IEEE 802.11
Wireless Access Methods and Physical Layer Specifications

TITLE: An RF Data Transport Protocol
The RF Adaption Sublayer and RF Physical Layer Specifications
for Slow Hopping Spread Spectrum Radio LAN

DATE: July 12-16th,1993

AUTHOR: Ed Geiger
Apple Computer
One Infinite Loop, MS 301-4J
Cupertino, CA 95014
edg@goofy.apple.com

Introduction

Sending and receiving data in an indoor RF LAN environment requires an RF Data Transport Protocol that will move packets accurately from the transmitter to the receiver. The indoor office environment offers several potentially serious problems to RF LANs: interference, the possibility of severe multipath fading and large propagation losses. Wireless communications systems employing conventional frequency hopping schemes where one or more packets are transmitted on a single RF channel might perform poorly in environments where the channel at any time may be experiencing substantial interference or fading.

One way to minimize the effects of interference and fading on data communications in a wireless LAN is to use "N" number of radio frequency channels during the transmission of a packet whose sequence is known by all potential receivers of the packet. In this system, a packet to be transmitted is divided into segments and each segment is transmitted on a different carrier frequency or channel.

The following is a description of the RF Data Transport Protocol for a Slow Hopping Spread Spectrum Radio LAN which uses multiple frequencies to transmit and receive data. It is independent of any higher layer Media Access Protocol including whether the MAC layer uses a decentralized or centralized communication system.

2.0 Introduction

The RF Data Transport Protocol is full duplex in function, meaning that each layer of the protocol stack performs some operation in either the transmit or receive direction. The best way to describe the protocol is to break it up into several layers and sub layers. These layers and sublayers are as follows:

RF Adaption Layer (RF_AL)
- RF_AL Segmentation and Reassembly
- RF_AL Protocol Data Unit generation and recovery
- Forward Error Correction generation and verification

RF Hopping Protocol Physical Layer (RF_PHY)
- Transmission Convergence Sublayer
    - Header BCH generation and verification
    - Data BCH generation and verification
    - RF Transmission Framing and Framing Recovery
    - RF Hopping Protocol
        - Scanning & Data Reception
        - Carrier Sense and Data Transmission
- Physical Media Dependent Sublayer

3.0 RF Adaption Layer

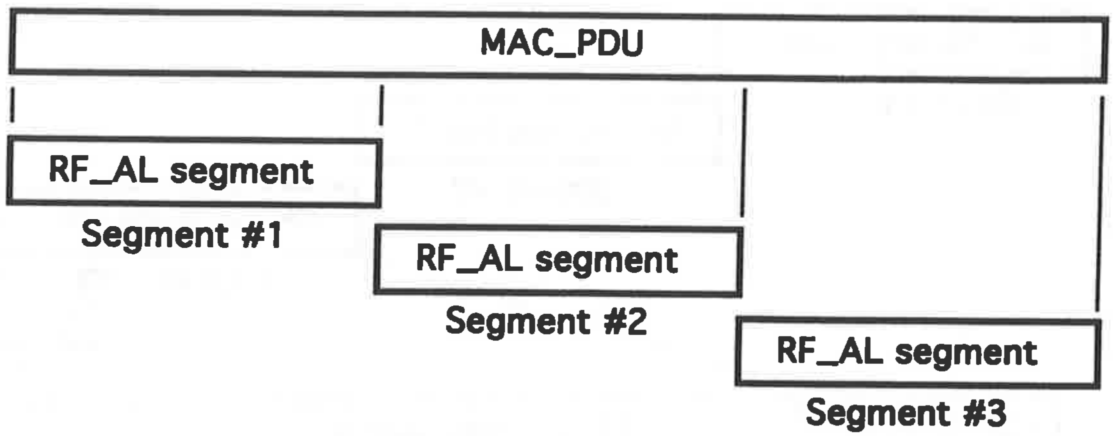
The RF Adaption Layer (RF_AL) performs several functions. In the transmit direction, it takes a MAC Protocol Data Unit (MAC_PDU) and segments into three parts of identical length "S", each called a RF_AL Segment. After segmentation, the RF_AL builds each RF_AL segment into a RF_AL Protocol Data Unit of length "L". It then calculates a sets of forward error corrections codes which make up an additional 2 RFAL_ECDUs also of length "L". Length "L" and the 3 RFAL_PDUs and 2 RFAL_ECDUs are passed down to the RF Hopping Protocol Physical Layer (RF_PHY).

In the receive direction, the RF_AL accepts "N" number of RFAL_PDUs and/or RFAL_ECDUs from the RF_PHY, where "N" is 1,2,3,4, or 5. It addition, it receives the "L" of the RFAL_PDUs and a map of "Fragment Errors" (BCH_FLG) for each RFAL_PDUs and RFAL_ECDUs. The RF_AL will then use this information to reassemble the MAC_PDU. If reassembly is successful, the RF_AL will pass a MAC_PDU to the Mac Layer. If the reassembly was unsuccessful, the RF_AL will send an error indication to the MAC Layer.

The next two sections detail the operation of the RF_AL in both the transmit and receive directions.

3.1 RF Adaption Segmentation

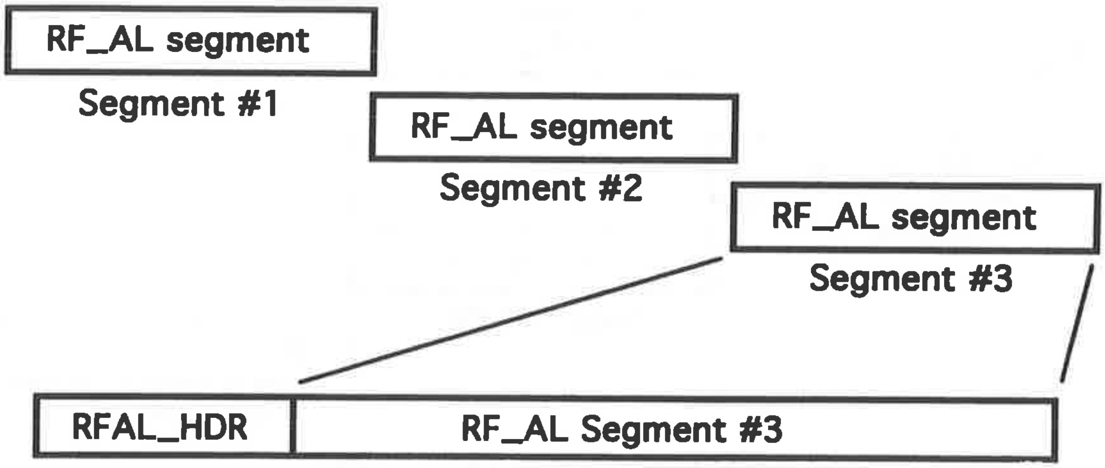
The MAC Layer passes the RF_AL a MAC_PDU packet of data which is of length "M" were "M" is an integer > or = 0 and < or = 753. Integer "M" must also be such that "M" MOD 3 = 0. Figure 3.0 shows how the MAC_PDU is segmented into the first 3 RF_AL Segments, each of length "S".



Segmentation of the MAC_PDU
Figure 3.0

3.2 Creation of RF Adaption Layer Protocol Data Unit

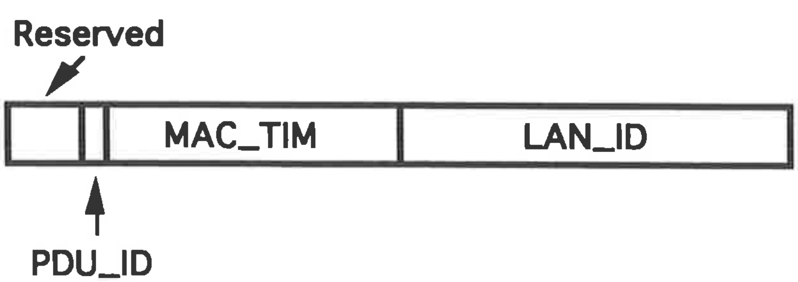
After segmentation, the RF_AL forms each RF_AL segment into a RFAL_PDU of length "S+4" or "L". This is done by adding a four byte RF_AL Header (RFAL_HDR) to the front of each RF_AL segment. Figure 3.1 shows the formation of RFAL_PDU #3 using the RF_AL segment #3. The RFAL_PDUs for segments 1, 2 and 3 each use the same RFAL_HDR.



