# **EECS 6381: Distributed Systems Principles, Spring 2017**

Instructor: Aniruddha Gokhale ([a.gokhale@vanderbilt.edu](mailto:a.gokhale@vanderbilt.edu))

**Due April 24, 2017; 11:59 pm in Blackboard (team-based submission)**

**ZooKeeper Register/Lookup in Publish-Subscribe using ZMQ and Mininet**

In this assignment we will build upon the key ideas from Assignments #1 and #2. Keep all the properties supported by the publishers and subscribers from assignment #1 including the QoS properties. Unlike Assignment #2, which used DHT for topic registration and lookup, we are now going to use ZooKeeper. Please see http://zookeeper.apache.org/

A publisher and subscriber can contact any event service that they know of and register themselves with that event service. Any time an event service gets new information, all other event services should acquire this information from ZooKeeper by keeping a watch. Use ZooKeeper to also select one of the event services as the leader who is responsible for all the dissemination. Thus for dissemination purposes you assignment will behave exactly like Assignment #1 because only one ES is responsible for all the mediation (the leader ES).

You should be able to demonstrate failures. So if a leader ES fails, then a new leader should be elected and all publishers/subscribers should find out about this. Once again, you use ZooKeeper’s watch mechanism. Similarly, if a publisher/subscriber were to fail, the ES are notified via the watch mechanism.

**Demonstration: (Similar to Assignment #1 and #2)**

* Please share private github repo with me. My github handle is asgokhale.
* Should work with all properties including failing publishers and even failing event services.
* Should work with multiple publishers publishing multiple different publications and multiple subscribers all of them distributed over different hosts over different kinds of network topologies that we can create in Mininet
* Subscribers may join any time and leave any time
* Do end-to-end measurements (time between publication and receipt of info; since the clock is the same on all emulated hosts, we do not have the issue of clocks drifting apart from each other).

**What to submit?**

Source code, results; README file indicating how to compile and run the code.