**KimonoNet  
Enabling Efficient and Reliable Inter-UAV   
Fluid Data Communication Link**

Protocol Specification Document

*Document*  
Version 1.0  
February 23, 2012

*Authors*  
Bollens, Eric <ebollens@oit.ucla.edu>   
Hung, James <james400mhz@yahoo.com>  
Khalapyan, Zorayr <zkhalapyan@oit.ucla.edu> Norris, Wade <wade.norris@ucla.edu>

**Table of Contents**

**TODO: Add section numbers once all sections are decided on**

Introduction 4

Overview 4

Terminology 4

Definitions 4

Routing 4

Greedy Perimeter Selection Routing (GPSR) 4

Greedy Forwarding 4

Perimeter Forwarding 4

Predictive Velocity in Neighbor Maintenance 5

Packet Format 5

General Packet Format 5

Composition 5

Common Header Format 5

Location Data Format 5

Vector Data Format 6

Type 0 Packet Format 6

Composition 6

Type 0 Header Format 6

Type 0 Extended Header 7

Type 2 & 3 Payload Format 7

Composition 7

Type 2 & 3 Header Format 7

Type 2 & 3 Neighbor Report Format 7

Composition 8

Packets 8

PKT-DATA (Type 0) 8

PKT-BEACON (Type 2) 9

PKT-BEACON-ACK (Type 3) 9

Structures 10

ROUTING-QUEUE 10

ROUTING-TABLE-1 10

ROUTING-TABLE-2 10

Timers 11

TMR-BEACON 11

Values 11

TMR-BEACON-VALUE 11

ROUTE-TABLE-REFRESH-VALUE 11

Functions 11

State Machine 11

# Introduction

## Overview

**TODO**

## Terminology

* NET – Network-layer protocol (often used synonymously with NIC in this document)
* NIC – Network interface card
* ULP – Upper-layer protocol

## Definitions

* FWD-GREEDY: Select via greedy mechanism the neighbor closest to destination for forwarding.
* FWD-PERIMETER: Use planar traversal to bypass when node cannot do greedy selection because it represents a local minima in greedy selection.
* PKT-DATA: Network packet with Type 0 that contains data to be routed.
* PKT-BEACON: Network packet with Type 2 that is sent on initialization and after a timer expires to allow neighbor nodes to maintain up-to-date neighborhood information.
* PKT-BEACON-ACK: Network packet with Type 3 that is sent upon receipt of any PKT-BEACON or PKT-BEACON-ACK that does not include the node in its neighbor report.

# Routing

## Greedy Perimeter Selection Routing (GPSR)

### Greedy Forwarding

When a packet is received in greedy mode:

* Node forwards received packet to neighbor closest to destination node.
* If node is closest of its known neighbors, then switch to perimeter forwarding.

**TODO: Please see Karp/Kung paper on the topic for now**

### Perimeter Forwarding

When a packet is received in or switched to perimeter mode...**TODO**

**TODO: Please see Karp/Kung paper on the topic for now**

## Predictive Velocity in Neighbor Maintenance

This extension of GPSR updates ROUTING-TABLE-1 that contains direct hop neighbors with information contained in ROUTING-TABLE-2 by factoring in velocity and time.

**TODO: Need to define this method**

# Packet Format

## General Packet Format

### Composition

All KimonoNet packets contain two components:

* Common Header
* Type-specific Header
* Type-specific Data

The common header type field HDR-TYPE defines the format of the type-specific header and data. The common header is always fixed width itself, and the type-specific header is fixed width for each given type, although the type-specific header varies between the types.

The full length of the header (common and type-specific combined) must be 4-byte word aligned. If it is not, then any extra bytes to achieve this alignment should be composed of 0x00.

### Common Header Format

The common header includes the following fields:

* HDR-MAGIC (2)
* HDR-VERSION (1)
* HDR-TYPE (1)
* HDR-SRC-ID (8)
* HDR-SRC-LOC (24)
* HDR-SRC-VEC (12)

The common header has a total length of 48 bytes.

### Location Data Format

Location data is represented as an object containing the following fields:

* LATITUDE (8)
* LONGITUDE (8)
* ACCURACY (4)
* TIME (4)

This data representation has a total length of 24 bytes.

### Vector Data Format

Vector data is represented as an object containing the following fields:

* SPEED (4)
* BEARING (8)

This data representation has a total length of 12 bytes.

## Type 0 Packet Format

### Composition

The Type 0 packet format begins with the common header.

Following the common header, Type 0 defines additional header fields for several purposes:

* Define the destination for packet routing via HDR-DST-ID and HDR-DST-LOC.
* Define whether the forwarding mode is FWD-GREEDY or FWD-DETOUR.
* Define the size of the variable-length data in octets (maximum data length is 2048 bytes).
* Provide a CRC-16 checksum of common and Type 0 header fields with HDR-HDR-CHK set to 0.