Creation of RFAL_PDU #3
Figure 3.1

3.2.1 RFAL Header Generation

Figure 3.1a shows the format of the RFAL_HDR. It is made up of four fields. The first field on the right is the LAN identification field (LAN_ID) which is used by the MAC layer to identify the packet with a specific LAN. The second field from the right is a MAC layer timing field (MAC_TIM). The third field is a single bit used to identify the packet as a user data packet or MAC layer management packet. The final field is reserved for future use. This header appears in both the RFAL_PDUs and RFAL_ECDUs.

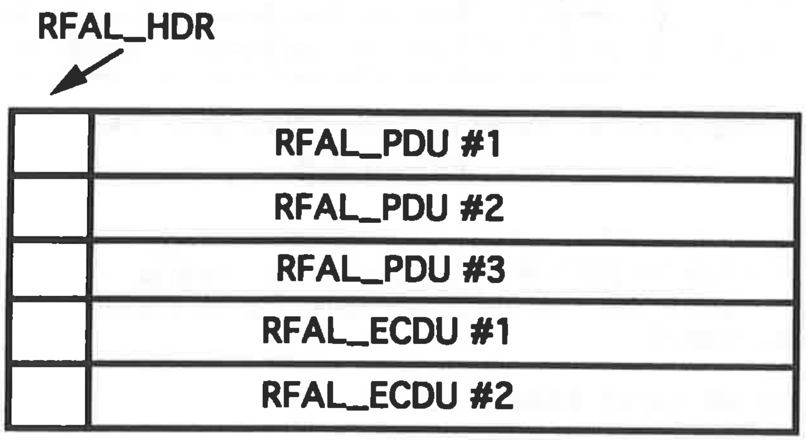


RFAL Header Format
Figure 3.1a

3.3 RF Adaption Layer Forward Error Correction Generation

After the RF_AL has built the 3 RFAL_PDUs from the MAC_PDU, it then produces an additional 2 sets of RFAL_PDU like data streams called RFAL Error Control Data Unit or RFAL_ECDU. Each RFAL_ECDU contain forward error correction information for rebuilding the RFAL_PDUs in case of lost or erred PDUs. This forward error correction scheme is designed so that only three of the five PDUs or ECDUs must be received correctly at the end station so that the receiving node can reconstruct the MAC_PDU.

Figure 3.2 shows the order of the PDUs and ECDUs sent by the RF_AL to the RF_PHY. Each PDU and ECDU is of length "L". Length "L" is an integer > or = 4 and < or = 255. In addition, each RFAL_ECDUs contains the RFAL_HDR just as the RFAL_PDUs.



The RFAL_PDUs and RFAL_ECDUs which pass to the RF_PHY for each MAC_PDU.
Figure 3.2

3.4 RF Adaption Layer Forward Error Correction Verification and Rebuild

The RF Adaption Layer in the receive direction may have to perform forward error correction to reconstruct any data lost during transmission. When the RF_PHY has finished receiving all the PDU's it can for a particular MAC_PDU, it notifies the RF_AL and passes it a RF Transport Convergence Burst Protocol Data Unit (RF_TC B_PDU) (See section 4.1.6) for each PDU it captured. These RF_TC B_PDUs can be of any combination of RFAL_PDUs and RFAL_ECDUs and could number 1,2,3,4, or 5 in quantity. If the total number of PDUs or ECDUs received is 1 or 2, the RF_AL will not have enough information to reconstruct the MAC_PDU. If the total number of PDUs and ECDUs is three or more, the RF_AL will construct a map as shown in figure 3.3.

The reconstruction map is built using several pieces of information from each of the RF_TC B_PBUs it receives from the RF_PHY. The first step is to identified which PDUs and ECDUs were received. The RF_TC ID field is used to determine this. As shown in figure 3.3 for a certain MAC_PDU, there were four PDUs and ECDUs received, RFAL_PDUs #1 and #2, and RFAL_ECDUs #1 and #2. RFAL_PDU #3 was not received.

Next the BCH Flag Field (BCH_FLG) for each RF_TC B_PDU was used to mark each fragment as good or bad. If every column has at least 3 "GOOD" fragments, the MAC_PDU can be rebuilt. In the case of figure 3.3, column five has only 2 "GOOD" fragments and can't be built, thus this MAC_PDU must be tossed.

	1	2	3	4	5	6	7	8
RFAL_PDU #1	Good	Good	Good	Good	Good	Good	Bad	Good
RFAL_PDU #2	Good	Bad	Good	Good	Bad	Good	Good	Good
RFAL_PDU #3	Miss	Miss	Miss	Miss	Miss	Miss	Miss	Miss
RFAL_ECDU #1	Good	Good	Good	Good	Bad	Good	Good	Good
RFAL_ECDU #2	Good	Good	Good	Good	Good	Good	Good	Good

Forward Error Correction Reconstruction Map
Figure 3.3

There is also some obvious time savings steps that can be implemented. In the case where the BCH_FLGs for PDUs #1, #2, and #3 are all zero's, no reconstruction is required. It also follows that if the top three fragments in any column are all "GOOD", no further action is required.

3.5 RF Adaption Layer Reassembly

Once the RF_AL has completed the error recovery process, the RFAL_ECDUs are tossed and the RFAL_PDUs are disassembled into RF_AL HDRs and RF_AL Segments as described in reverse in figure 3.1. Then, as shown in reverse in figure 3.0, the RF_AL Segments are re-assembled back into a MAC_PDU.

4.0 RF Physical Layer

The RF Physical Layer is divided into two sublayers: the RF Transmission Convergence Sublayer (RF_TC) and the RF Physical Media Dependent Sublayer (RF_PMD). The RF Transmission Convergence Sublayer prepares the RFAL PDU and ECDUs for transmission and is also responsible for the implementing the RF hopping protocol. The RF Physical Media Dependent Sublayer is responsible sending or receiving bits over the RF media. The following sections will describe in detail the RF_TC Sublayer.

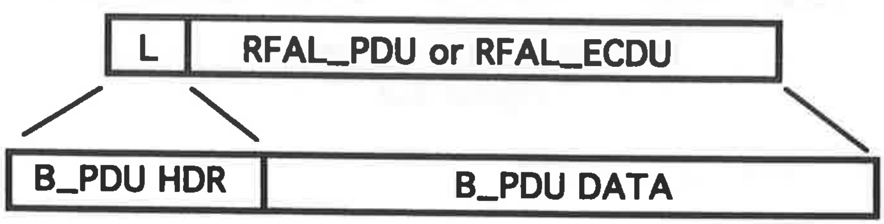
4.1 RF Transmission Convergence Sublayer

The RF Transmission Convergence Sublayer (RF_TC) performs several functions. In the transmit direction, it takes each RFAF_PDU and ECDU and builds a Burst Protocol Data Unit or B_PDU. Once the RF_TC determines that the Hop Group is clear for transmission by performing Carrier Sense, it then enters the Transmit state. In this state it adds framing bits to each B_PDU and passes these bits and the B_PDU to the RF_PMD.

In the receive direction, the RF_TC Scans the Hop Group (HOP_GRP) looking for a B_PDU framing sequence from the RF_PMD. Once it detects the framing sequence and receives a valid B_PDU, it enters the Receive state. Once in the Receive state, the RF_TC layer will be synchronize with the transmitting node and attempt to receive the other four B_PDUs. Once the RF_TC has attempted to receive the full set of B_PDUs, it will strip off the B_PDU formatting to reveal any of the RFAL_PDUs and the RFAL_ECDUs it received. It then will pass the RFAL PDUs and ECDUs to the RF_AL with the FE_MAP.

