CS 620 Project Update

Group 7 (Rohan, Anuj, Arun)

Topic: Implementing a publish-subscribe architecture without using IP layer and above for the data path.

Current Status:

1. We have read and understood the OpenFlow specification (http://www.openflow.org) for use as a Software-Defined-Networking (SDN) solution to our network architecture.
2. We wish to use MPLS with OpenFlow in order to provide a layer 2 (or 2.5) data-path. However, MPLS functions were added only in OpenFlow v1.1, and most available implementations of OpenFlow switches and controllers support only v1.0, which did not have the MPLS functions.
3. We have identified the key components that will be our deliverables, namely:
    - An OpenFlow switch which can handle rules for pushing/popping MPLS labels. We have found an experimental implementation in one of the development branches and got it working for MPLS rules.
    - An OpenFlow controller which can communicate with this switch for adding/removing MPLS-related rules. In this regard, we did not find any way of making currently available OpenFlow controllers work with the experimental v1.1 switch. So, we are going to use a workaround for interfacing with the control-tool provided by the experimental v1.1 switch directly.
    - End-hosts to simulate publish-subscribe produces/consumers. Implementation is remaining, but this is the easiest part.
    - A mechanism for mapping pub/sub flows to MPLS labels: We have thought of an approach in which we can overload the destination port field of the TCP header which the switches can use to identify an event (in coordination with the controller, of course) and accordingly add the required MPLS label.

In summary, our immediate next step is to implement the OpenFlow controller which can accept requests from publishers and subscribers, maintain this state, and accordingly send rules for pushing/popping MPLS labels to the compliant-switches.

We will be simulating the entire system using mininet, which is a virtual network capable of running on a desktop computer. We have already got this simulator working for our initial tests with the experimental v1.1 switch.