The total length of the Type 0 header format, including the common header, is 96 bytes.

### Type 0 Header Format

The Type 0 header includes the following fields:

* HDR-DST-ID (8)
* HDR-DST-LOC (24)
* HDR-FWD-DST-ID (8)
* HDR-FWD-MODE (1)
* HDR-DATA-LEN (2)
* HDR-HDR-CHK (2)
* HDR-QOS (1)
* HDR-PADDING (2)

### Type 0 Extended Header

If HDR-FWD-MODE provides additional fields needed for perimeter forwarding as defined by the GPSR paper:

* XHDR-ENTERED-LOC (24)
* XHDR-FACE-ENTERED-LOC (24)
* XHDR-FACE-FIRST-EDGE-SRC (24)
* XHDR-FACE-FIRST-EDGE-DST (24)

This extended header adds an overhead of 96 bytes. This could be heavily optimized in a production implementation, but is kept generic for this prototypical implementation.

These bytes are not included in the HDR-HDR-CHK.

## Type 2 & 3 Payload Format

### Composition

Following the common header, the Type 2 & 3 payload defines two additional header fields.

If HDR-NEIGHBOR-COUNT is greater than zero, then it has a non-zero data field with length as the multiple of HDR-NEIGHBOR-COUNT and the neighbor report packet length (44 bytes). To keep packet size below 1500 bytes, HDR-NEIGHBOR-COUNT has a maximum value of 34.

### Type 2 & 3 Header Format

The Type 2 &3 header includes the following fields:

* HDR-NEIGHBOR-COUNT (1)
* HDR-PKT-CHK (2)

The total length of the Type 2 & 3 header format, including the common header, is 52 bytes.

### Type 2 & 3 Neighbor Report Format

Following the header, Type 2 & 3 packets include HDR-NEIGHBOR-COUNT neighbor reports.

A neighbor report includes the following fields:

* NEIGHBOR-ID (8)
* NEIGHBOR-LOC (24)
* NEIGHBOR-VEC (12)

The total length of a neighbor report is 44 bytes.

# Composition

## Packets

All packets are sent via UDP broadcast, and all nodes listen for UDP broadcast.

Among packets received over UDP broadcast, a node acts based on HDR-TYPE:

* **Type 0 (Data)**: A node processes data packets if HDR-FWD-DST-ID is node ID. When HDR-DST-ID is also the node ID, then the data packet is passed to ULP. Otherwise, the routing algorithm determines the next HDR-FWD-DST and rebroadcasts the UDP packet again.
* **Type 2 (Beacon)**: A node processes all beacon packets. If the HDR-SRC-LOC timestamp is newer than the latest record in ROUTING-TABLE for the sender, then it updates the table. It also updates ROUTING-TABLE for any neighbor report with a newer NEIGHBOR-LOC timestamp. If no row for HDR-SRC-ID exists, then the node transmits a Type 3 (Beacon Acknowledgement) packet in response, including the new node in its neighbor report.
* **Type 3 (Beacon Acknowledgement)**: A node processes all beacon acknowledgement packets. It treats them identically to Type 2 packets with one exception: if it receives a Type 3 packet that contains its own node ID in a neighbor report, it does not send its own beacon acknowledgement. This prevents loops in neighbor discovery.

Further details on these packets follow forthwith.

### PKT-DATA (Type 0)

This packet contains data that the network routes through internal nodes until it reaches a destination.

When PKT-DATA arrives via UDP broadcast:

* If HDR-SRC-ID is the same as the node's ID:
  + Discard PKT-DATA.
* Else if HDR-FWD-DST-ID is not the same as the node's ID:
  + Discard PKT-DATA.
* Else if HDR-DST-ID is the same as node's ID:
  + Extract and pass to ULP.
* Else:
  + Pass to ROUTING-QUEUE.

This conceptually means that the node drops any packet that it itself originated or that it receives without being marked as the intended node to forward. Meanwhile, it extracts and asses any data packet to ULP if it is the final destination, or else it places the packet in the routing queue

### PKT-BEACON (Type 2)

This packet is used to deliver information about a node to all nodes within reception range. It also conveys information about any direct neighbors it has that it knows about.

This packet is sent under two conditions:

1. Initialization
2. Expiration of TMR-BEACON

Whenever PKT-BEACON is sent, TMR-BEACON is reset to TMR-BEACON-VALUE.

On initialization, PKT-BEACON will include no neighbor reports; in future iterations, the packet may contain up to 38 neighbor reports.