4.1.1 RF_TC Burst Protocol Data Unit Generation

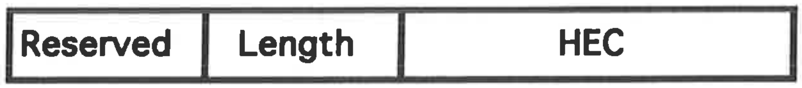
The RF_TC takes the length "L" and the RFAL_PDUs and RFAL_ECDUs it receives from the RF_AL and converts the "L" parameter and RF_AL data to a Burst Protocol Data Unit (B_PDU). Figure 4.0 shows that the B_PDU is made up of two parts: the B_PDU Header (B_PDU HDR) and the B_PDU DATA.



B_PDU Header and Data Fields
Figure 4.0

4.1.1.1 RF_TC Burst Protocol Data Unit Header Generation

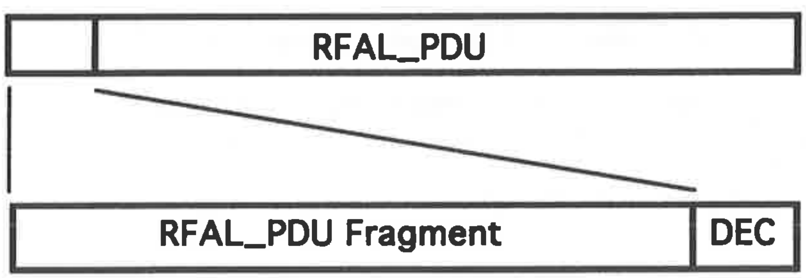
Figure 4.1 shows the construction of the B_PDU Header. The first field is a reserved at this time. The second field is the length field which contains the value of "L". "L" currently is an integer of 0 through 255. The third field is called the Header Error Control (HEC). This field contains a 16 bit BCH code used to protect the first two bytes in the B_PDU HDR against error.



Four Byte B_PDU Header Fields
Figure 4.1

4.1.1.2 RF_TC Burst Protocol Data Unit Data Generation

The B_PDU Data field is created as shown in figure 4.2 by first dividing the RFAL PDU or ECDU into subunits called RFAL_PDU Fragments. Each fragment currently consists of a 14 byte field of RFAL PDU or ECDU data and a one byte Data Error Control (DEC) field. The DEC field contains an eight bit BCH code used to protect the 14 byte data field against error. Since "L" can be any integer from 0 to 255, the final B_PDU Fragment in the B_PDU Data field may not be a full 14 bytes but some number greater than zero and less than or equal to 14.

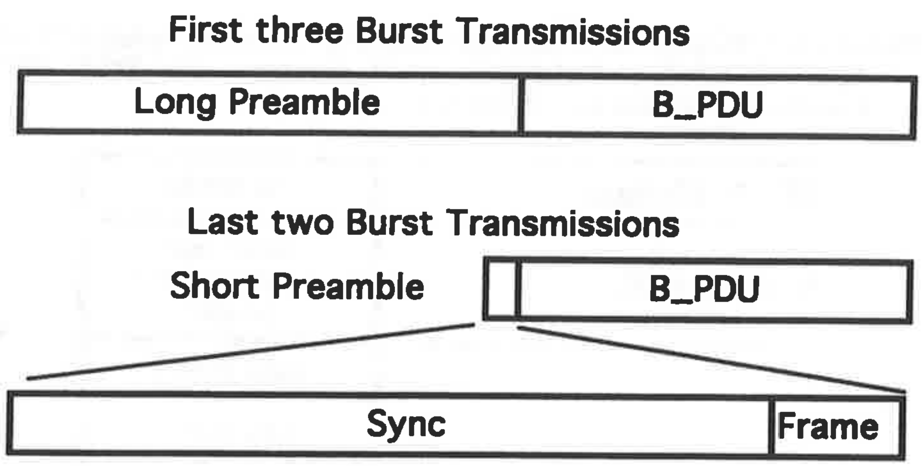


14 Byte B_PDU Fragment
Figure 4.2

4.1.2 RF_TC Framing Generation

The RF Hopping Protocol which will be discussed in detail in the next section, requires that a certain framing structure be used during transmission. This framing structure has two versions as illustrated in figure 4.3, one which precedes each of the first three B_PDUs to be transmitted and the second version which precedes the last two B_PDUs to be transmitted. As also shown in figure 4.3, the preamble is made up of two types of fields, the SYNC field (SYNC_FLD) and the FRAME field (FRM_FLD). The SYNC_FLD contains a fixed number of SYNC symbols and the FRM_FLD contains one FRAME symbol. (A symbol is a set of bits of a known length). The SYNC symbol is used primarily by the RF_PMD at the receiver to recover the bit clock and data. It also can be used by the receiver to identify that the data stream is that of the preamble and not the data portion of a B_PDU. In addition, the receiver will use the preamble to distinguish a friendly transmission from a foreign transmission.

The FRAME symbol is a sequence of bits used to signal the start of the B_PDU. The FRAME symbol is designed to be unique enough that when coupled with a fixed number of good SYNC symbols the probability that the resulting pattern be found in normal data would be extremely low.



B_PDU Preambles
Figure 4.3

4.1.3 RF_TC Framing Verification - SYNC Recovery

Framing verification is performed at the receiver. This verification first involves looking at the channel and determining if there really is data on the channel. Once it is determined that real data exists on the channel, the data pattern is searched for the SYNC or FRAME symbols after which a determination is made that the data might be something appearing in the middle of a burst. (Insert SYNC state machine, FRAME state machine. This to be finished later).

If there is no data, the SYNC state machine returns sync done (SYNC_DNE). If the SYNC state machine finds data but not SYNC_FLD or FRM_FLD, it returns SYNC_DNE with the SYNC_BSY flag indicating that the channel was busy. If the SYNC state machine finds sync it will return a SYNC indicator and continue until FRAME. At this point the frame recovery of the B_PDU will begin.

4.1.4 RF_TC Burst Protocol Data Unit Header Verification

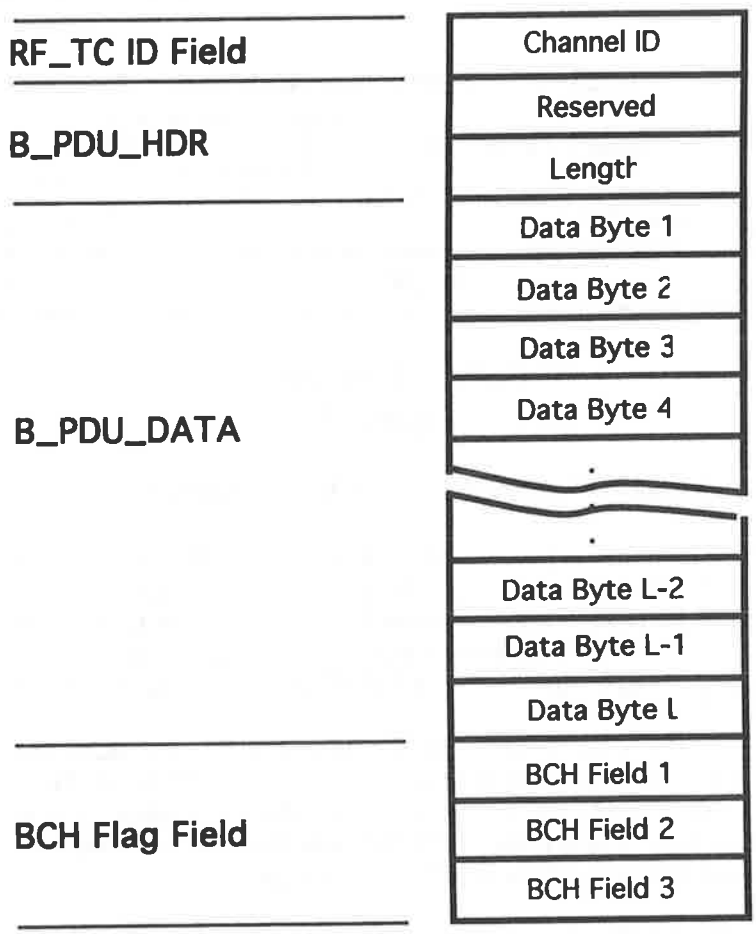
Upon the reception of the first potentially valid B_PDU_HDR, the RF_TC will compare its calculated BCH code of the header versus the one transmitted in the HEC field of the B_PDU_HDR. If they match, the B_PDU_HDR will be declared valid and the length field "L" will be saved. If the BCH check fails, the B_PDU will be ignored. If the B_PDU is not the first, then RF_TC will mark the B_PDU_HDR as bad but will continue to receive the B_PDU.

