matching. This causes promiscuous mode to receive all frames originating within the BSS, which is equivalent to its function on wired networks. The test of multicast source address, to exclude multicasts originating at this station, is needed to avoid a station receiving its own multicast a variable, and unpredictable, amount of time after sending that multicast to distribution services in a directed, "ToDS" frame.

State R7: Filter Duplicates: In this state directed receptions, which might be duplicates if prior acknowledgments were lost, are inspected for such duplication and duplicates are discarded.

R7:9, Discard duplicate frame: If the current MPDU is a duplicate, or otherwise out of sequence, this transition is taken to discard the MPDU. rDir is set to –1 to indicate the error prior to entering state 9 to report the reception to the control state machine. The reception must be reported, even though the MPDU is not going to be processed, because if the MPDU were discarded without informing the control state machine the transmission of a directed MPDU would go unacknowledged, thereby wasting time on the medium for retransmission attempts.

R7:8, Retain non–duplicate frame: If the current MPDU is not a duplicate, this transition is taken to continue with processing the MPDU for possible reporting to the control state machine as a valid reception.

State R8: Decrypt: This state handles the decryption and validation of MPDUs encrypted using the privacy function. This state is not optional, because stations which do not support the privacy function may receive frames with encrypted MPDU payloads.

R8:9a, Discard non–decryptable WEP frame: If an MPDU with encrypted payload is received at a station that does not support the privacy function, or if the ICV check fails on the decrypted MPDU payload, this transition is taken to discard the MPDU by setting rDir to –1 so the MPDU is acknowledged but not processed.

R8:9b, Retain non–encrypted frame: If the MPDU is not encrypted, this null transition is taken to bypass the decryption step.

R8:9c, Retain successfully decrypted frame (optional): If the privacy function is supported at the station, and the MPDU payload is successfully decrypted, this transition is taken to remove the IV and ICV fields and to decrement the MPDU length by 8, prior to reporting the reception to the control state machine, thereby rendering the privacy function "transparent" upon reception.

State R9: Report Result: This state completes the reception process and reports the result of the reception to the control state machine. There are two report functions, one for valid receptions and one for erroneous receptions that require acknowledgment.

R9:1a, Report successful reception: If the received MPDU is valid and has an address and type appropriate for processing at this station, this transition is taken to provide the MPDU to the control state machine, along with the rDir value, the saved timestamp (rTs) value, and a length 4 octets shorter than the MPDU length to omit the CRC–32 field. This transition also resets T_Busy, so the receiver state machine will keep F_Mbusy set until a CCA idle condition is reported by the PHY, and updates the values of counters, including aReceived_Frame_Count, aOctets_Received_Count, aBroadcast_Received_Count, and aMulticast_Received_Count.

R9:1b, Report unsuccessful reception: If the received MPDU is a duplicate or is not decryptable, but still requires acknowledgment, this transition is taken to indicate the reception and acknowledgment address to the control state machine without providing the erroneous MPDU. This transition also resets T_Busy, so the receiver state machine will keep F_Mbusy set until a CCA idle condition is reported by the PHY, and updates the values of counters, including aReceived_Frame_Count, aOctets_Received_Count, and aError_Count.

6.7.9.4. Known Limitations of the Receiver State Machine

The receiver wakes up treating the medium as busy in order to avoid interfering with a transmission already in progress. The current definition assumes that a PHY_CCA.indicate() for either busy or idle will occur within a DIFS interval after the receiver and resets the PHY's CCA state machine. If this is not a valid assumption, some other technique must be used to prevent interference from newly–awakened stations when there is a detectable transmission in progress.

In Idle state the PHY receiver is active, whereas in Inactive state the PHY receiver is supposed to be inactive. However, there does not seem to be a uniform means to turn PHY receivers on and off, so this state machine does not issue a request to disable the PHY receiver when entering inactive state.

MPDUs which fail the ICV check upon WEP decryption must be discarded, not reported upward. The privacy function states that MAC management is "informed" of this occurrence, but neither the management functions nor the MIB contain an appropriate means for this to occur. If such a means is defined, the decryption failure is detected at state transition R8:9a.