dronePuppet

0.1

Generated by Doxygen 1.8.11

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

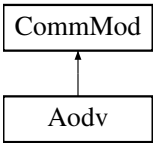
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 Aodv Class Reference

Inheritance diagram for Aodv:



Public Member Functions

- Aodv (Environment *, std::string, std::atomic_flag *, bool)

  Implementation of the AODV routing protocol.

Protected Member Functions

- void comm_function ()

  The main communications loop which handles incomming and outgoing messages.

Private Member Functions

- Aodv_rreq * create_hello ()
- Aodv_rreq * create_rreq (std::string, std::string, int)

  Helper method for creating route requests.
- Aodv_rrep * create_rrep (std::string, std::string, int)
- Aodv_rerr * create_rerr (std::string, int)

  Helper method for creating route errors.
- void process_rreq (Aodv_rreq *)

  Handle an RREQ that was received.
- void process_rrep (Aodv_rrep *)

  Handle an RREP that was received.
- void process_rerr (Aodv_rerr *)

Handle an RERR that was received.

- void process_data (std::string)

  Handle an data packet that was received.
- std::string get_attribute (std::string)

  Attribute extraction for AODV messages.
- bool have_route (std::string)

  Checks if a route to the destination is known.
- void add_route (std::string, int, int, std::string)

  Creates a route in the routing table.
- Aodv_rreq * deserialize_rreq (std::string)

  Deserialization for route requests.
- Aodv_rrep * deserialize_rrep (std::string)

  Deserialization for route replies.
- Aodv_rerr * deserialize_rerr (std::string)

  Deserialization for route errors.
- void log (std::string)

  Helper function to log internal information.
- void broadcast (std::string)

  Helper function to broadcast a message through the environment.

Private Attributes

- std::map< std::string, Aodv_route * > route_table

  Routing table for the communication module.
- std::string ip_address

  The IP address of the communication module.
- double HELLO_INTERVAL

  The interval at which hello messages are sent to discover nearby nodes.
- int SEQUENCE_NUMBER

  The AODV sequence number of the communication module.
- double ACTIVE_ROUTE_TIMEOUT

  The AODV active route timeout used to determine how long routes stay fresh for.
- double PATH_DISCOVERY_TIME

  How long to wait for relies to route requests.
- int BROADCAST_ID

  AODV broadcast ID used to prevent loops and ensure fresh information.
- int RANGE

  The amount of power used to broadcast messages.
- int TTL

  Default time to live for messages, dependent on network size.
- double last_hello

  The time at which the last hello message was sent.
- std::pair< std::string, std::string > **current_message**
- int state

  The internal state of the AODV implementation.
- std::atomic_flag * lock

  An atomic lock to regulate access to stdout.
- bool logging

  Switch to enable or disable logging.

Additional Inherited Members

3.1.1 Member Function Documentation

3.1.1.1 void Aodv::add_route (std::string *ip*, int *dest_seq*, int *hop_count*, std::string *next_hop*) [private]

Creates a route in the routing table.

Adds in the standard route timeout and ensures that all variables are set to prevent memory issues later

3.1.1.2 void Aodv::broadcast (std::string *message*) [private]

Helper function to broadcast a message through the environment.

Broadcast requires the coordinates of the broadcasting unit which can be unwieldy to do every time This process is simplified by wrapping it in a helper function

3.1.1.3 void Aodv::comm_function ( ) [protected], [virtual]

The main communications loop which handles incomming and outgoing messages.

Every loop we check to see if we should send a hello, based on the hello interval Next, any incomming messages are deserialized and handled appropriately If our messageable told us to shut down, the communications module exits Next, any outgoing messages are either immediately sent (if we have a route to the destination already) or a route request is distributed (if we do not have a route)

Implements CommMod.

3.1.1.4 Aodv_rreq * Aodv::create_hello ( ) [private]

Creates a special route request for a route to this node, with TTL 1

3.1.1.5 Aodv_rerr * Aodv::create_rerr (std::string *dst_ip*, int *ttl*) [private]

Helper method for creating route errors.

