OpenFlow-based Adaptive Routing for Wireless Networks

Alok Kulkarni
akulkar4@ncsu.edu

Angelyn Arputha Babu John
ababujo@ncsu.edu

Jignesh Darji
jndarji@ncsu.edu

Nishad Sabnis
nsabnis@ncsu.edu

`https://sites.google.com/a/ncsu.edu/openflow-based-adaptive-routing-for-wireless-networks`
October 4th, 2015

1 Introduction

1.1 Problem Statement

Designing a system to enable adaptive routing in a wireless network in order to make a comparative analysis of the throughput efficiency of ground nodes versus aerial nodes.

1.2 Problem Description

We aim to implement OpenFlow based adaptive routing in an ad-hoc network by monitoring the link quality between wireless nodes. We anticipate that in such a network which offers multiple wireless routes between two end-points, the fluctuations in the RF link qualities between the endpoints will play an important role in determining the best end to end path. Determining the wireless link quality between each and every inter-connected node and making routing decisions based on this information constitute the two major parts of the problem.

We plan to make aerial nodes a part of the network which will be used for testing. Aerial nodes have their own set of advantages and disadvantages. They are less susceptible to electromagnetic interferences and can beam wifi over a large area if the antennae are powerful enough. However, the number of aerial nodes and naturally the number of available links through such nodes is likely to be lesser due to the low prevalence of such nodes. These trade offs need to be accounted for while making the routing decisions as well. The final aim is to ensure that the flow tables are dynamically modified to ensure effective end to end packet transmission.

2 Components

2.1 Platforms for the project

- CentMesh
- OpenFlow v1.3
- OpenDaylight
- Ubuntu

2.2 Areas for the project

- Link Quality Monitoring
- Adaptive Routing
- Quality of Service

2.3 Major Components

2.3.1 Wireless Ground Nodes

The ground nodes will be movable CentMesh carts. They will have the following features:

- At least one wireless interface
- Open vSwitch module installed
- One of the ground nodes will be the controller

2.3.2 Aerial Nodes

The aerial nodes will have the capacity to go up till 30 feet and beem signals from above. The key features of the aerial nodes are:

- At least one wireless interface
- BeagleBone black Linux boards
- Open vSwitch kernel module installed

2.3.3 Software Components

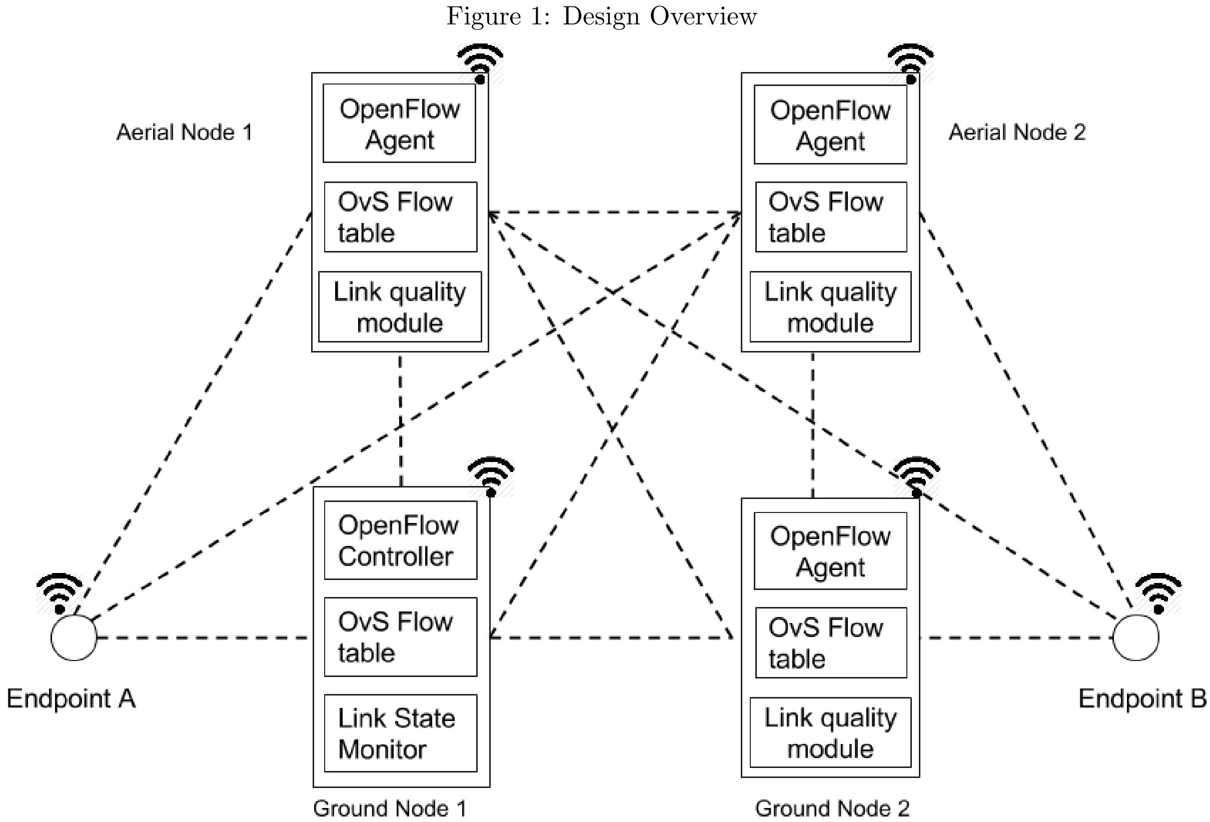
The software components will help determine the link quality and the optimum path, and they will configure the network with the optimal path.