The expiration condition, meanwhile, ensures that either PKT-BEACON or PKT-BEACON-ACK is sent every TMR-BEACON-VALUE seconds. Because PKT-BEACON-ACK includes the same information as PKT-BEACON, it is not necessary to send PKT-BEACON if PKT-BEACON-ACK has just been sent.

When PKT-BEACON arrives via UDP broadcast:

* If HDR-SRC-ID is the same as the node's ID:
  + Discard PKT-BEACON and exit.
* If HDR-SRC-LOC is newer than ROUTING-TABLE[HDR-SRC-ID],
  + Update ROUTING-TABLE[HDR-SRC-ID] = HDR-SRC-LOC
* For each NEIGHBOR-REPORT:
  + If NEIGHBOR-LOC is newer than ROUTING-TABLE[NEIGHBOR-ID]
    - Update ROUTING-TABLE[NEIGHBOR-ID] = NEIGHBOR-LOC
* Create PKT-BEACON-ACK for node and send via UDP broadcast.
* Reset TMR-BEACON.

PKT-BEACON is never forwarded.

### PKT-BEACON-ACK (Type 3)

This packet is sent in response to reception of any PKT-BEACON or PKT-BEACON-ACK that does not include the node receiving PKT-BEACON-ACK in its neighbor report.

Whenever PKT-BEACON-ACK is sent, TMR-BEACON is reset to TMR-BEACON-VALUE. This ensures that only the minimum required number of PKT-BEACON packets are sent.

When PKT-BEACON arrives via UDP broadcast:

* If HDR-SRC-ID is the same as the node's ID:
  + Discard PKT-BEACON and exit.
* If HDR-SRC-LOC is newer than ROUTING-TABLE[HDR-SRC-ID],
  + Update ROUTING-TABLE[HDR-SRC-ID] = HDR-SRC-LOC
* For each NEIGHBOR-REPORT:
  + If NEIGHBOR-LOC is newer than ROUTING-TABLE[NEIGHBOR-ID]
    - Update ROUTING-TABLE[NEIGHBOR-ID] = NEIGHBOR-LOC
* If NEIGHBOR-REPORT did not include the node's ID:
  + Create PKT-BEACON-ACK for node and send via UDP broadcast.
  + Reset TMR-BEACON.

PKT-BEACON-ACK is never forwarded.

## Structures

### ROUTING-QUEUE

ROUTING-QUEUE contains PKT-DATA datagrams added because they had HDR-FWD-DST-ID corresponding to the node. This means that the previous relay intended this node as the next hop.

When a datagram is popped from ROUTING-QUEUE, the routing algorithm is invoked to determine the new forwarding destination address for PKT-DATA.

The routing algorithm considers three components:

* HDR-FWD-MODE: Whether the packet is currently in greedy or perimeter mode.
* HDR-DST-LOC: The intended final destination of the packet.
* ROUTING-TABLE-1: The table of known neighbors

Once this address has been determined, HDR-FWD-DST-ID is updated. In addition, HDR-FWD-MODE may also be updated, depending on if the routing algorithm has to switch forwarding mode.

### ROUTING-TABLE-1

This table contains all current neighbors through which DATA-PKT may be routed.

The values in this table are refreshed at an interval of ROUTING-TABLE-REFRESH-VALUE based on the position and velocity data contained within ROUTING-TABLE-2. This allows for nodes to be considered neighbors that have been learned about only from a neighbor report by a direct neighbor, if their location and velocity data indicates as such.

### ROUTING-TABLE-2

This table contains all known neighbors and all known neighbors of neighbors.

During a refresh of ROUTING-TABLE-1, the location, velocity and timestamps contained in this table are used to determine all direct neighbors. This allows neighbors of neighbors to be considered as neighbors if their location and velocity information indicates that they have moved into range.

All routes more than ROUTE-TABLE-EXPIRE-VALUE should be removed from this table before a refresh of ROUTING-TABLE-1 to ensure that routes do not become too stale.

## Timers

### TMR-BEACON

This timer tracks how long since the last PKT-BEACON or PKT-BEACON-ACK was sent.

If this timer expires, the node is responsible for sending a new PKT-BEACON with a priority equivalent to Control Data.

This timer is thus reset whenever PKT-BEACON or PKT-BEACON-ACK is sent.

The value that this timer is always set to is defined as TMR-BEACON-VALUE.

## Values

### TMR-BEACON-VALUE

Default value is **TO BE DEFINED**

### ROUTE-TABLE-REFRESH-VALUE

Default value is TMR-BEACON / 2 **TO DEFINE THIS BETTER**

**ROUTE-TABLE-EXPIRE-VALUE**

Default value is TMR-BEACON \* 4 **TO DEFINE THIS BETTER**

# Functions

**TODO**

# State Machine

**TODO**