Abstracts away the destination sequence number lookup for convienence

3.1.1.6 Aodv_rrep * Aodv::create_rrep (std::string *dst_ip*, std::string *src_ip*, int *ttl*) [private]

Abstracts away next hop lookup for convienence

3.1.1.7 Aodv_rreq * Aodv::create_rreq (std::string *dst_ip*, std::string *src_ip*, int *ttl*) [private]

Helper method for creating route requests.

Abstracts away the destination sequence number lookup for convienence

3.1.1.8 Aodv_rerr * Aodv::deserialize_rerr (std::string *message*) [private]

Deserialization for route errors.

Takes a raw message string of a route error message and returns an AODV object representing that message

3.1.1.9 Aodv_rrep * Aodv::deserialize_rrep (std::string *message*) [private]

Deserialization for route replies.

Takes a raw message string of a route reply message and returns an AODV object representing that message

3.1.1.10 Aodv_rreq * Aodv::deserialize_rreq (std::string *message*) [private]

Deserialization for route requests.

Takes a raw message string of a route reuest message and returns an AODV object representing that message

3.1.1.11 std::string Aodv::get_attribute (std::string *message*) [private]

Attribute extraction for AODV messages.

Takes in a message string and returns the first field of that message (semicolon delimited)

3.1.1.12 bool Aodv::have_route (std::string *ip*) [private]

Checks if a route to the destination is known.

Determines if a route to the destination is known, and if that route is fresh

3.1.1.13 void Aodv::log (std::string *log_message*) [private]

Helper function to log internal information.

Makes use of the stdout lock we were passed on creation to make sure that only one thread is printing at any one time Also prints the ip of the communications module and the current environment time to help with debugging

3.1.1.14 void Aodv::process_data (std::string *message*) [private]

Handle an data packet that was received.

First we check if the message was for us, and if so we deliver the contents to our messageable If the message is destined for another node and we are specified as the next hop, then we forward that packet to the next hop according to our own routing table Note that messages destined for other nodes that we are not specified as a next hop for will be dropped

3.1.1.15 void Aodv::process_rerr (Aodv_rerr * message) [private]

Handle an RERR that was received.

This functionality os not yet implemented

3.1.1.16 void Aodv::process_rrep (Aodv_rrep * message) [private]

Handle an RREP that was received.

First we check if the message contains new information about other nodes, and if so we always update our routing table If the message was a response to a request we initiated, then we should send our data packet. If the message was a response to a request we forwarded for another node, then we should pass it on Note that nodes will only act as a middle hop for one message at a time (other reponses will be dropped)

3.1.1.17 void Aodv::process_rreq (Aodv_rreq * message) [private]

Handle an RREQ that was received.

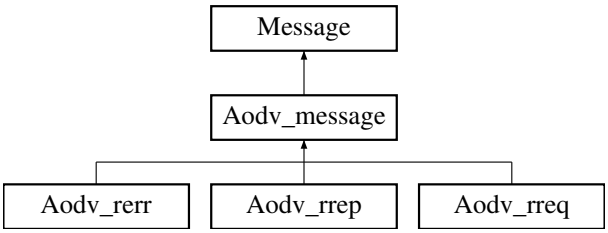
First we check if the message was a hello message, and if so we always respond to it If the message was a normal request, then we either answer it (if we have the info) or forward it if we do not Note that non-hello RREQs will be dropped if we are currently trying to send a message

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/communication/aodv/Aodv.hpp
- /home/will/Documents/effacious-octo-weasel/code/communication/aodv/Aodv.cpp

3.2 Aodv_message Class Reference

Inheritance diagram for Aodv_message:



Public Member Functions

- Aodv_message (std::string, int, int)
  Base class for all AODV message types.
- std::string get_dest_ip ()
  Getter method for the IP address of the destination node.
- std::string serialize ()
  Returns a string representation of the message.
- int get_dest_seq ()
  Getter method for the sequence number of the destination node.
- int get_ttl ()
  Getter method for the time to live.