- Link Quality Information module
- OpenFlow Control module

3 Design

3.1 Overview

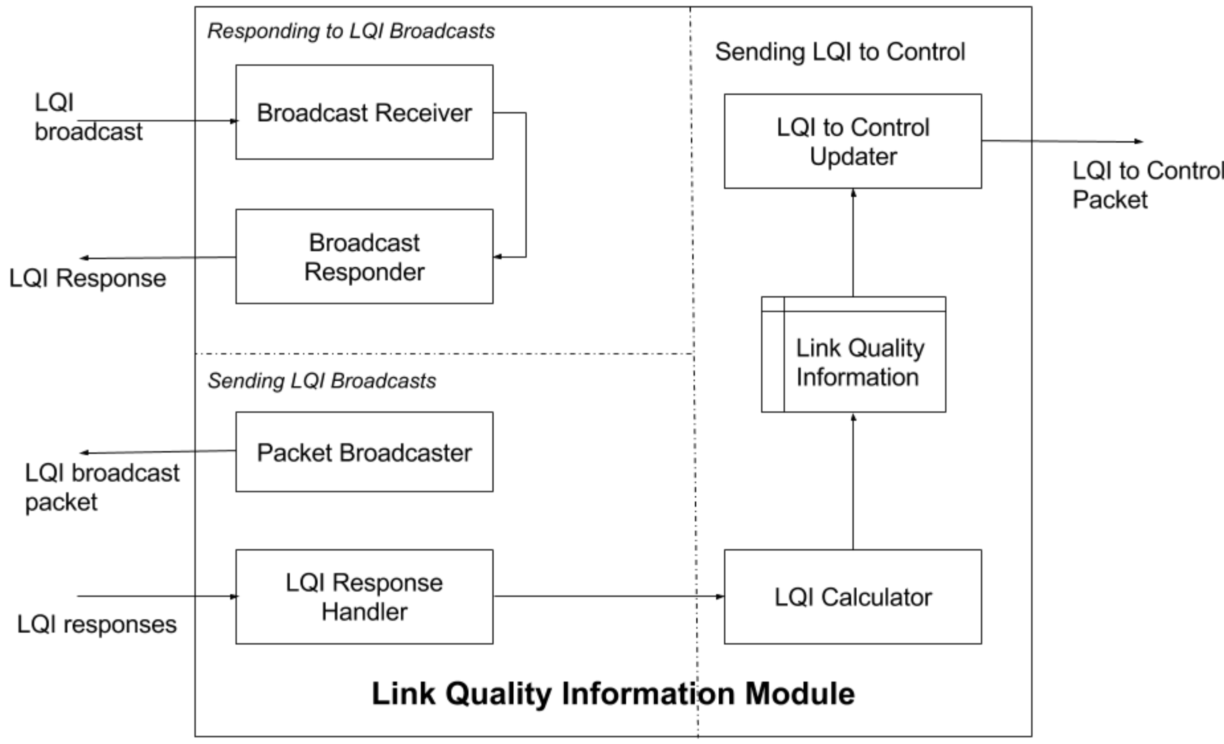
Figure 1: Design Overview



The above figure describes the overview of the components constituting this system. There exists a wireless ad-hoc network of aerial and ground nodes. One of the nodes acts as an SDN controller and the others act as agents. A link quality information (LQI) module is running on all the nodes in the network and this information is forwarded to the OpenFlow controller which makes adaptive routing decisions. The controller will use this information to compute the optimum end-to-end route between the endpoints. These routes will then be configured into the nodes using OpenFlow. The nodes will have OpenFlow agent running on them which will configure the routes sent by the Controller.

3.2 Link Quality Information Module