4.1.5 RF_TC Burst Protocol Data Unit Data Verification

After the framing and header verification requirements of a B_PDU have been meet, the RF_TC begins doing B_PDU Data Verification. Data verification is done over every fragment by comparing the transmitted BCH code in the DEC field with the calculated BCH code at the receiver. If the transmitted DEC field matches the calculated BCH, the fragment is marked good. If the check fails, the fragment is marked bad.

4.1.6 RF_TC Burst Protocol Data Recovery

When the RF_TC receives a completed B_PDU and has completed BCH verification of the header field and all data fragments, it returns a RF_TC B_PDU to RF_AL using the format as shown in figure 4.4.



Format for Receiver RF_TC B_PDU
Figure 4.4

The receive B_PDU is built up into four parts: the RF_TC Channel ID field

(RFTC_ID), the B_PDU_HDR field, the B_PDU Data field (B_PDU_DATA), and the BCH Flag field (BCH_FLG). Each of these fields are used to pass some indications or data to the RF_AL for each B_PDU received for a given MAC_PDU. The RFTC_ID field is used to tell the RF_AL from which channel in the HOP_GRP this B_PDU was received. Remember that not all burst might be received because of interference. This channel ID will be used to help the RF_AL reconstruct any missing burst. This field will have a 0,1,2,3,or 4 in it representing channels 1,2,3,4, or 5.

The B_PDU_HDR field will contain the first two bytes of the received B_PDU_HDR: the reserved byte and the length "L" byte with the HEC field removed. The next set of "L" bytes is the B_PDU Data field or B_PDU_DATA. This field is equivalent to the RFAL_PDU but is uncorrected. The final field is the BCH_FLG field. This is a three byte field which contains a BCH Flag indicator for each of the BCH fields the burst contained. If the bit for a particular BCH field is set, the indication is that the BCH calculation failed and the header or fragment is in error. If the BCH calculation is correct, then the bit for that field will be cleared. The following is a table for the three BCH Flag fields.

Byte	Bit	BCH Field
BCH Field 1	7	Header BCH
	5	Fragment 1
	4	Fragment 2
	3	Fragment 3
	2	Fragment 4
	1	Fragment 5
	0	Fragment 6
BCH Field 2	7	Fragment 7
	6	Fragment 8
	5	Fragment 9
	4	Fragment 10
	3	Fragment 11
	2	Fragment 12
	1	Fragment 13
	0	Fragment 14
BCH Field 3	7	Fragment 15
	6	Fragment 16
	5	Fragment 17
	4	Fragment 18
	3	Fragment 19
	2	always 0
	1	always 0
	0	always 0

If all the BCH calculations passed for this burst, then the three BCH bytes would contain all zeros. If the third data fragment in a B_PDU was bad, then bit three in BCH Field 1 would be set to a zero. Since each data fragment is 14 bytes or less and each B_PDU cannot exceed 255 bytes, then the most fragments within a B_PDU will be 19. Bits 2,1, and 0 of BCH Field 3 will always be zero.

4.2 The RF Hopping Protocol

The following section describes in detail the implementation of the RF Hopping Protocol. The RF spectrum is divided into multiple channels or frequencies. These channels are grouped into multiple sets of five unique channels called a Hopping Group (HOP_GRP). The Wireless LAN under control of the MAC Protocol will visit each of these HOP_GRPs during a specified period of time following a predetermined Hop Sequence. Data is only transferred between MAC Protocol entities operating within the same HOP_GRP in the Hop Sequence. This allows multiple Wireless LANs to operate in the same spectrum without interference from each other providing they are located in different points in time within the Hop Sequence. These requirements are generally met within the MAC Layer Protocol and don't necessarily affect the operation of the RF Data Transport Protocol described within this document.

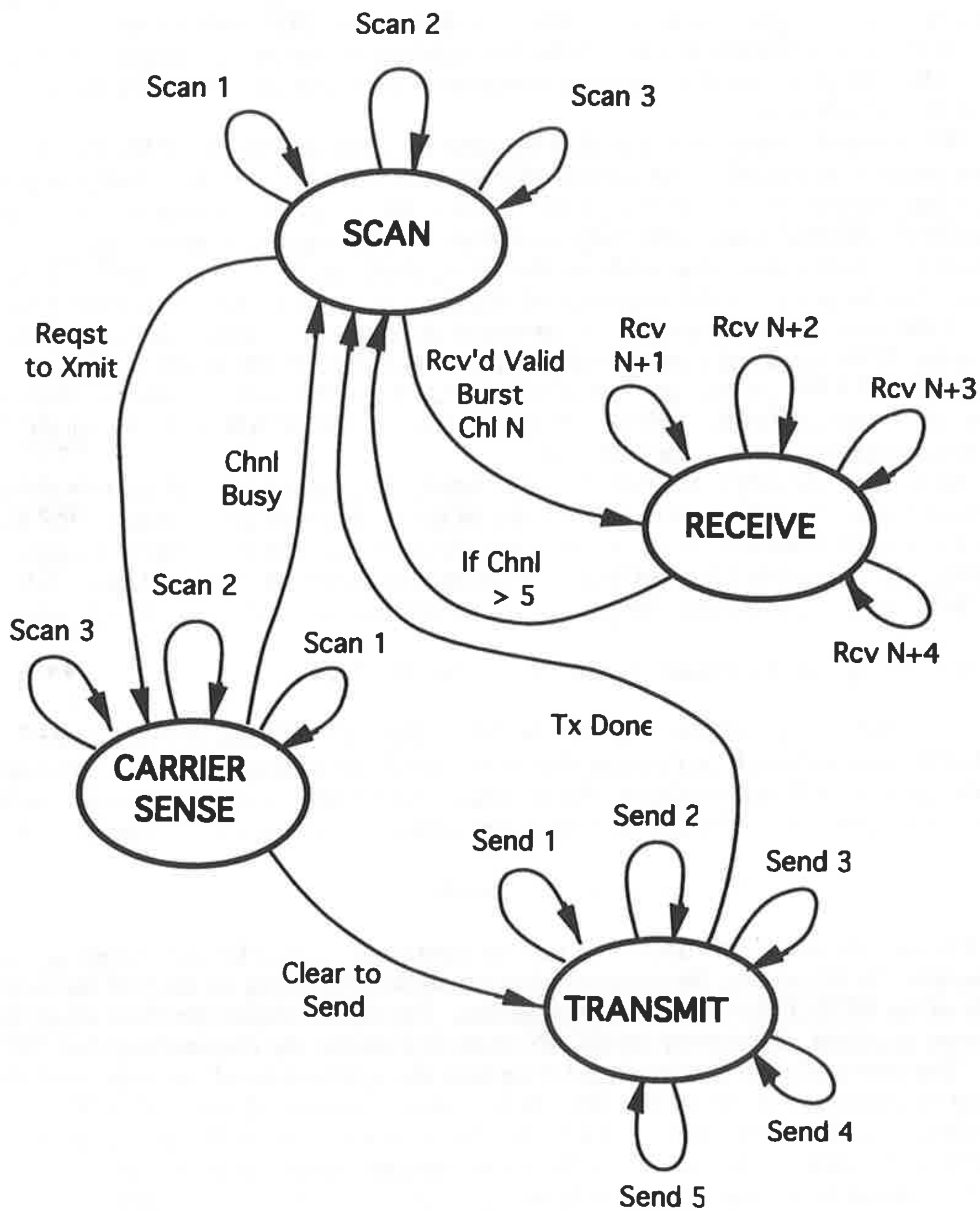
The actual transmission of data takes place by sending each of the five B_PDUs associated with a single MAC_PDU out over the RF_PMD, one per frequency in the HOP_GRP. B_PDU #1 will be transmitted on channel #1, B_PDU #2 on channel #2, and so on. The receiver scans the first three channels in the same HOP_GRP looking for a framing indication on any one of the first three channels. If the receiver is successful in synchronizing with the transmitter on any one of the first three frequencies it is scanning, it then should be able to follow the transmitter through the HOP_GRP as it continues to transmit the remainder of the five bursts.