Private Attributes

- std::string dest_ip

  The IP address of the detination node.
- int dest_seq

  The sequence number of the destination node.
- int ttl

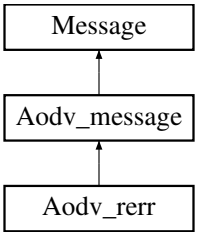
  The time to live of the message.

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/communication/aodv/Aodv_message.hpp
- /home/will/Documents/effacious-octo-weasel/code/communication/aodv/Aodv_message.cpp

3.3 Aodv_rerr Class Reference

Inheritance diagram for Aodv_rerr:



Public Member Functions

- Aodv_rerr (std::string dst_ip, int dst_seq, int ttl)

  AODV route error message.
- std::string to_string ()

  Returns a string representation of the message.

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/communication/aodv/Aodv_rerr.hpp
- /home/will/Documents/effacious-octo-weasel/code/communication/aodv/Aodv_rerr.cpp

3.4 Aodv_route Class Reference

Public Member Functions

- Aodv_route (int, int, std::string, int)

  Entry in ann AODV routing table.
- int get_seq ()

  Getter method for the sequence number of the route destination.
- int get_hop ()

  Getter method for the route hop count.
- std::string get_next_hop ()

  Getter method for the next hop on this route.
- int get_life ()

  Getter method for the route life time.

Private Attributes

- int dst_seq

  Sequence number of the route destination.
- int hop_count

  Number of hops required to get to the destination node.
- std::string next_hop

  Next hop on this route.
- int life_time

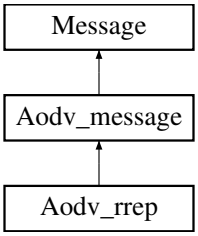
  Lifetime of this route.

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/communication/aodv/Aodv_route.hpp
- /home/will/Documents/effacious-octo-weasel/code/communication/aodv/Aodv_route.cpp

3.5 Aodv_rrep Class Reference

Inheritance diagram for Aodv_rrep:



Public Member Functions

- Aodv_rrep (int, std::string, std::string, std::string, int, int, int, std::string)

  Aodv route reply message.
- int get_hop_count ()

  Getter method for the hop count of the message.
- std::string get_source_ip ()

  Getter method for the IP address of the sending node.
- std::string to_string ()

  Returns a string representation of the message.
- int get_life_time ()

  Getter method for the life time of the route.
- std::string get_last_hop ()

  Getter method for te last hop on this route.
- std::string get_next_hop ()

  Getter method for the next hop on this route.

Private Attributes

- int m_hop_count

  Number of hops required to get from the source to the destination of the route.
- std::string m_source_ip

  IP address of the node at which the route terminates.
- int m_life_time

  Time for which this route is valid.
- std::string m_last_hop

  The last node which forwarded this reply.
- std::string m_next_hop

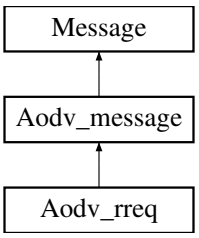
  The next node to which this reply will be forwarded.

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/communication/aodv/Aodv_rrep.hpp
- /home/will/Documents/effacious-octo-weasel/code/communication/aodv/Aodv_rrep.cpp

3.6 Aodv_rreq Class Reference

Inheritance diagram for Aodv_rreq:



Public Member Functions

- Aodv_rreq (int hop, std::string src_ip, std::string dst_ip, int src_seq, int dst_seq, int ttl)

  AODV route request message.
- int get_hop_count ()

  Getter method for the hop count of the message.
- std::string get_source_ip ()

  Getter method for the IP address of the sending node.
- std::string to_string ()

  Returns a string representation of the message.
- int get_source_seq ()

  Getter method for the sequence number of the sending node.

Private Attributes

- int hop_count

  The number of hops taken towards the destination.
- std::string source_ip

  The IP address of the sending node.
- int source_seq

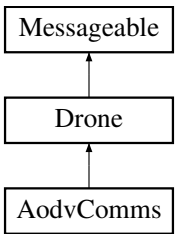
  The sequence number of the sending node.