Figure 2: Components in the Link Quality Information Module



Packet Broadcaster
The packet broadcaster will broadcast packets at a regular interval to initiate the neighbour discovery.

LQI Response Handler
The LQI Response handler will wait for the responses to the broadcast packets sent by the Packet Broadcaster. It will then forward these responses to the LQI calculator.

Broadcast Receiver
The broadcast receiver will receive the broadcast packets sent by the neighboring node LQI modules.

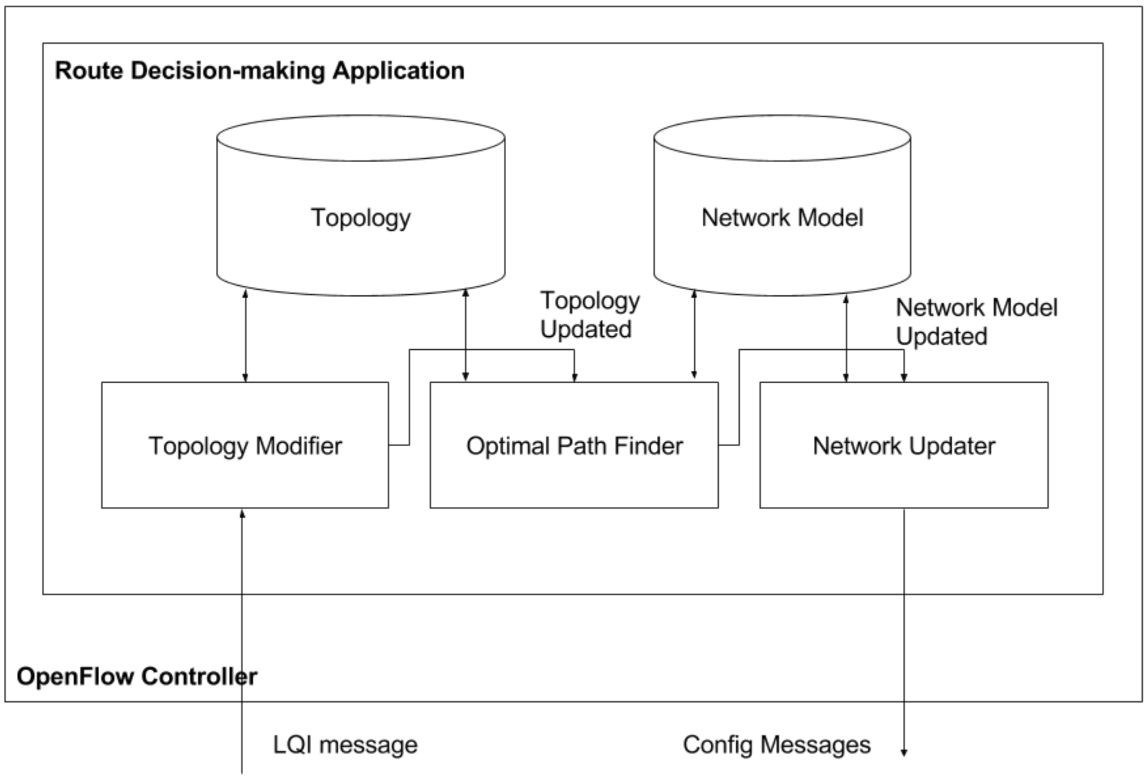
Broadcast Responder
Upon receiving the broadcast packets from the other LQI modules, the broadcast responder will send a response to the appropriate nodes from where it received the broadcast packet. The response will be such that the other side will appropriately be able to establish the link quality.

