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

Requirements Document

*Document*  
Version 1.0  
February 15, 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](#_Toc317105466)

[I.1. Purpose 3](#_Toc317105467)

[I.2. Scope 3](#_Toc317105468)

[I.3. Definitions 3](#_Toc317105469)

[I.4. References 4](#_Toc317105470)

[II. Overall Description 4](#_Toc317105471)

[II.1. Product Perspective 4](#_Toc317105472)

[II.1.A. Background Information 4](#_Toc317105473)

[II.1.B. Proposed Solution 4](#_Toc317105474)

[II.1.C. Use Cases 5](#_Toc317105475)

[II.2. User Characteristics 5](#_Toc317105476)

[II.2.A. Autonomous Nodes 5](#_Toc317105477)

[II.2.B. Communication Uplink 6](#_Toc317105478)

[II.2.C. Ground End Users 6](#_Toc317105479)

[II.3. Constraints 6](#_Toc317105480)

[II.3.A. High Churn 6](#_Toc317105481)

[II.3.B. Mixed Horizons 6](#_Toc317105482)

[III. Specific Requirements 7](#_Toc317105483)

[III.1. System Requirements 7](#_Toc317105484)

[III.1.A. Java Virtual Machine 7](#_Toc317105485)

[III.2. Functional Requirements 7](#_Toc317105486)

[III.2.A. Public Beaconing 7](#_Toc317105487)

[III.2.B. Beacon Acknowledging 7](#_Toc317105488)

[III.2.C. Peer Information Sharing 7](#_Toc317105489)

[III.2.D. Peer Communication 7](#_Toc317105490)

[III.2.E. Location and Velocity Routing 7](#_Toc317105491)

[III.3. Performance Requirements 8](#_Toc317105492)

[III.3.A. Control Data 8](#_Toc317105493)

[III.3.B. Communication Data 8](#_Toc317105494)

[III.3.C. Surveillance Data 8](#_Toc317105495)

[III.3.D. Standard Data 8](#_Toc317105496)

[III.4. Software System Attributes 8](#_Toc317105497)

[III.4.A. Attribute A 8](#_Toc317105498)

[IV. Appendix 9](#_Toc317105499)

# 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 the communication aspect of a group of these UAVs in order to create an ad hoc peer-peer network. This can be especially useful in combat areas where stationary or grounded communication nodes will be vulnerable to enemy attack and it is hard to extend coverage from centralized communication sources across an entire battlefield.
* 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 Java on a JVM the code can be written once and allow for installation and functionality on any device with a Network Interface Card (NIC). By leveraging this power the code will be guaranteed to maximize compatibility with all possible nodes in the network.
* 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 discussion of NICs will be specifically in reference to wireless network cards allowing for ad hoc communication between UAVs and centralized uplinks.
* Quality of Service (QoC) – In military operations Quality of Service (QoC) or Military Quality of Service (M-QoC) is defined as the prioritization of data delivery depending on its military value. In military scenarios there are many communications and depending on the type they have varying degrees of importance and therefore shall have varying degrees of priority over communication channels. Some of this data is extremely time sensitive such as control and communication data while other data such as surveillance data does not require the same urgency. At the same time there is a need for duplication to prevent the loss of surveillance data, while other data may be lost and later retransmitted.

## I.4. References

Eric I figured after all of the papers you had done this would be a good section for you to fill out based on which papers you found relevant enough to cite as a reference.

AODV

DSR

OSLR

# 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

The primary use case for this routing protocol will be in military operations where there is a high frequency of change in the network topology. In military operations UAVs will be on the move and can at any given point be shot down or destroyed. Due to this the algorithm must take heavy consideration on the possible dangerous environment the protocol will be used in and must not depend on the continuing existence of any given node in the network.

Further due to the sparse nature of nodes in this use case the algorithm must not assume a densely populated network of nodes. The majority of nodes in the network will most likely be out of range of each other and the routing protocol must leverage knowledge of other nodes’ locations and velocities to make transmissions when in range.

This routing protocol however may also extend beyond military operations and be applied to any network with high churn and known location and velocity of nodes. For example with the growing use of orbiting satellites with a very predictable location and travel pattern, this ad hoc communication protocol could very easily be extended to allow for faster transmission to areas that a passing satellite will be in range of sooner.

