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

*Document*  
Version 1.0.00  
March 14, 2012

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

*KimonoNet addresses a peer-to-peer network research topic related to routing and transport control issues in sparse networks of highly mobile ad hoc peers such as unmanned aerial vehicles. Traditional distance-vector and link-state algorithms are not suited for such a topology given the high mobility and churn of its constituents, which causes routes to shift too quickly for global route propagation. Therefore, this approach leverages a coordinate-based routing system proposed by Greedy Perimeter Stateless Routing and adds two-hop neighbor awareness to minimize control packets and improve reliability in fluid, sparse network graphs.*

# 1. Introduction

## 1.1. 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 through a remote operational zone back to a secure base station.

Rather than providing the route via stationary relays, which make easy targets, unmanned aerial vehicles and other mobile equipment already equipped with basic network capabilities may provide such a network in hostile environments.

The described 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 shall suffer from high churn. Consequently, an implementation must adapt quickly to topological changes and provide reasonably reliable service even when individual route exist only tenuously.

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 Ad hoc On-Demand Distance Vector Routing (AODV) and Dynamic Source Routing (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 the Optimized Link State Routing Protocol (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.

## 1.2. Objectives

This project seeks to provide a comprehensive description of the KimonoNet protocol including its algorithms, message formats, structures, states and functions. While this document summarizes its key principles, a full description of the protocol may be found in *KimonoNet: Protocol Specification Document* (Bollens, E., Hung, J., Khalapyan, Z., and W. Norris, 2012).

Further, this project also includes a KimonoNet client prototype running under the Java runtime. This prototype includes a production mode for field deployment and a test mode for simulation on a single host.

Finally, this project includes simulation results that measure the viability and effectiveness of KimonoNet through metrics including packet delivery, routing overhead, average latency and path optimality.

## 1.3. Scope

### 1.3.1. Network Layers

The KimonoNet protocol describes a mesh overlay optimally implemented directly atop Layer 2. However, its flexibility also supports Layer 3 and 4 implementations that leverage wireless broadcast or multicast.

For convenience, the prototype implementation of KimonoNet runs atop UDP multicast, with special provisions taken to support test mode, which can simulate multiple nodes under a single host environment.

### 1.3.2. Destination Locations

The KimonoNet protocol foregoes the ability to transmit to a mobile node, instead assuming that another mechanism will be used to locate the destination if it is mobile. This is a purposeful design decision to make the protocol deployable regardless of how the destination location is determined.

For simplicity, the implementation does not provide a search algorithm to locate a mobile destination, but instead assumes that the destination is fixed. A flooding search algorithm could be implemented in a future iteration so that a source can locate a mobile destination.

### 1.3.3. Data Payloads

This protocol provides an overlay for routing and transport across a fluid mesh network. As such, the protocol is ambivalent to the data it carries. It relies on the source to package this data and the destination to interpret this data. Consequently, it may transport application-layer data or lower-layer payloads tunneling across KimonoNet.

However, while KimonoNet supports tunneling, the implications of such are not considered forthwith and duplex communication may not be possible in the current iteration because it requires packets to have a known location as their destination, which is not viable in highly mobile mesh networks with a large diameter.

## 1.4. 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 predict topology changes based on two-hop knowledge.

Further due to the sparse nature of this node graph, the protocol 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 it must leverage knowledge of other node 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.

## 1.5. Use Cases

Two primary constituents exist in this scenario:

1. Autonomous nodes that send packets from ULP and route packets received from NET.
2. End points that receive data packets from NET and deliver packets to ULP.

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

This protocol assumes that the initial communication between these two user groups is sent from an autonomous node to an end point of known location. This avoids the need to introduce a flooding search algorithm in addition to the routing and transport mechanism that this protocol shall present. However, in real world scenarios, a search algorithm is anticipated to be necessary.

This protocol makes two additional simplifying assumptions: (1) it does not consider altitude, and (2) it assumes symmetric communication between any two nodes in the graph whereby, if a node can communicate with another node, the reverse is true as well.