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/communication/aodv/Aodv_rreq.hpp
- /home/will/Documents/effacious-octo-weasel/code/communication/aodv/Aodv_rreq.cpp

3.7 AodvComms Class Reference

Inheritance diagram for AodvComms:



Public Member Functions

- **AodvComms** (CommMod ∗, double, double, double, double, Environment ∗, int, int ∗, std::atomic_flag ∗)
- bool **message_callback** (Message ∗)
- void **run** ()

Private Attributes

- int **m_task**
- int ∗ **m_flag**
- std::atomic_flag ∗ **m_lock**
- Environment ∗ **m_env**

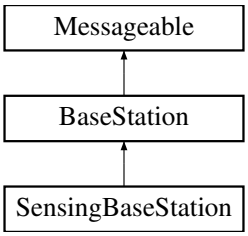
Additional Inherited Members

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/programs/AodvComms.hpp
- /home/will/Documents/effacious-octo-weasel/code/programs/AodvComms.cpp

3.8 BaseStation Class Reference

Inheritance diagram for BaseStation:



Public Member Functions

- **BaseStation** (CommMod ∗cm, double xp, double yp, double zp)

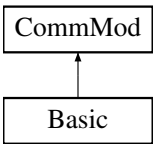
Additional Inherited Members

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/simulator/BaseStation.hpp
- /home/will/Documents/effacious-octo-weasel/code/simulator/BaseStation.cpp

3.9 Basic Class Reference

Inheritance diagram for Basic:



Public Member Functions

- Basic (Environment ∗, std::atomic_flag ∗)

  Basic messaging protocol for testing where nodes receive all messages sent if they are within range.

Protected Member Functions

- void comm_function ()

  The main communications loop which handles incomming and outgoing messages.

Private Member Functions

- void log (std::string)

  Helper function to log internal information.

Private Attributes

- double RANGE

  The amount of power used to broadcast messages.
- std::atomic_flag * lock

  An atomic lock to regulate access to stdout.

Additional Inherited Members

3.9.1 Member Function Documentation

3.9.1.1 void Basic::comm_function ( ) [protected], [virtual]

The main communications loop which handles incomming and outgoing messages.

Every loop we send any messages that are waiting to be sent Then we deliver any messages that we have received

Implements CommMod.

3.9.1.2 void Basic::log (std::string *log_message*) [private]

Helper function to log internal information.

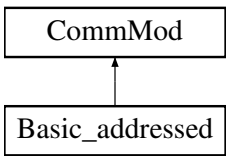
Makes use of the stdout lock we were passed on creation to make sure that only one thread is printing at any one time

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/communication/basic/Basic.hpp
- /home/will/Documents/effacious-octo-weasel/code/communication/basic/Basic.cpp

3.10 Basic_addressed Class Reference

Inheritance diagram for Basic_addressed:



Public Member Functions

- Basic_addressed (Environment *, std::atomic_flag *, std::string ip)

  Basic messaging protocol with addressing for testing, where nodes receive all messages sent if they are within range and addressed to them.

Protected Member Functions

- void comm_function ()

  The main communications loop which handles incomming and outgoing messages.

Private Member Functions

- void **log** (std::string)
- std::string **get_attribute** (std::string)

Private Attributes

- double RANGE

  The amount of power used to boradcast messages.
- std::atomic_flag * lock

  An atomic lock to regulate access to stdout.
- std::string ip_address

  The IP address of the communication module.

Additional Inherited Members

3.10.1 Member Function Documentation

3.10.1.1 void Basic_addressed::comm_function ( ) [protected], [virtual]

The main communications loop which handles incomming and outgoing messages.

Every loop any incomming messages are deserialized and delivered if they match our IP address Next, any outgoing messages are sent Note that received messages not addressed to this IP address will be dropped

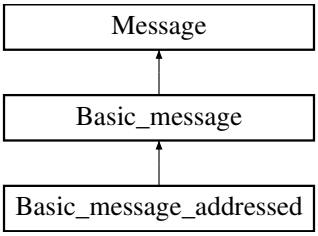
Implements CommMod.

