**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

I.1. Purpose 3

I.2. Scope 3

I.3. Definitions 3

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 5

II.2.A. Autonomous Nodes 5

II.2.B. End Points 5

II.3. Constraints 6

II.3.A. High Churn 6

II.3.B. Mixed Horizons 6

III. Specific Requirements 6

III.1. System Requirements 6

III.1.A. Java Virtual Machine 6

III.1.B. Network Interface Card 6

III.2. Functional Requirements 6

III.2.A. Initialization Beacon 6

III.2.B. Beacon Acknowledgement 6

III.2.C. Recurring Beacon 7

III.2.D. Peer Communication 7

III.2.E. Location and Velocity Routing 7

III.3. Performance Requirements 7

III.3.A. Control Data 7

III.3.B. Communication Data 8

III.3.C. Surveillance Data 8

III.3.D. Standard Data 8

# 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.

# 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

Two primary constituents exist in this ad hoc scenario: (1) autonomous nodes that send and forward packets, and (2) end points that receive the data packets from internal nodes

In the motivating scenario, unmanned aerial vehicles meet the former classification whereas command posts and other external uplinks satisfy the latter.

This project shall assume that the initial communication between these two user groups is sent from an internal node to an end point with a known location. This avoids the need to introduce a flooding search algorithm in addition to the routing and transport mechanism that this project shall present. However, in real world scenarios, such a search algorithm might prove necessary.

### II.2.A. Autonomous Nodes

The most common members of the network, autonomous nodes know their own position and velocity and have a NIC that supports ad hoc communication.

Autonomous nodes collect data or accomplish an objective and then seek further instructions. In order to transmit this data or receive further instructions, these nodes introduce data packets into the ad hoc network. These packets are addressed to an end point with known location, and the node forwards this packet to a peer based on the routing algorithm.

Further, beyond introducing packets into the network, autonomous nodes must also receive data packets passed to them by other nodes and then forward them on based on the routing algorithm.

These nodes are regarded as autonomous because they make independent decisions about position and velocity without considering its implications on routing.

### II.2.B. End Points

Under this scenario, a command post or other external uplink to the Internet serves as the destination endpoint (sink) of a packet originating within the ad hoc network.

This user class may receive communications from autonomous nodes with only transport functionality and not the routing algorithm; however, in order to reply to an autonomous node, it too must implement the routing algorithm. This is required not only for duplex communication, but also for acknowledgements in reliable communication.

As this project does not consider the mechanism to define the destination location, in the case of a reply to an autonomous node, this project assumes that the autonomous node will have the same position as when it sent the message. Given the velocity of autonomous nodes, this assumption is not completely adequate, but it focuses the scope of this project upon routing while allowing for further exploration at a later date.

## II.3. Constraints

### II.3.A. High Churn

Due to the operating environment, the routing protocol must handle high churn effectively.

Neighboring nodes each have a velocity that affects its position over time; consequently, the neighborhood for a given node may change swiftly and completely as nodes enter and recede from a given node's network horizon. The algorithm may attempt to handle this problem by considering 2-hop neighbors and extrapolating velocity over time to determine which nodes will be leaving range and which will be entering so as to help minimize required control traffic.

Further, given the motivating scenario, military operations also include the real possibility of nodes becoming unavailable due to failure or destruction. The algorithm presented must handle this independent influence on churn adequately, compensating for the fact that a route may become unavailable quickly and without warning.

### II.3.B. Mixed Horizons

In a sparse network, only a small set of nodes shall be in range of any individual node, and adjacent neighborhoods will only share a few nodes, if any. The implementation of this routing algorithm must take this into account, as even adjacent neighborhoods may not be able to communicate and a route must instead be established through other nodes to traverse the gap.

Consequently, it reasons that link-state broadcasts might require a significant number of packets to propagate global route information, by which time the routes may have changed due to mobility. This protocol must thus operate under the assumption that nodes may not be able to self-route data through the entire network, and instead that nodes have only knowledge of their local neighborhood.

# III. Specific Requirements

## III.1. System Requirements

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

The Java language system shall power this implementation, allowing for its deployment within any Java Virtual Machine environment.