The advantage of this protocol is many fold. First, the receiver only needs to hear any one of the first three burst to become synchronized with the transmitting node. Once synchronized, the receiver will then know when to expect the next burst on the next channel within the HOP_GRP. The receiver must then receive three of the five burst intact to reconstruct the MAC_PDU. Interference can cause the receiver not to hear the first and or second burst of a transmission but the receiver can still be successful by syncing up to the third burst and receiving the fourth and fifth. In a different interference scenario, the receiver can hear the first burst, synchronize with the transmitter, miss two of the next four burst because of interference but still rebuild the MAC_PDU.

4.2.1 RF Hopping Protocol State Machine Overview

The RF Hopping Protocol State Machine has four high level states: SCAN, RECEIVE, CARRIER SENSE, TRANSMIT. SCAN and RECEIVE are states designed to listen for and receive data. CARRIER SENSE and TRANSMIT are states dealing with data transmission. Figure 4.5 is a high level state diagram showing some of the events which drive the protocol and how each state interacts with the others. Each of these states have fixed operational requirements which the protocol must execute exactly in order for the protocol to function. These operations will be detailed in the following sections. This section will give a general overview of the high level protocol.

The default state of the protocol state machine is SCAN. When in this state, the protocol is basically continually scanning or listening for data on the first three channels of the HOP_GRP. If the protocol detects a transmission on any of these three channels, it will attempt to recover the B_PDU and synchronize with the transmitter. Successful synchronization of the receiver to the transmitter only requires that the B_PDU HDR be found to be error free. A good B_PDU HDR then provides the state machine with a valid length for this B_PDU and the remaining B_PDUs for this transmission. This also provides the time measurement means to get to the next burst on the next channel and all remaining bursts and channels for the current transmission.



RF Hopping Protocol High Level State Diagram
Figure 4.5

After the protocol state machine determines it has recovered a valid first burst, it will advance from the SCAN state to the RECEIVE state. Once in this mode, the protocol state machine is now synchronized with the transmitter and will hop to the next channel along with the transmitting node. The protocol state machine will now time the preamble and B_PDU of the next burst to determine when each should appear. If it sees the SYNC_FLD and FRAME, it will receive the next B_PDU. If some interference causes the preamble and B_PDU to be missed, the RECEIVE state machine will insert an ERASURE DELAY which is designed to simulate the preamble and B_PDU time so that synchronization is maintained between the transmitting and receiving nodes during lost bursts. After the protocol state machine receives or fails to receive the fifth burst, it will return to the SCAN state.

When a node wants to transmit, a Request to Transmit (RQST_XMIT) is submitted to the protocol state machine. At the end of each scan during the SCAN state, the protocol state machine checks for any RQST_XMIT. If it is set, the protocol state machine enters the CARRIER SENSE state. The purpose of this state is to provide an means for determining if there is any other node on the air transmitting first. The CARRIER SENSE state functions by scanning the sequence of channels 3, 2 and 1. This sequence is the reverse of the scan sequence used by the protocol state machine when scanning in the SCAN state. If the protocol state machine detects valid data which might be a SYNC_FLD, FRAME, or any portion of a B_PDU, it will declare the channel BUSY. It will then abort the CARRIER SENSE state and return to the SCAN state. In the SCAN state it will attempt to recover the burst, etc.

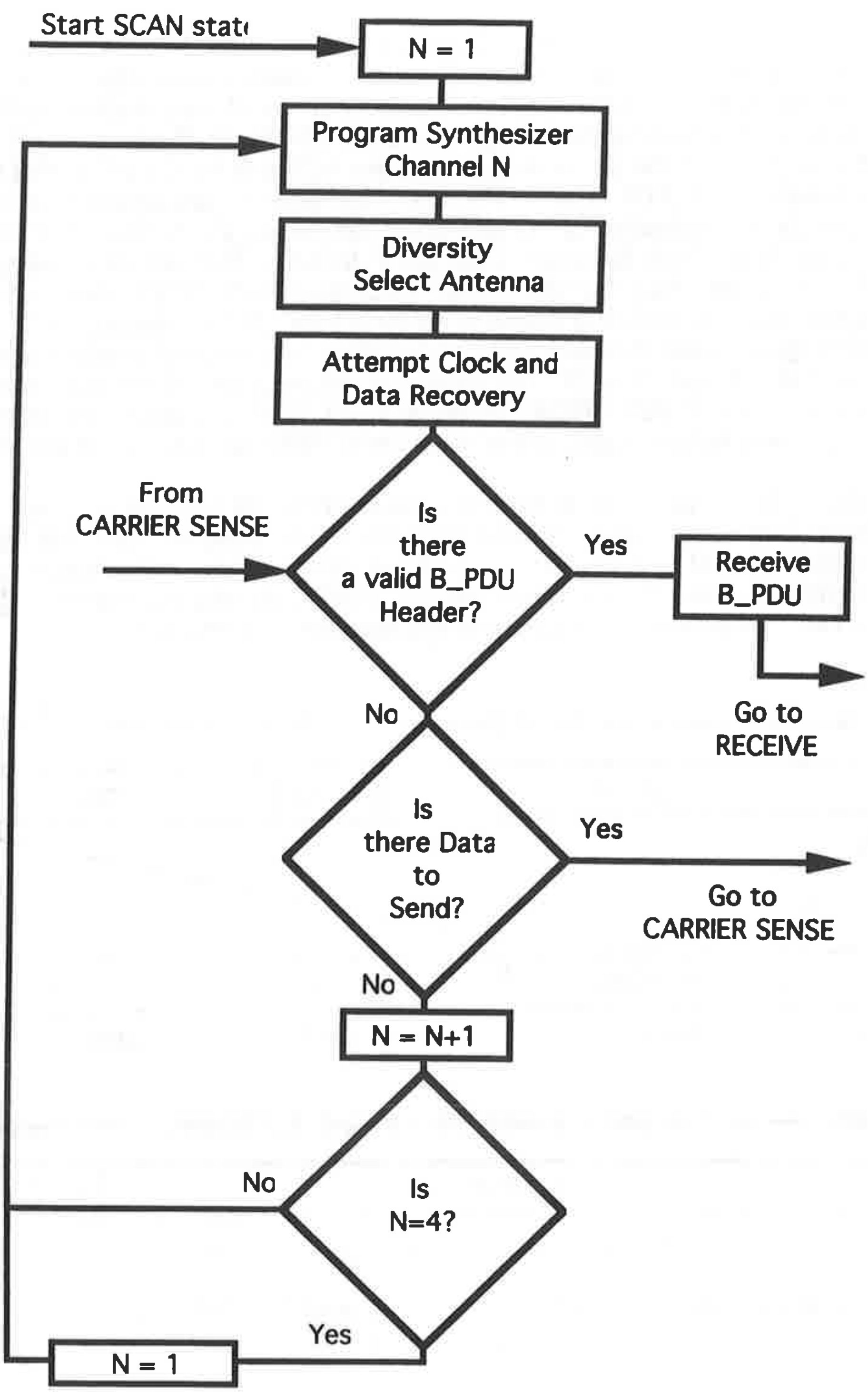
When the CARRIER SENSE state has finished scanning each of the first three channels and has detected no valid data on any of these channels, the protocol state machine advances to the TRANSMIT state. At this time the protocol state machine will begin transmitting the PREAMBLEs and B_PDUs generated by the RF_TC sublayer. When the TRANSMIT state is completed, the protocol state machine returns to the SCAN state.

