In Comms library:

boolean sendMessage(Message m) - send a new message (of any type), can be called anywhere

Packet[] packageMessage(Message m) - splits message into packets

Packet createPacket(byte[] content, Service sender, … )

boolean sendPacket(Packet p)

byte[] serialisePacket(Packet p) - including byte stuffing

void writePacketToSerial(byte[] p) - write packet to serial

void checkMessages() - read messages from message queue and process them, called in main loop

Message fetchMessageFromQueue() - pop first message off queue

void readMessage(Message m) - parse actual message contents

void readArduinoSerial() - triggered on interrupt, reads bytes from hardware serial and handles

packages

void writeToSoftwareSerial(byte b)

Packet getPacketFromSerial()

void writePacketToSerial(byte[] p) - cf. sendMessage()

void addPacketToQueue(byte[] p)

Packet deserialisePacket(byte[] p) - remove byte stuffing and convert to packet struct

Message buildMessage()

void addMessageToQueue(Message m)

void initialiseNode() - set up the node and look-up table and advertise services

In DCS library: (to be implemented later)

subscribe()

request()

unsubscribe()