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

Requirements Document

*Document*  
Version 1.0  
February 11, 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**

I. Introduction 3

I.1. Purpose 3

I.2. Scope 3

I.3. Definitions 3

I.4. References 4

II. Overall Description 4

II.1. Product Perspective 4

II.1.A. Background Information 4

II.1.B. Proposed Solution 4

II.1.C. Use Cases 4

II.2. User Characteristics 4

II.2.A. User Group A 4

II.2.B. User Group B 4

II.3. Constraints 4

II.3.A. High Churn 4

II.3.B. Mixed Horizons 5

III. Specific Requirements 5

III.1. System Requirements 5

III.1.A. Java Virtual Machine 5

III.2. Functional Requirements 5

III.2.A. Neighbor Discovery 5

III.2.B. Authentication Data Security 5

III.2.C. Peer Communication 5

III.2.D. Location and Velocity Routing 5

III.3. Performance Requirements 6

III.3.A. Control Data 6

III.3.B. Communication Data 6

III.3.C. Surveillance Data 6

III.3.D. Standard Data 6

III.4. Software System Attributes 6

III.4.A. Attribute A 6

IV. Appendix 7

# I. Introduction

## I.1. Purpose

The purpose of this document is to present a detailed description of the scope, perspective and requirements of the KimonoNet Project. The necessary tasks for completion of this project will be outlined throughout the document.

## I.2. Scope

This project addresses a peer-to-peer network research topic related to routing and transport control issues that arise in ad hoc communication between autonomous peers such as unmanned aerial vehicles (UAVs). Under such scenarios, limited external uplinks exist, and these uplinks may lie beyond the network horizon of an individual node; this motivates the need to establish routing across a fluid peer network. Traditional routing algorithms are not suited for such a topology given the high mobility and high churn of network peers and thus a rapidly shifting set of routes.

The project described herein will provide (1) a description of necessary protocols and algorithms and (2) an implementation of an ad hoc peer-to-peer network based on the above description that performs routing with consideration for the position and velocity of relays in order to improve efficiency and reliability.

The project involves a routing and transport implementation over OSI Layers 3 and 4. It will also provide a demonstrable application of the peer-to-peer network developed forthwith. In accomplishing this overlay, the project may leverage existing protocols such as IP and UDP.

## I.3. Definitions

* Unmanned Areal Vehicle (UAV) – UAVs are areal vehicles controlled by a remote pilot or autonomously by a computer system. These vehicles are very common in combat environments as they minimize the risks of a loss of life and allow for airborne collection of information and communication. In the context of this project we are only concerned with a collection of UAVs that will be able to communicate with each other allowing for the creation of an ad hoc wireless network. This can be especially useful in combat areas where stationary or grounded communication nodes will be vulnerable to enemy attack.
* Java Virtual Machine (JVM) – The Java Virtual Machine is a virtual machine capable of executing compiled Java code on a large variety of devices with varying hardware. By utilizing the JVM code can be written once and allow for installation and functionality on a wide variety of devices. By leveraging this we will guarantee our code maximizes compatibility with all possible nodes in a battlefield.
* Network Interface Card (NIC) – A Network Interface Card is a device that allows for connection of a computer to a communications network. For this project we will be discussing specifically wireless networks allowing for communication between UAVs.
* Quality of Service (QoC) – In military operations Quality of Service is defined to be………….

## I.4. References

# II. Overall Description

## II.1. Product Perspective

### II.1.A. Background Information

The military has significantly increased the computing power it places in the field in recent years; however, the increase in network coverage has not followed suit. One option to increasing coverage relies on the development of an ad hoc peer-to-peer network whereby network-enabled devices in the field may form routes from remote operational zones back to a secure base station.

Rather than providing the route via stationary relays that make easy targets, unmanned aerial vehicles (UAVs) and other mobile equipment already equipped with basic network capabilities may be configured to provide such a network.

However, a network of this type would likely form a very sparse graph with limited routes to any endpoint. Further, given changing positions of nodes in the network and possible calamities that might befall the nodes, the network would suffer high churn. Consequently, an implementation to cover this use case must both adapt quickly to topological changes and also provide reliable data delivery even when route reliability is tenuous.

Beyond military application, this sort of research topic may provide insight into broader ad hoc networking issues. To address the dynamics of ad hoc networks, some protocols including AODV and DSR have taken the approach of on-demand routing, ascertaining a path at send-time rather than predetermining routes. These suffer from disadvantages including inconsistent, unreliable data routing and extra control overhead. Other ad hoc protocols such as OSLR employ table-driven approaches, many reminiscent to traditional distance-vector protocols.