4.2.2 RF Hopping Protocol State Machine & Timing

In the following sections, the SCAN, RECEIVE, CARRIER SENSE and TRANSMIT state will each be described in more detail. In addition, the timing required by each state machine will be detailed so that synchronization between the transmitter and receiver can be achieved. The importance of the preamble timing will also be described.

4.2.2.1 RF Hopping Protocol SCAN State

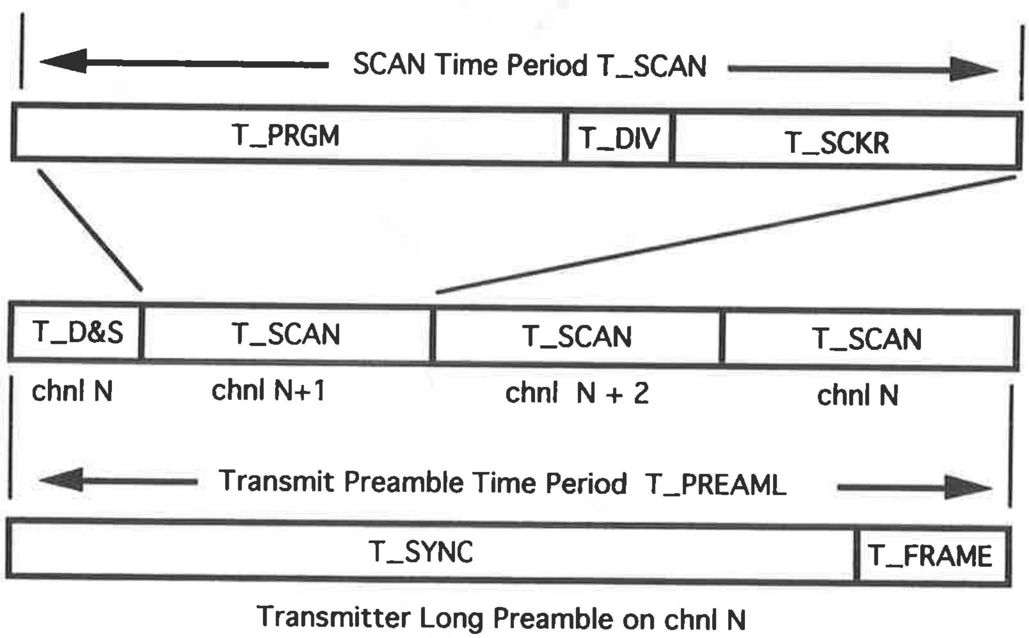
The SCAN state is the default or normal operating state of the RF Hopping Protocol state machine. In this mode, the protocol state machine is listening on each of the first three channels of the HOP_GRP for any transmissions. Figure 4.6 shows the flow chart for the SCAN state machine. Whenever the SCAN mode is entered, the channel number "N" is set to 1. The protocol state machine then programs the synthesizer of the radio with the frequency of channel 1 in the HOP_GRP. It then waits a period of time called T_PRGM for the radio to settle on the channel. Next the state machine does a diversity function which selects the antenna (assumes that the radio has more than one antenna) with the best signal. This function requires the radio to wait a period of time called T_DIV. When this function has completed, the protocol state machine advances to the clock and data recovery state.



SCAN State Machine
Figure 4.6

The clock and data recovery function is where the protocol state machine attempts to recover clock and data from the channel. If there is no node transmitting, no clock or data will be detected. If there is valid clock and data, the protocol state machine will look for the SYNC symbol to determine if the data pattern is the SYNC_FLD of a burst. If a SYNC_FLD is detected, the protocol state machine will wait for the FRM_FLD and then attempt to receive the B_PDU_HDR. There is a time limit for attempting to recover clock and data, and the determination of the data being the SYNC_FLD. This time period is specified as the Scan Clock Recovery Time or (T_SCKR). If the clock and data recovery function doesn't successfully find the SYNC_FLD after the T_SCKR time period, it will abort the clock and data recovery function. If the SYNC_FLD is found, the T_SCKR period will be ignored and the clock and data recovery function will continue until a FRAME symbol is found. If a FRAME symbol is detected, the SCAN state machine will attempt to recover the B_PDU_HDR. IF the B_PDU_HDR is valid, (the HEC field is correct) the protocol state machine will receive the B_PDU and advance to the RECEIVE state.

If the T_SCKR period ends without a valid SYNC_FLD being detected or if a B_PDU_HDR is received with an invalid HEC, the SCAN state machine will check to see if there is a RQST_XMIT. If there is, the protocol state machine will advance to the CARRIER SENSE state. If there is not a RQST_XMIT, the channel number will be incremented and the protocol state machine will scan the next channel.



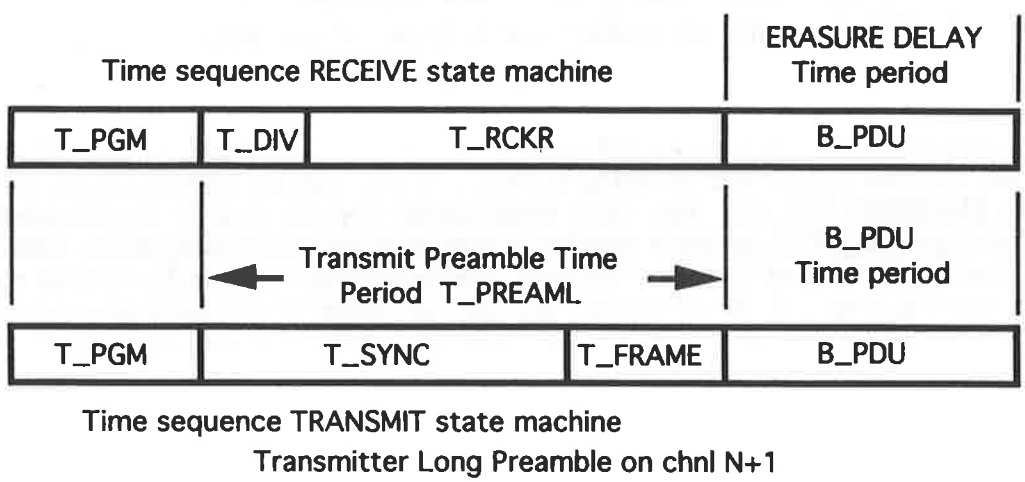
Relationship Between T_SCAN and T_PREAML
Figure 4.7

Figure 4.7 shows the relationship between the time for scanning one channel (T_SCAN) and the time for the transmitter to transmit a long preamble (T_PREAML). As the top of the figure shows, T_SCAN is the sum of several time periods: T_PRGM + T_DIV + TSCKR. The bottom of the figure shows the transmission time for a long preamble of time period T_PREAML. The preamble time period T_PREAML is the sum of the time period of the SYNC_FLD (T_SYNC) plus the time period for the FRM_FLD (T_FRAME).

The center of the figure compares the time it takes a receiver to scan three channels versus a node transmitting a long preamble. Lets say that a node begins transmitting on channel N while a receiver running asynchronously with the transmitting node is trying to do diversity and recover clock and data on channel N (T_D&S). If the transmitter started transmitting just long enough after the receiver has started the diversity and recovery function, the receiver may not have enough time to complete the functions it must to detect SYNC on that channel before the T_SCKR time expires. The receiver will then finish the scan on channel N and proceed to N+1. It will then proceed to N+2 and back to N. During this time the transmitter will still be sending the preamble. In theory this will guarantee that any receiver in any scan state will always be able to get on channel and synchronized to a transmitting node during the long preamble of a burst. If the channel N is being interfered with so that any receiver can't recover the B_PDU, the same receive and transmit scenario exists when the transmitting node begins sending the preamble on channel N+1 and N+2.