### 1.5.1. Autonomous Nodes

The primary constituents of the network, autonomous nodes have (1) awareness of their position and velocity, (2) a NIC that supports ad hoc communication, (3) a network layer that supports broadcast, multicast or promiscuous packet delivery, and (4) a running instance of the KimonoNet client. Further, these nodes are regarded as autonomous because they make independent decisions about position and velocity without considering its implications on routing.

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 the known location of an endpoint (§2.5.2), and then node forwards this packet to a peer based on the routing algorithm (§3).

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.

### 1.5.2. 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. Because this protocol does not consider a mechanism to determine endpoint location, it instead requires that the originator knows the location of the end point and that the position of the endpoint does not change over time.

As with autonomous nodes, end points must also have (1) awareness of their position and velocity, (2) a NIC that supports ad hoc communication, (3) a network layer that supports broadcast, multicast or promiscuous packet delivery, and (4) a running instance of the KimonoNet client that minimally supports transport functionality.

Except for in the case of the reliable QoS classification (surveillance data), the end point may use location and velocity information to reply to an autonomous node by ascertaining its position. Because an autonomous node has variable position, all responses from an end point should have QoS classification of time-sensitive but loss-tolerant.

## 1.6. Constraints

### 1.6.1. High Churn

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

Each autonomous node has 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 network horizon. The routing algorithm handles this 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. However, this protocol acknowledges that the unexpected disappearance of a node may cause the loss of a packet, but it takes strides to minimize the likelihood of this and provides a reliable QoS classification that may be used at the cost of no timeliness guarantee on delivery.

### 1.6.2. Mixed Horizons

In a sparse network, only a small set of nodes is likely within in range of any individual node; further, adjacent neighborhoods will share only a few nodes, if any. The routing algorithm must thus 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.

This constraint is the reason that link-state management is non-optimal: by the time global route information has propagated, routes may have changed substantially due to mobility. This protocol thus operates under the assumption that nodes cannot self-route data through the entire network, and instead that nodes make decisions based solely on their local neighborhood.

## 1.7. Related Work

G. G. Finn proposed greedy route selection for coordinate-based routing in 1987, using a flood-based approach to resolve situations where greedy selection failed []. B. Karp and H. T. Kung reduced routing overhead by providing a method for detouring without flooding by way of graph planarization []. Their work, namely Greedy Perimeter Stateless Routing, serves as the basis for KimonoNet, which is then extended through two-hop awareness for velocity-based prediction of neighborhood changes.

# 2. PROTOCOL

## 2.1. Beacon Initialization

## 2.2. Beacon Acknowledgement

## 2.3. Beacon Updates

## 2.4. Predictive Neighbor Updates

## 2.5. Greedy Forwarding

## 2.6. Perimeter Forwarding

## 2.7. Quality-of-Service

# 3. Implementation

# 4. Simulation

# 5. ConclusionS

## 5.1. Outcomes

## 5.2. Future Work

Provide an additional protocol for finding the location of a mobile node

QoS to include reliability

Hold packets for short duration if greedy node is coming into range

# 6. Acknowledgements

We thank G. Pau, a mentor who advised us throughout this process, which itself was an outcome of his CS 114: Peer-to-Peer Networks class at UCLA.

Further, we thank B. Karp and H. T. Kung for their work on Greedy Perimeter Stateless Routing [], as well as G. G. Finn, who first proposed greedy routing through a coordinate-based system [].

# 7. REFERENCES

[] Finn, G. G. Routing and addressing problems in large metropolitan-scale internetworks. Tech. Rep. ISI/RR-87-180, Information Sciences Institute, Mar. 1987.

[] Karp, B. Geographic Routing for Wireless Networks. Ph.D. Dissertation, Harvard University, Cambridge, MA., Oct. 2000.

[] Karp, B., Kung, H. T. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks, in Proceedings of the sixth ACM/IEEE International Conference on Mobile Computing and Networking (MOBICON 2000), Boston, MA, Aug. 2000, pp. 243-254.