### II.1.B. Proposed Solution

This project shall address routing and transport in ad hoc peer-to-peer networks where constituents form a sparse, fluid network, prone to shifting network horizons due to both movement and failure. Traditional routing protocols do not adequately handle this, befallen to delays in convergence or a need to understand global network topology. In this void, several ad hoc protocols have been developed. Some use on-demand routing to ascertain a route at send time, while others use more traditional table-driven approaches for determining route. This research shall extend the table-driven ad hoc approach, introducing velocity as a predictive factor upon the distance-vector routing.

### II.1.C. Use Cases

## II.2. User Characteristics

### II.2.A. User Group A

### II.2.B. User Group B

## II.3. Constraints

### II.3.A. High Churn

Due to the nature of the environment the devices will be operating in the routing algorithm must be able to properly handle high churn. With the possibility in a military scenario of nodes going offline or being destroyed at any given point the algorithm must be able to handle a break in the network and reroute appropriately around this loss of a node.

Further with the constant movement of nodes in the network, frequently paths will become impossible due to nodes going out of range. This high churn rate must be taken into account and routes must be constantly adjusted based on the possibility of movement of the nodes in the network.

### II.3.B. Mixed Horizons

In a large network of UAVs many individual nodes will be out of range of each other. Due to this the network will have a large variety of mixed horizons within it. The implementation of this routing algorithm must take into account the constraint that many of these nodes will have varying horizons and inability to directly contact many nodes in the network. Due to this knowledge of the entire network will be impossible at an individual node and we must operate under the assumption that nodes may not be able to self-route date through the entire network. Therefore the source node cannot preplan the route and we must assume that data being sent will be assisted along by knowledge of neighboring peers.

# III. Specific Requirements

## III.1. System Requirements

### III.1.A. Java Virtual Machine

The implementation shall be programmed in Java to allow for running within a Java Virtual Machine (JVM). This will allow the implementation to be run on any device with a Network Interface Card (NIC).

## III.2. Functional Requirements

### III.2.A. Neighbor Discovery

The implementation shall allow for the discovery of neighboring nodes via a UDP broadcast and reciprocated message from peers in range. This initial discovery shall allow for unique identification of the node broadcasting and likewise of responding peers. Further during this initial communication the broadcasting node shall be able to acquire information about the location and velocity of the neighboring nodes to be used for routing later.

### III.2.B. Authentication Data Security

The implementation shall require authentication of peers and encryption of communication to prevent malicious interception of data or communication by outside agents. Since the environment these nodes will be operating in may be hostile we must prevent malicious use of the peer-to-peer network by unauthorized agents. If nodes are to join the network and communicate through it they shall be required to utilize a pre-shared private key for encryption of communication and data.

### III.2.C. Peer Communication

The implementation shall allow for communication between nodes in range of each other once the initial discovery has occurred. This shall allow for later routing of packets across multiple hops once nodes are able to communicate directly with each other when neighboring.

### III.2.D. Location and Velocity Routing

The routing algorithm shall allow for routing of communication across nodes in the network by utilizing the neighboring nodes’ location and velocity to predict possible routes and when they will become available. This shall allow for nodes across the network to communicate or send messages to a nearby uplink even if out or direct contact range.

To do so the communication may have to traverse several hops between many nodes in the network and the algorithm shall allow this to be done in an optimal way by predicting when nodes will be in range for communication and subsequent routing. This will be calculated using the known neighbors’ location and velocity to predict when nodes will leave the horizon of a given node and when others will come into range.

## III.3. Performance Requirements

The routing algorithm will be implemented with three distinct categories of Quality of Service (QoS) allowing for varying levels of performance.

### III.3.A. Control Data

Control data is used for communication of routing information and planning of routes for later communications. Since this is required to provide a good quality of transmission for other communications it is critical for the proper functioning of the algorithm and will be of the highest priority. This type of communication will be extremely time sensitive and therefore the implementation shall provide for very fast performance and timing for these types of communications.

### III.3.B. Communication Data

This data is not critical for continuing function of the algorithm and therefore is loss tolerant, however the performance of the algorithm hinges upon fast transmission of this type of communication. The implementation shall therefore provide for timely transmission of this type of communication. Since it is loss tolerant though there will be no need for duplication or alternate routing.

### III.3.C. Surveillance Data

This type of communication is not time sensitive, but is also not loss-tolerant. This data shall be protected from loss, and be duplicated into local buffer storage in the case that data is not successfully delivered to an endpoint.

### III.3.D. Standard Data

This type of communication includes data without specific QoS requirements. This data shall be routed with best-effort delivery. Data aggregation, duplication and alternate routing shall proceed according to prioritization at a specific node.

## III.4. Software System Attributes

### III.4.A. Attribute A

# IV. Appendix