## II.2. User Characteristics

While there ultimately will be very little direct human interaction with the protocol the autonomous users will have distinctly different roles in interaction with the protocol. The most common user of this protocol will be the UAV or other autonomous nodes actively in the field. Additionally there will be ground units or end-users utilizing this overhead network for communication or transmission of data. Lastly there will be main communication uplinks which will be the portal to the outside world or to the main command base.

### II.2.A. Autonomous Nodes

The autonomous nodes in the network will be the most common type of user of this protocol. They will be predominantly comprised of UAVs or other autonomous devices allowing for connection between the ad hoc network. These devices will serve four main purposes in the network allowing for the proper and efficient functioning of the protocol.

The first function will be the transmission of data between nodes in the networks. This is where the knowledge of peers’ location and velocity will come into play and the node will use this information to route data arriving appropriately.

The next function of this user class is transmission of data on the network directly to a main communication uplink. Most traffic on the network will either be coming from a communication uplink or going to a communication uplink. If an autonomous node is in range of an uplink then getting transmissions destined to one will be one of its highest priorities.

On the opposite end of the communication system will be transmitting communications directly to ground units or receiving communication from them and sending them over the network for routing. Communication between the uplinks and the ground end-users depends on the right nodes in the network releasing this data to the ground units when in range or collecting messages from these end-users when over head.

The last function of this user class is data collection. The routing protocol will not be concerned with how this is done but many of these UAVs will be in the field to gather surveillance data and this information must be sent onto the network and routed to an uplink to be utilized appropriately. The routing protocol must therefore also accept information onto the communications network directly from the node itself.

### II.2.B. Communication Uplink

The communication uplinks will be the main source and sink of all communication on the network. This user class will be a special device with a direct connection to a control base or command center. All surveillance data will eventually need to be delivered to a communication uplink and all communications from ground end-users must be routed to an uplink for someone to receive it. Due to this the communication uplink must be treated appropriately and each node should be able to route to the closest uplink quickly and efficiently.

### II.2.C. Ground End Users

The final user class of this communication protocol will be the ground end users deployed in the field. This may vary from soldiers with the ability to communicate with UAVs overhead to stationary outpost communication devices. The algorithm will not be concerned with the implementation of these devices although it will need to know how to route communications to these end nodes from the communication uplink and vice versa.

## II.3. Constraints

### II.3.A. High Churn

Due to the nature of the environment the devices will be operating in the routing protocol 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 data 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. Public Beaconing

The implementation shall allow for neighboring nodes in range to discover a node by on a specified time interval sending out a UDP broadcast message. This shall include unique identifying information for this peer along with its vector information of geo-location and velocity. By doing this we will allow for a communication to be setup between peers that hear this message and can send an acknowledgement back with the same information to be used for later routing of other packets.

### III.2.B. Beacon Acknowledging

The implementation shall implement a listening system for nodes in the network that will listen for nearby nodes sending out discovery beacons via a UDP broadcast. The node shall store this information to be used for later routing. Further the node shall send an acknowledgement back to the source of the beacon sharing the listening node’s unique identifier and vector information. This will allow the connection of peers for a communication channel and later routing between the two.

### III.2.C. Peer Information Sharing

The implementation shall allow for sharing of neighboring peer information with other peers on a specified interval. This will allow other peers to gain knowledge of some peers outside of their own horizon and use this information for routing of later communications. Peers receiving this shared information can decide whether or not it will be relevant and whether or not it should be stored on their own however the protocol will operate under the assumption that peers will not have comprehensive knowledge of all nodes in the network.

### III.2.D. Peer Communication

The implementation shall allow for communication between nodes in range of each other once the initial discovery and acknowledgment has occurred. Further these nodes shall allow for the routing of packets between each other to an ultimate destination. Packets may be destined to an uplink or another grounded end-user however the nodes will be required to assist in the routing of neighboring peers’ communications onward to their ultimate end destination.

### III.2.E. 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 that specific node is out of 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.

Eric, maybe you can expand on this but otherwise I have no idea how the algorithm will work.

## 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

Do we have any specific attributes?

# IV. Appendix

Anything here?