CS838 HW #3: OpenFlow

In this assignment, you will continue your work on OpenFlow by extending the controller to interact with middle boxes. You must write a component which checks the health state of one or many middle boxes and load balances traffic through the nearest middle box. Any traffic sent from end hosts must go through this middle boxes if it online. If there are no middle boxes available, traffic should be delivered directly to the destination.

The middle boxes are actually OpenFlow switches. You will be provided with an input file with their DataPath id, this will help you send messages to them as well as to test their health states.

TESTBED:

You will be provided with a new configuration file and support files for a virtual test bed (similar to those used in HW #1). This test bed will contain 3 switches, 2 end hosts, and 2 middle boxes. You can get a package containing the topology file and virtual machines at www.cs.wisc.edu/~akella/CS838/F09/hw3/vms/environment.tar.gz. You should read the README file as some changes have been made to the script, had.disk, and OpenFlow kernel module.

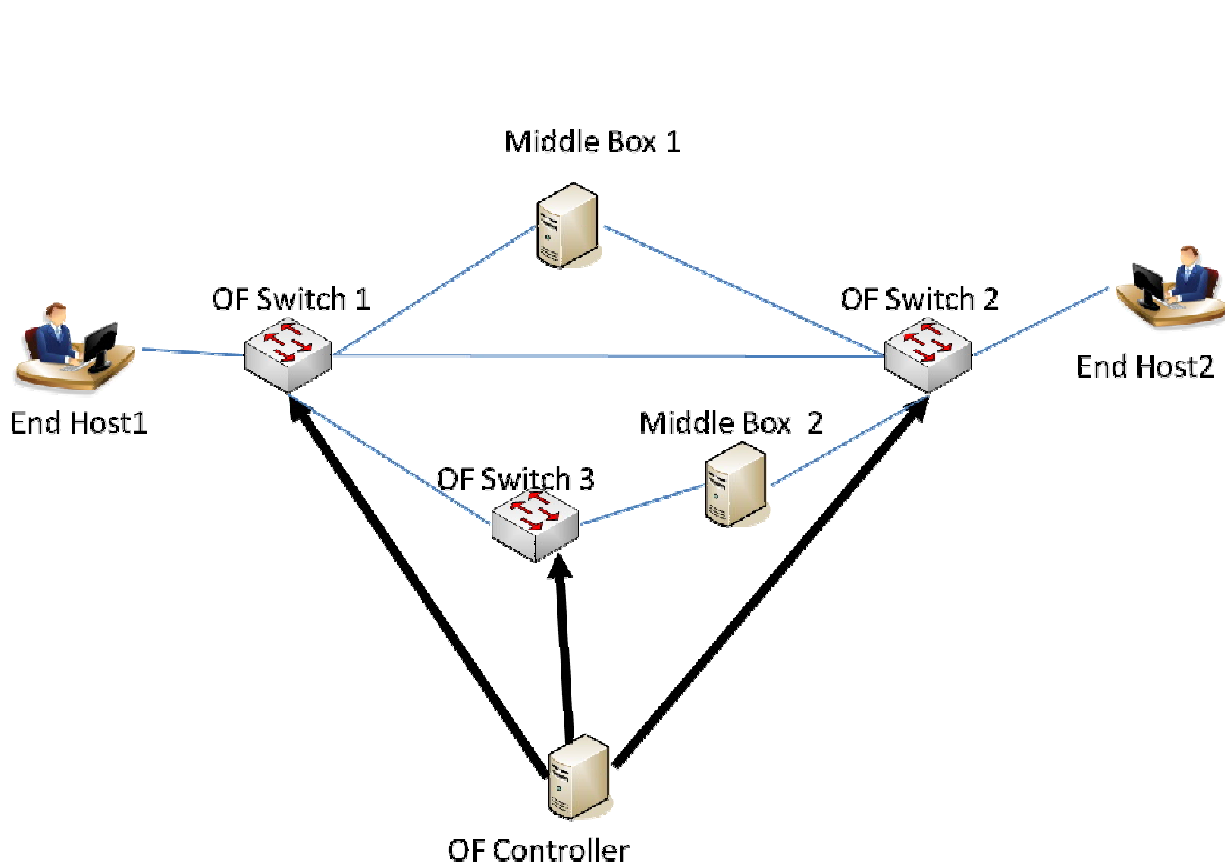To get setup, you need to follow the steps below.

Step 1: Get Spanning tree package from
http://www.openflowswitch.org/wk/index.php/Basic_Spanning_Tree

Step 2: Incorporate spanning tree into your Nox source tree

Step 3: Get topology package from
www.cs.wisc.edu/~akella/CS838/F09/vms/environment.tar.gz

Step 4: Read the README

Step 5: Start your homework

INPUTS:

Your controller should accept as input a file containing the DataPath ID of all the middle boxes.

TESTING:

You can test your controller by using pings.

If middle box1 is active, traffic should go through regardless of whether Middle box 2 is active; however, if middle box1 is inactive then traffic should go through middle box 2. If both middle boxes are inactive traffic should go directly over the link from OFSwitch1 to OFSwitch2. You can take down the virtual machines for the middle boxes to test your controller

WHAT TO SUBMIT:

You should submit a tar ball containing the files required for your component as well as a README file detailing the steps required to setup/install your component.