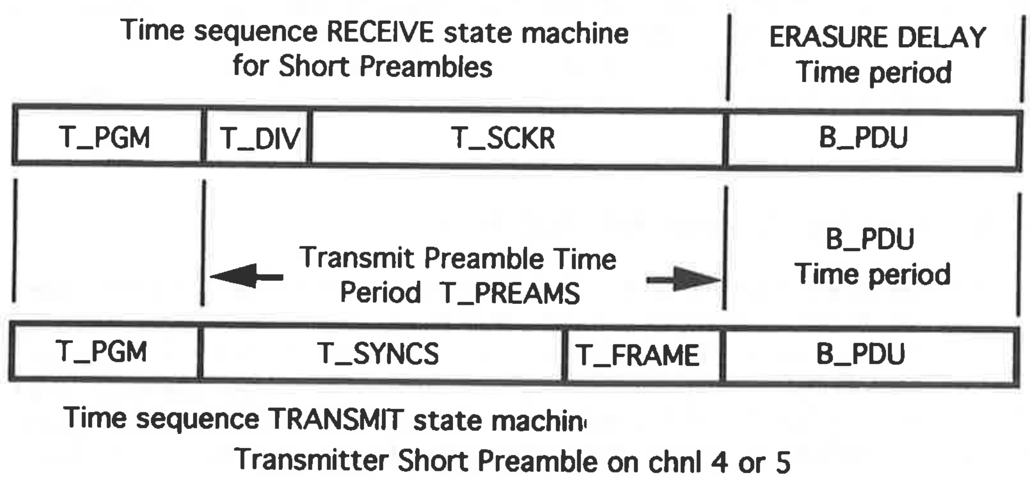
4.2.3 RF Hopping Protocol RECIVE State

The RECEIVE state machine is very similar to the SCAN state machine with a few exceptions. The requirement for entering the RECEIVE state is the reception of a valid B_PDU_HDR. Since all B_PDU_HDRs are identical for the B_PDUs of the same MAC_PDU, the RECEIVE state machine does not have to search for any more valid B_PDU_HDRs. The length field found in the first valid B_PDU_HDR will be used by the RECEIVE state machine to capture the remaining B_PDUs or to insert ERASURE DELAYs.



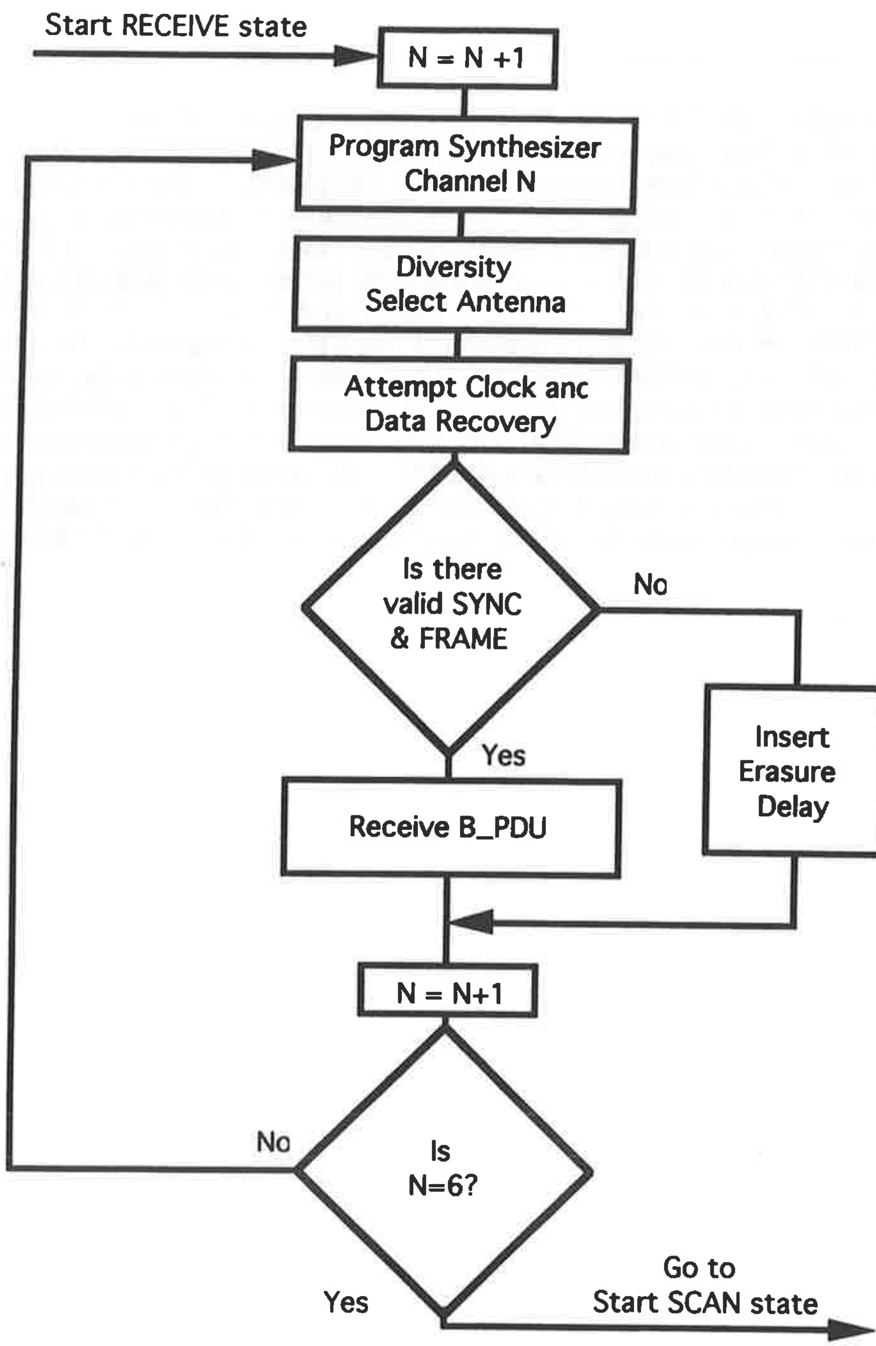
Timing Relationship Between TRANSMIT & RECEIVE state Machines
Figure 4.8

The RECEIVE state machine starts out as show in figure 4.10 by incrementing the channel number and programming the synthesizer for the next channel. This requires a time period of T_PGM as in SCAN . Diversity is next and its time period is again T_DIV. Since there is no longer a requirement to quickly scan because the receiver is now synchronized with a transmitting node, the RECEIVE state machine can take a time period of T_RCKR to attempt to recover clock and data, detect the SYNC_FLD and then see the FRM_FLD as shown in figure 4.8 for the first 3 burst. This time period reverts back to T_SCKR for the final two burst as shown in figure 4.9. Bursts 4 and 5 are transmitted with shorter preambles because any receiver which is attempting to acquire the packet must have received at least one of the first three burst and be in sync in order to have a need to receive the last two.