### III.1.B. Network Interface Card

This implementation will require a network interface card that supports sending and receiving UDP broadcast.

## III.2. Functional Requirements

### III.2.A. Initialization Beacon

Upon initialization, each peer shall transmit a beacon packet that allows for discovery by neighboring nodes. This beacon shall employ UDP broadcast to provide a description of the node's identity, location and velocity vector.

Because this is during initialization, the beacon shall contain only information about the peer and not any neighbors. This is different from beacon acknowledgement (III.2.B) and recurring beacon (III.2.C) packets, which also include information about neighboring nodes.

### III.2.B. Beacon Acknowledgement

Upon receiving a beacon, a node shall transmit a beacon acknowledgement (handshake) also via UDP broadcast. This ensures that the node that initially beaconed knows of the existence of this other peer. Like a beacon, an acknowledgement also contains the identification, location and velocity vector of the peer and any neighbors it has already identifies.

In the event that a peer receives a beacon acknowledgement from a peer that it does not already have knowledge of, it shall store this information and then reply with a beacon acknowledgement of its own. If it already knows about the peer, it will store the updated information but not reply with another acknowledgement. This minimizes excess control signals and prevents loops.

### III.2.C. Recurring Beacon

At regular intervals, a node shall transmit another beacon. If the node knows of any other peers within its network horizon, the beacon shall also include the identification, position and velocity vector of these peers.

When a node receives a beacon that does not include information about itself, it will transmit a beacon acknowledgement (II.2.B). Otherwise, it will make no acknowledgement, and instead the sending peer will not get an update about this peer until it beacons itself.

### III.2.D. Peer Communication

This implementation shall provide a third packet format for conveying data. This data will have an intended recipient, and only the intended recipient shall receive and act on this data. In conjunction with the routing algorithm, this will allow for the packet to travel across the network to its ultimate destination, which is also included in the data packet.

In the event that reliable QoS is required, a beacon acknowledgement including the data identifier may prove necessary. Therefore, each data packet shall also include a sequence number.

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

The routing algorithm shall use its knowledge of the local neighborhood to determine which node to route the packet to. Based on Greedy Perimeter Stateless Routing (GPSR), this routing will occur either by greedy selection of the neighbor closest to the destination or, in the case that the node is a local maxima, then it will employ perimeter routing using the Relative Neighborhood Graph planarization to construct a detour route towards the destination. The aim is to use greedy routing as often as possible, but the sparse graph may require detours to reach the intended destination.

In order to simultaneously minimize control packets and ensure greedy selection whenever possible, the routing algorithm shall use not only location information, but also a predictive algorithm to determine if a neighbor of a neighbor, as described in the neighbor's beacon, has come into range since the beacon was sent.

Further, if time allows, the implementation will also consider holding onto a data packet for a relatively short period of time if a neighbor of a neighbor, as described in the neighbor's beacon, will soon come into range. This feature may require significant tuning to ensure optimality.

## III.3. Performance Requirements

The routing algorithm will be implemented with four distinct Quality of Service (QoS) grades.

### III.3.A. Control Data

Control data is critical to network management, such as beacons and acknowledgement. It must thus be transmitted without delay, and interpreted immediately upon reception.

Time sensitive data packet should instead be classified as communication data.

### III.3.B. Communication Data

While not critical for the proper functioning of the ad hoc network, these data packets are regarded as time sensitive and thus, with the exception of control data, communication data should be transmitted as soon as possible. While timeliness is required, such data shall not require reliable transport between nodes and must thus be loss tolerant. Any effort to ensure reliability of communication data should come from a further transport protocol.

### III.3.C. Surveillance Data

While not time sensitive, surveillance data should be provided lossless by the protocol. This means that the protocol may employ data duplication or require acknowledgements hop-by-hop to ensure that it is not flushed from the buffer of a peer until another peer has acknowledged reception and responsibility for its delivery.

### III.3.D. Standard Data

This data does not have any specified QoS requirements. It may be routed in parallel with surveillance data, but does not require acknowledgement.