LQI Calculator
The LQI Calculator will assimilate all the responses from the neighbouring nodes and update the link quality information table.

LQI to Control Updater
The LQI to Control Updater gets the calculated Link Quality Information from the LQI Calculator. It will send this information over to the controller.

3.3 OpenFlow Control Module

Figure 3: Components in the OpenFlow Control application



Topology Modifier
The LQI packets received by the controller will be send to this module to generate/update the network topology. The topology will consists of the nodes and the link costs associated between each nodes. If theres is a modification in the topology, the Optimal Path Finder module will be notified to update the routes.

Optimal Path Finder
This module will keep a snapshot of the Network Model and compute the new model from the updated Topology. The new model will be compared to the previous snapshot to detect changes. If the model has been modified, it will intimate the Network Updater to configure these changes in Network.

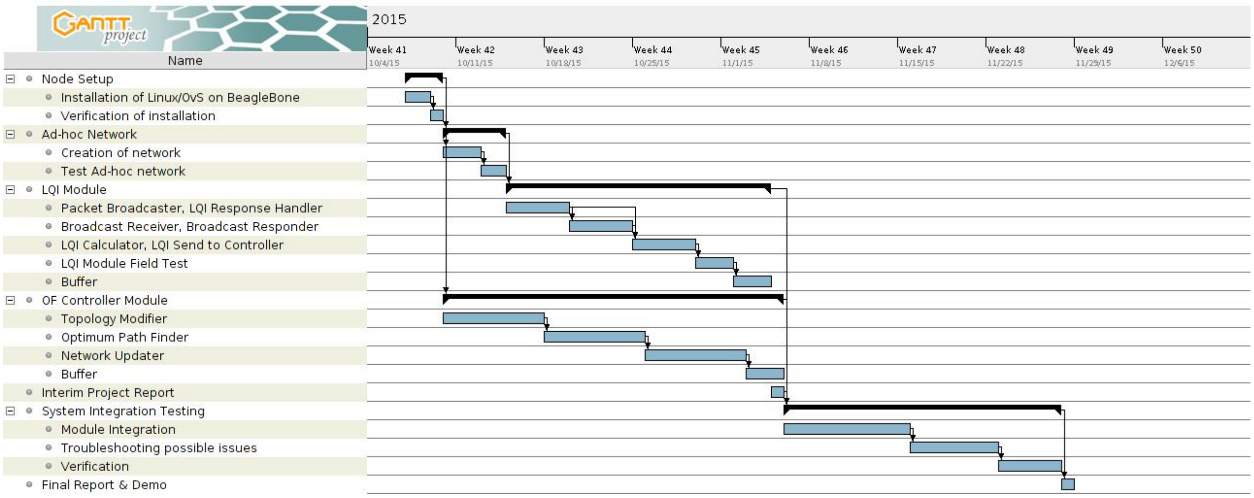
Network Updater
If the Network Updater receives a call to configure the changes in the network, it will compute the changes from the previous snapshot and then configure the routes that have been changed to appropriate nodes.