Timing Relationship Between TRANSMIT & RECEIVE state Machines for a Short Preamble
Figure 4.9

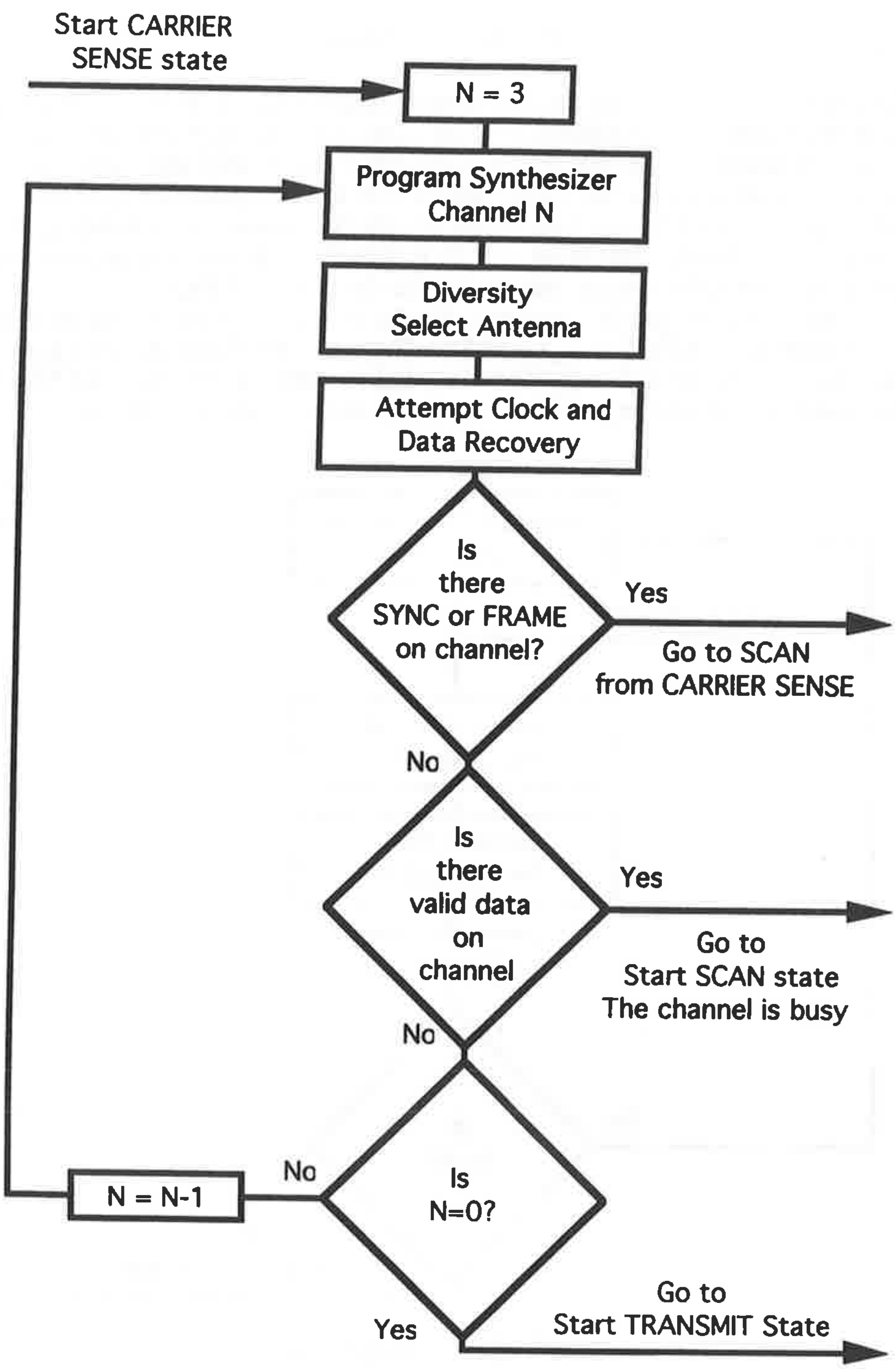
After the T_SCKR or T_RCKR period, the receiver must find FRAME. If frame is found, the receiver accepts the incoming B_PDU. If FRAME isn't found, the receiver insert an ERASURE DELAY. This delay simulates the time the receiver would have taken to receive a good B_PDU so that it remains in sync with the transmitting node. After the completion of the either of these two functions, the receiver increments the channel number and continues in sync with the transmitter until all five channels have been programmed. When the RECEIVE state is completed, the protocol state machine returns to SCAN.



RECEIVE State Machine
Figure 4.10

4.2.4 RF Hopping Protocol CARRIER SENSE State

As stated before, CARRIER SENSE is a state the protocol state machine enters before transmitting data. The procedure followed in the CARRIER SENSE state is used to determine if the media is busy prior to the start of TRANSMIT. The CARRIER SENSE state machine is very close in function to the SCAN state machine except for the fact that the first three channels are scanned in reverse order. As shown in figure 4.11, the CARRIER SENSE state machine uses the same time period for programming the synthesizer, T_PGM, diversity, T_DIV, and clock and data recovery, T_SCKR. If it detects the SYNC _FLD or FRM_FLD during T_SCKR, it will abandon the CARRIER SENSE state and move back into the SCAN state to attempt to recover the burst. If the clock and date recovery circuit recover clock and data but not SYNC or FRAME, the function will return a channel busy flag. This will indicate to the protocol state machine that the channel contained the middle of a B_PDU and that the protocol state machine should return to SCAN to attempt to synchronize on the next burst. If CARRIER SENSE doesn't find any channels busy, the protocol state machine advances to TRANSMIT.

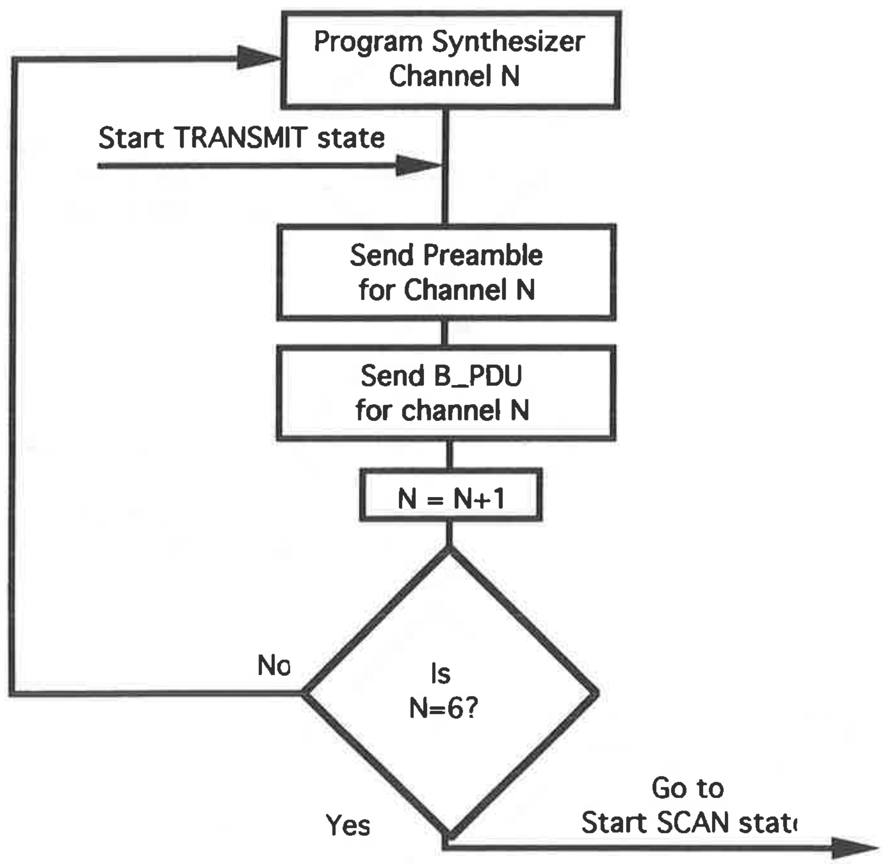


CARRIER SENSE State Machine
Figure 4.11

4.2.5 RF Hopping Protocol TRANSMIT State

The TRANMIT state machine is shown in figure 4.12. The TRANSMIT state machine is entered from the CARRIER SENSE state machine after channel 1 has been scanned. Since channel 1 is also the first channel the TRANSMIT state machine will send data out on, there is no need for the TRANSMITTER to reprogram the synthesizer, so it immediately begins to send the long preamble for the first burst. After sending the preamble and B_PDU for the first burst, the TRANSMIT state machine increments the channel count, programs the synthesizer and sends the next B_PDU.

As shown in figure 4.8, the preamble for the first three bursts consists of the long preamble time periods T_SYNC + T_FRAME. The next two channels use a short preamble as shown in figure 4.9 of T_SYNCS +T_FRAME. When the TRANSMIT state machine has finished transmitting all five bursts, it returns to the SCAN state.



TRANSMIT State Machine
Figure 4.12