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/communication/basic_addressed/Basic_addressed.hpp
- /home/will/Documents/effacious-octo-weasel/code/communication/basic_addressed/Basic_addressed.cpp

3.11 Basic_message Class Reference

Inheritance diagram for Basic_message:



Public Member Functions

- Basic_message (std::string)

  A basic message containing a payload wth no addressing information.
- std::string to_string ()

  Returns a string representation of the message.

Private Attributes

- std::string message

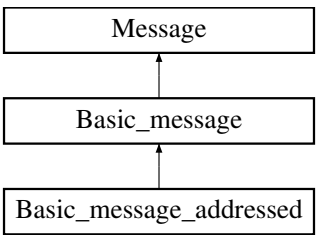
  The contents of the message.

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/communication/basic/Basic_message.hpp
- /home/will/Documents/effacious-octo-weasel/code/communication/basic/Basic_message.cpp

3.12 Basic_message_addressed Class Reference

Inheritance diagram for Basic_message_addressed:



Public Member Functions

- Basic_message_addressed (std::string, std::string, std::string)

  Basic message with addressing information.
- std::string to_string ()

  Returns a string representation of the message.
- std::string get_message ()

  Getter method for the message content.
- std::string get_destination ()

  Getter method for the IP address of the node this message is destined for.
- std::string get_source ()

  Getter method for the IP address of the node which sent this message.

Private Attributes

- std::string message

  The contents of the message.
- std::string destination

  The IP address of the messages destination.
- std::string source

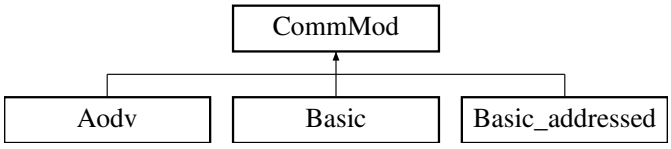
  The IP address of the node which sent the message.

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/communication/basic_addressed/Basic_message_↩ addressed.hpp
- /home/will/Documents/effacious-octo-weasel/code/communication/basic_addressed/Basic_message_↩ addressed.cpp

3.13 CommMod Class Reference

Inheritance diagram for CommMod:



Public Member Functions

- **CommMod** (Environment ∗env)
- void **set_messageable** (Messageable ∗msg)
- void **broadcast** (std::string message, double xPos, double yPos, double zPos, double range)
- void **broadcast** (Message ∗message, double xPos, double yPos, double zPos, double range)
- void **push_out_message** (Message ∗message)
- void **push_in_message** (std::string message)
- virtual void **comm_function** ()=0
- double **getTime** ()

Protected Attributes

- std::queue< Message * > **outQueue**
- std::queue< std::string > **inQueue**
- Environment * **environment**
- Messageable * **messageable**

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/simulator/CommMod.hpp
- /home/will/Documents/effacious-octo-weasel/code/simulator/CommMod.cpp

3.14 Coord Struct Reference

Public Attributes

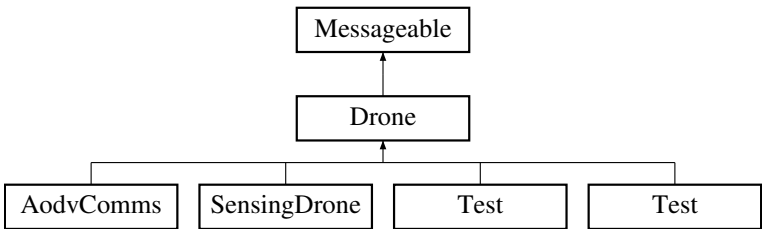
- double **x**
- double **y**
- double **z**

The documentation for this struct was generated from the following file:

- /home/will/Documents/effacious-octo-weasel/code/simulator/Messageable.hpp

3.15 Drone Class Reference

Inheritance diagram for Drone:



Public Member Functions

- **Drone** (CommMod *cm, double iX, double iY, double iZ, double maxSpeed, Environment *e)
- bool **isAlive** ()
- void **upkeep** ()

Protected Member Functions

- void **kill** ()
- void **turn** (double dAngle)
- void **move** (Direction direction, double speed, double distance)
- double **getMaxSpeed** ()
- double **getAngle** ()
- bool **hasFinishedMoving** ()
- double **sense** (std::string type)

Private Attributes

- double **oTime**
- bool **alive** = true
- Direction **dir**
- double **maxSpeed**
- double **ang** = 0
- Environment * **env**
- double **moveDR**
- double **moveSpd**

Additional Inherited Members

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/simulator/Drone.hpp
- /home/will/Documents/effacious-octo-weasel/code/simulator/Drone.cpp

3.16 Environment Class Reference

Public Member Functions

- **Environment** (std::map< std::string, data_type >, std::function< std::string(std::string)>, double timestep)
- **Environment** (std::map< std::string, data_type >, double timestep)
- void **broadcast** (std::string message, double xOrigin, double yOrigin, double zOrigin, double range, CommMod *)
- void **addData** (std::string type, data_type d)
- void **addDrone** (Drone *m)
- void **setBaseStation** (BaseStation *m)
- double **getData** (std::string type, double x, double y, double z)
- double **getTime** ()
- void **run** ()

Private Types

- typedef std::vector< std::vector< std::vector< double > > > **data_type**

Private Attributes

- double **timeElapsed**
- double **timeStep**
- BaseStation * **baseStation**
- std::vector< Drone * > **drones**
- std::map< std::string, data_type > **data**
- std::function< std::string(std::string)> **noiseFun**

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/simulator/Environment.hpp
- /home/will/Documents/effacious-octo-weasel/code/simulator/Environment.cpp

3.17 IpAllocator Class Reference

Public Member Functions

- **IpAllocator** (int, int, int, int)
- std::string **next** ()

Private Attributes

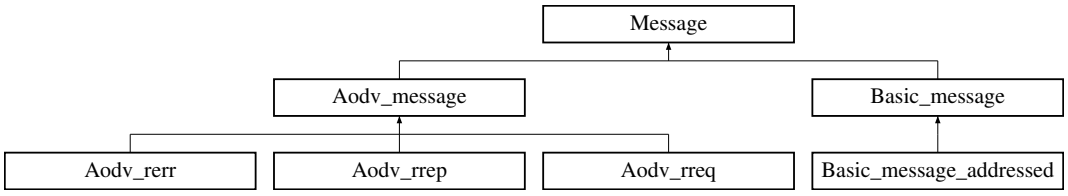
- int **m_first**
- int **m_second**
- int **m_third**
- int **m_fourth**

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/simulator/IpAllocator.hpp
- /home/will/Documents/effacious-octo-weasel/code/simulator/IpAllocator.cpp

3.18 Message Class Reference

Inheritance diagram for Message:



Public Member Functions

- **Message** (std::string type)
- time_t **get_current_time** ()
- virtual std::string **to_string** ()=0
- std::string **get_type** ()

Private Attributes

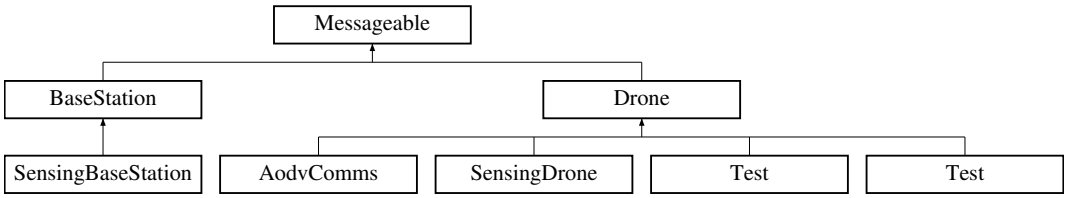
- std::string **type**

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/simulator/Message.hpp
- /home/will/Documents/effacious-octo-weasel/code/simulator/Message.cpp