4 Per-member Responsibilities

Tasks	**Angelyn Arputha Babu John**	**Jignesh Darji**	**Nishad Sabnis**	**Alok Kulkarni**
Node Setup	Implement	Implement	Implement	Implement
Creation of ad-hoc network	Review	Review	Implement	Implement
LQI Message Handling	Review	Review	Implement	Implement
LQI Calculator	Review	Review	Review	Implement
LQI to Control Updater	Review	Review	Implement	Review
Topology Modifier	Implement	Implement	Review	Review
Optimal Path Finder	Implement	Implement	Review	Review
Network Updater	Implement	Implement	Review	Review

5 Timeline

Figure 4: Project Timeline



Name	**Begin date**	**End date**	**Duration (Days)**
Node Setup	10/7/15	10/9/15	3
-Installation of Linux/OvS on BeagleBone	10/7/15	10/8/15	2
-Verification of installation	10/9/15	10/9/15	1
Ad-hoc Network	10/10/15	10/14/15	5
-Creation of network	10/10/15	10/12/15	3
-Test Ad-hoc network	10/13/15	10/14/15	2
LQI Module	10/15/15	11/4/15	21
-Packet Broadcaster, LQI Response Handler	10/15/15	10/19/15	5
-Broadcast Receiver, Broadcast Responder	10/20/15	10/24/15	5
-LQI Calculator, LQI Send to Controller	10/25/15	10/29/15	5
-LQI Module Field Test	10/30/15	11/1/15	3
-Buffer	11/2/15	11/4/15	3
OF Controller Module	10/10/15	11/5/15	27
-Topology Modifier	10/10/15	10/17/15	8
-Optimum Path Finder	10/18/15	10/25/15	8
-Network Updater	10/26/15	11/2/15	8
-Buffer	11/3/15	11/5/15	3
Interim Project Report	11/5/15	11/5/15	1
System Integration Testing	11/6/15	11/27/15	22
-Module Integration	11/6/15	11/15/15	10
-Troubleshooting possible issues	11/16/15	11/22/15	7
-Verification	11/23/15	11/27/15	5
Final Report and Demo	11/28/15	11/28/15	1

6 Test Plan

Test	**LQI Module**	**Topology Modifier(TM)**	**Optimal Path Finder (OPF)**	**Network Updater(NU)**
Add a node with low path cost	Connected nodes should transmit cost with the new node to the LQI messages	Add node to Topology; intimate OPF	Add to Network Model; intimate NU	Update affected nodes
Add a node with high path cost	Connected nodes should transmit cost with the new node to the LQI messages	Add node to Topology; intimate OPF	Network Model remains same	NA
Remove node from Network Model	LQI messages will not contain costs to this node	Remove node from Topology; intimate OPF	Modifies Network Model; intimate NU	Update affected nodes
Remove node not part of Network Model	LQI messages will not contain costs to this node	Remove node from Topology; intimate OPF	Network Model remains same	NA
Increase associated path cost of a node in Network Model	Cost to this node should be increased in the LQI messages	Update path costs in Topology; intimate OPF	Modifies Network Model; intimate NU	Update affected nodes
Increase associated path cost of a node NOT in Network Model	Cost to this node should be increased in the LQI messages	Update path costs in Topology; intimate OPF	Network Model remains same	NA
Decrease associated path cost of a node in Network Model	Cost to this node should be decreased in the LQI messages	Update path costs in Topology; intimate OPF	Network Model remains same	NA
Decrease associated path cost of a node NOT in Network Model	Cost to this node should be decreased in the LQI messages	Update path costs in Topology; intimate OPF	Modifies Network Model; intimate NU	Update affected nodes

7 Demo Plan

1. Once the ad-hoc network is setup, the controller will generate the Network Model and configure the rules in all the nodes for the optimal path.
2. A display utility for the Network Model will print the current Network Model.
3. We'll display that these rules have been correctly configured in the nodes by connecting to each node and displaying their rule tables.
4. Thus, on changing the distance between the nodes, we plan to show that the routes between the two endpoints change dynamically based on the link strength.