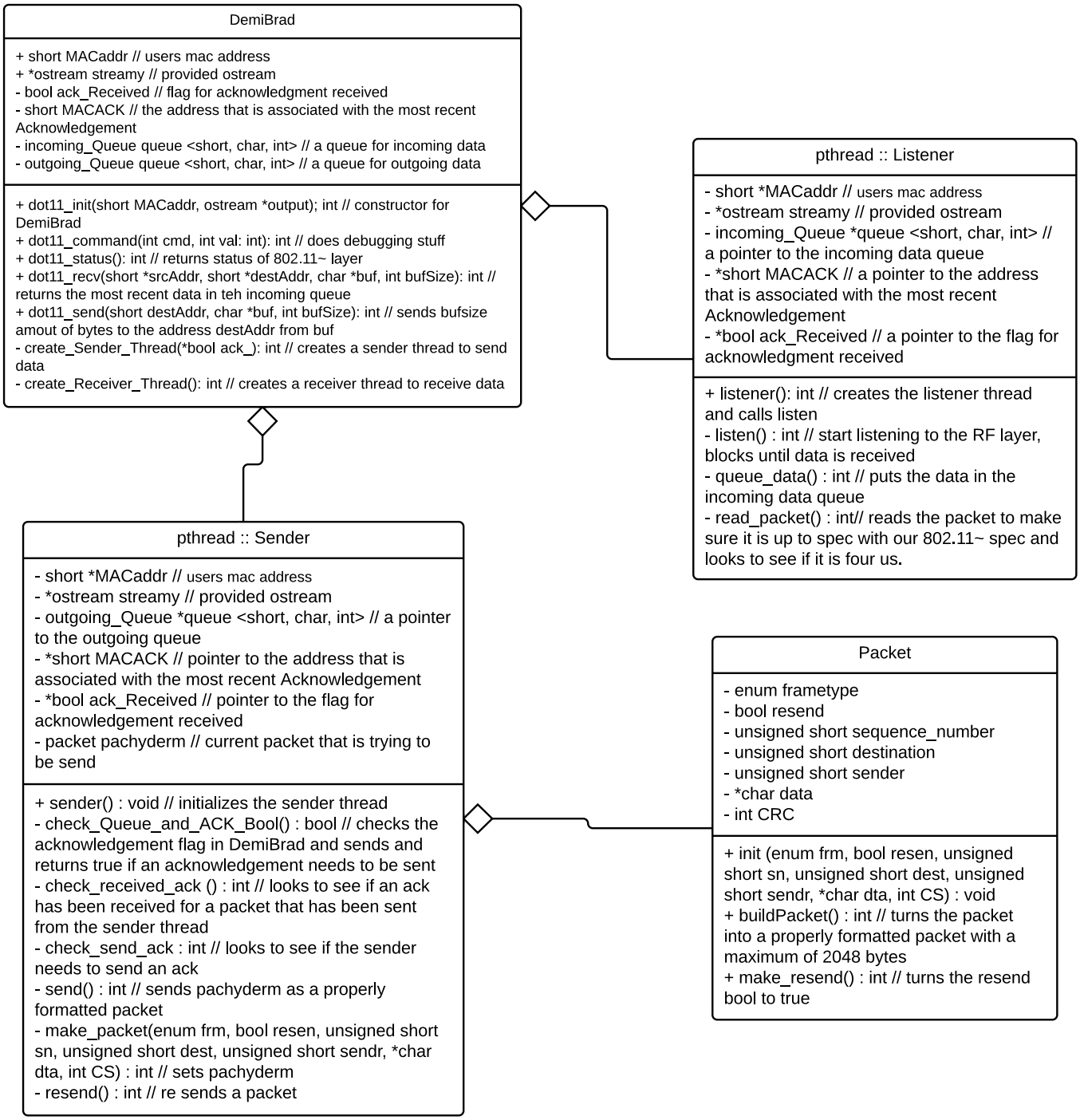
DemiBrad
+ short MACaddr // users mac address
+ *ostream streamy // provided ostream
- bool ack_Received // flag for acknowledgment received
- short MACACK // the address that is associated with the most recent Acknowledgement
- incoming_Queue queue <short, char, int> // a queue for incoming data
- outgoing_Queue queue <short, char, int> // a queue for outgoing data
+ dot11_init(short MACaddr, ostream *output); int // constructor for DemiBrad
+ dot11_command(int cmd, int val: int): int // does debugging stuff
+ dot11_status(): int // returns status of 802.11~ layer
+ dot11_recv(short *srcAddr, short *destAddr, char *buf, int bufSize): int // returns the most recent data in teh incoming queue
+ dot11_send(short destAddr, char *buf, int bufSize): int // sends bufsize amout of bytes to the address destAddr from buf
- create_Sender_Thread(*bool ack_): int // creates a sender thread to send data
- create_Receiver_Thread(): int // creates a receiver thread to receive data
pthread :: Listener
- short *MACaddr // users mac address
- *ostream streamy // provided ostream
- incoming_Queue *queue <short, char, int> // a pointer to the incoming data queue
- *short MACACK // a pointer to the address that is associated with the most recent Acknowledgement
- *bool ack_Received // a pointer to the flag for acknowledgment received
+ listener(): int // creates the listener thread and calls listen
- listen() : int // start listening to the RF layer, blocks until data is received
- queue_data() : int // puts the data in the incoming data queue
- read_packet() : int// reads the packet to make sure it is up to spec with our 802.11~ spec and looks to see if it is four us.
pthread :: Sender
- short *MACaddr // users mac address
- *ostream streamy // provided ostream
- outgoing_Queue *queue <short, char, int> // a pointer to the outgoing queue
- *short MACACK // pointer to the address that is associated with the most recent Acknowledgement
- *bool ack_Received // pointer to the flag for acknowledgement received
- packet pachyderm // current packet that is trying to be send
+ sender() : void // initializes the sender thread
- check_Queue_and_ACK_Bool() : bool // checks the acknowledgement flag in DemiBrad and sends and returns true if an acknowledgement needs to be sent
- check_received_ack () : int // looks to see if an ack has been received for a packet that has been sent from the sender thread
- check_send_ack : int // looks to see if the sender needs to send an ack
- send() : int // sends pachyderm as a properly formatted packet
- make_packet(enum frm, bool resen, unsigned short sn, unsigned short dest, unsigned short sendr, *char dta, int CS) : int // sets pachyderm
- resend() : int // re sends a packet
Packet
- enum frametype
- bool resend
- unsigned short sequence_number
- unsigned short destination
- unsigned short sender
- *char data
- int CRC
+ init (enum frm, bool resen, unsigned short sn, unsigned short dest, unsigned short sendr, *char dta, int CS) : void
+ buildPacket() : int // turns the packet into a properly formatted packet with a maximum of 2048 bytes
+ make_resend() : int // turns the resend bool to true