3.19 Messageable Class Reference

Inheritance diagram for Messageable:



Public Member Functions

- **Messageable** (CommMod ∗cm, double xp, double yp, double zp)
- void **send_message** (Message ∗contents)
- Message ∗ **wait_for_message** ()
- void **push_message** (Message ∗contents)
- void **receive_message** (std::string contents)
- CommMod ∗ **get_comm_mod** ()
- double **getX** ()
- double **getY** ()
- double **getZ** ()
- Coord **getPosition** ()
- double **getTime** ()
- virtual bool **message_callback** (Message ∗message)=0
- virtual void **run** ()=0
- void **runCommMod** ()

Protected Attributes

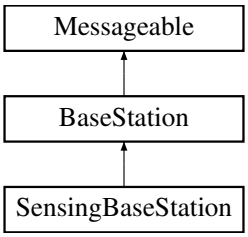
- std::queue< Message * > **inQueue**
- CommMod * **communicationsModule**
- Coord **position**

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/simulator/Messageable.hpp
- /home/will/Documents/effacious-octo-weasel/code/simulator/Messageable.cpp

3.20 SensingBaseStation Class Reference

Inheritance diagram for SensingBaseStation:



Public Member Functions

- **SensingBaseStation** (CommMod *cm, double xp, double yp, double zp, double areaX1, double areaY1, double areaX2, double areaY2)
- void **run** ()
- bool **message_callback** (Message *message)

Private Attributes

- std::vector< std::string > **droneIPs**
- double **areaX1**
- double **areaX2**
- double **areaY1**
- double **areaY2**

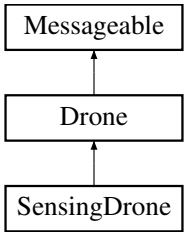
Additional Inherited Members

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/programs/SensingBaseStation.hpp
- /home/will/Documents/effacious-octo-weasel/code/programs/SensingBaseStation.cpp

3.21 SensingDrone Class Reference

Inheritance diagram for SensingDrone:



Public Member Functions

- **SensingDrone** (CommMod ∗, double, double, double, double, double, Environment ∗, bool)
- bool **message_callback** (Message ∗)
- void **run** ()
- void **continueJob** ()
- int **atLoc** (Coord location)
- void **newArea** (double x1, double y1, double x2, double y2, double height)

Private Member Functions

- void **quit** ()

Private Attributes

- int **m_task**
- int ∗ **m_flag**
- double **sensorRadius**
- std::queue< Coord > **remainingPoints**
- bool **sink_node**
- std::string **baseStationIP**

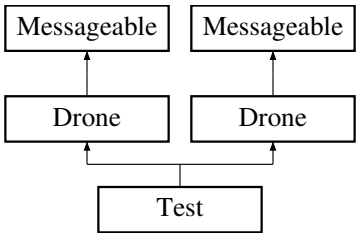
Additional Inherited Members

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/programs/SensingDrone.hpp
- /home/will/Documents/effacious-octo-weasel/code/programs/SensingDrone.cpp

3.22 Test Class Reference

Inheritance diagram for Test:



Public Member Functions

- **Test** (CommMod ∗, double, double, double, double, Environment ∗, bool)
- bool **message_callback** (Message ∗)
- void **run** ()
- **Test** (CommMod ∗, double, double, double, double, Environment ∗, bool)
- bool **message_callback** (Message ∗)
- void **run** ()

Private Attributes

- bool **sink_node**

Additional Inherited Members

The documentation for this class was generated from the following files:

- /home/will/Documents/effacious-octo-weasel/code/programs/BasicAddrComms.hpp
- /home/will/Documents/effacious-octo-weasel/code/programs/BasicComms.hpp
- /home/will/Documents/effacious-octo-weasel/code/programs/BasicAddrComms.cpp
- /home/will/Documents/effacious-octo-weasel/code/programs/BasicComms.cpp

Index