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

Requirements Document

*Document*  
Version 1.1  
February 18, 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 in sparse networks of highly mobile ad hoc peers. This sort of scenario may exist, for example, when deploying unmanned aerial vehicles in an area of operations without pervasive Internet coverage. Under such a scenario, limited routes exist to a destination, and these may lie well beyond the network horizon of an individual node, motivating the need to establish routing across the peer-to-peer network. Traditional routing algorithms are not suited for this topology given the high mobility and high churn of its constituents, which cause routes to shift quickly.

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

The project involves of routing and transport over OSI Layers 3 and 4, and it may leverage existing protocols such as IP and UDP where useful. It will provide a demonstrable application of the peer-to-peer network developed forthwith.

## I.3. Definitions

* Unmanned Aerial Vehicle (UAV) – Aerial vehicles controlled by a remote pilot or computer system. These vehicles have become common in combat environments as they minimize loss of life and enable airborne information collection and communication. The context of this project considers only the communication aspect of UAVs, seeking to establish a fluid ad hoc peer-peer network across a set of these peers. This proves useful in combat areas where stationary or ground communication nodes will be vulnerable to enemy attack or where the area of operations presents challenges in extending existing coverage.
* Java Virtual Machine (JVM) – A portable runtime that executes the same Java code on numerous devices. By utilizing Java on a JVM, the code can be written once and used in heterogeneous environments.
* Network Interface Card (NIC) – A Network Interface Card provides a connection to a communications network. For this project, the discussion of NICs refers directly to wireless network cards that support for ad hoc communication.
* Quality of Service (QoS) – Quality of Service (QoS) classifies data delivery depending on its value. In military scenarios, service demands vary significantly: 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 but does require reliability.

## I.4. References

**TODO**

# 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 and other mobile equipment already equipped with basic network capabilities may provide such a network.

However, this type of network may 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 its nodes, the network may suffer from high churn. Consequently, an implementation must adapt quickly to topological changes and provide reasonably reliable service even when individual route reliability is very 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, a number of 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 reminiscent to traditional distance-vector protocols, but they struggle to adequately handle the rapidly changing nature of such a network without inordinate communication.

### II.1.B. Proposed Solution

To address the challenge of routing and transport in ad hoc peer-to-peer networks, this project will consider an array of existing routing strategies, including on-demand, link-state and distance vector protocols, to develop a basis for an approach to meet the conditions of this scenario.

Based on research to date, it is anticipated that it shall settle on a solution that extends Greedy Perimeter Stateless Routing (GPSR), first proposed by Karp and Kung, with neighborhood composition prediction based on velocity. This may reduce control traffic, as well as allow for a node to make more efficient choices about routing by holding onto data for a short period of time to reach a more direct route rather than selecting a circuitous one immediately.

### II.1.C. Use Cases

The motivating use case for this routing protocol is military operations with a high frequency of change in the network topology. In military operations, UAVs move rapidly and may also disappear from the network due to other failures. Consequently, the algorithm must not depend on the continuing existence of any given node in the network, but it may take advantage of prediction of topology changes based on known velocity.

Further due to the sparse nature of this node graph, the algorithm must operate efficiently given limited routing options. The majority of nodes in the network will likely be out of range of any individual node, and the routing protocol must leverage knowledge of other nodes’ locations and velocities to make transmissions when in range.

This routing protocol may also extend beyond military operations. It has applicability to any network with high churn where the location and velocity of nodes is generally known. For example, this ad hoc communication protocol could very easily be extended to support orbiting satellites or maritime expeditions. However, this algorithm will not likely expand to other scenarios where position is not readily available or velocity regularly changes unexpectedly, and in such situations, other approaches may be better suited.

## 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?