CS 4287/5287: Cloud Computing Fall 2015

Programming Assignment #2 (Team-based)

# Handed: 11/03/2015; Due: 11/17/2015 (in Blackboard) by 11:59 pm.

# Originally developed by Prithviraj Patil

## Goal

* To create a fast local testing environment for large scale networking applications using Mininet and Docker/LXC containers.
* To test large scale networking applications with varying network requirements like bandwidth, delay, packet loss, throughput, private/public network, LAN/WAN etc.
* This assignment also uses vagrant (which is covered in a student presentation later but we will just blindly use it for now ☺)

## Motivation and Description

This assignment first gets us started with Docker/LXC to test/deploy large-scale distributed application. We then supplement it by adding some networking-level artifacts. For example, in the cloud environments, we need complex networking scenarios among the host operating systems. Network emulators like mininet/dummynet are very useful in these scenarios. They are virtualization technologies for networks just as hypervisors are for operating systems. They can emulate different types of network entities (switch, router, network cards), network topologies (lan, wan, private/public n/w, vlan etc.), routing algorithms and also network properties like bandwidth, packet loss, throughput etc. Slides for SDN are uploaded and so are some additional handouts.

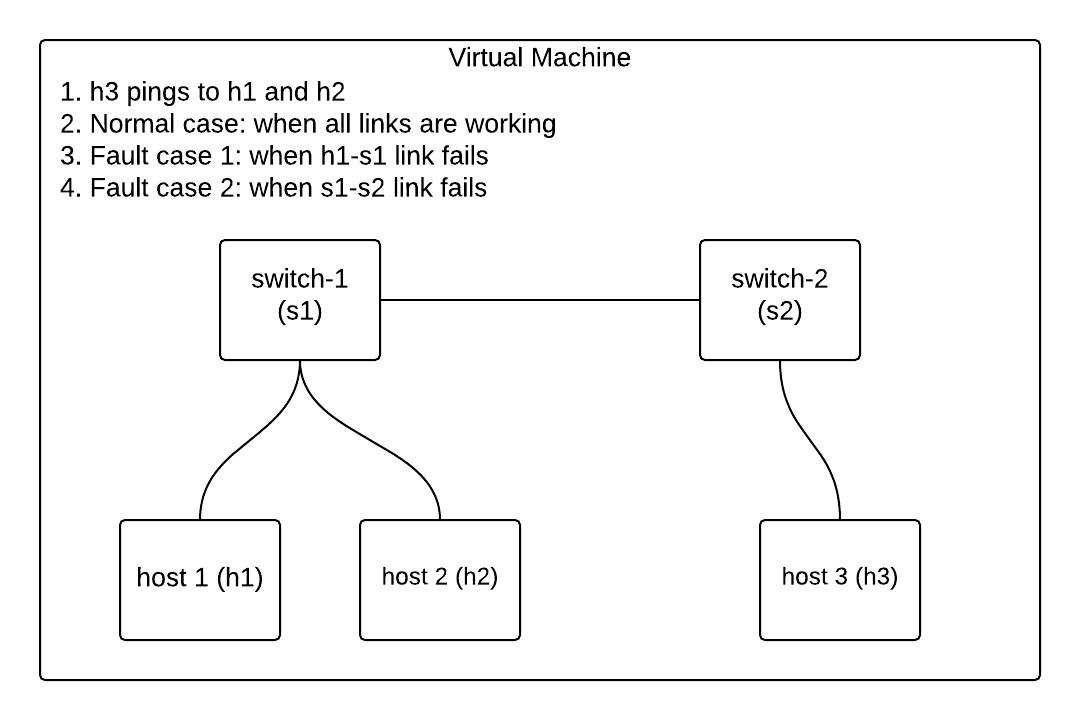
## Notes:

1. Skeleton code for this assignment is provided Blackboard.
2. You need to only modify network\_application.py script file. You should not need to add any extra files.
3. ug folder is for under-graduates (4287) and gr is for graduates (5287).
4. See the README in the respective folders for more details.

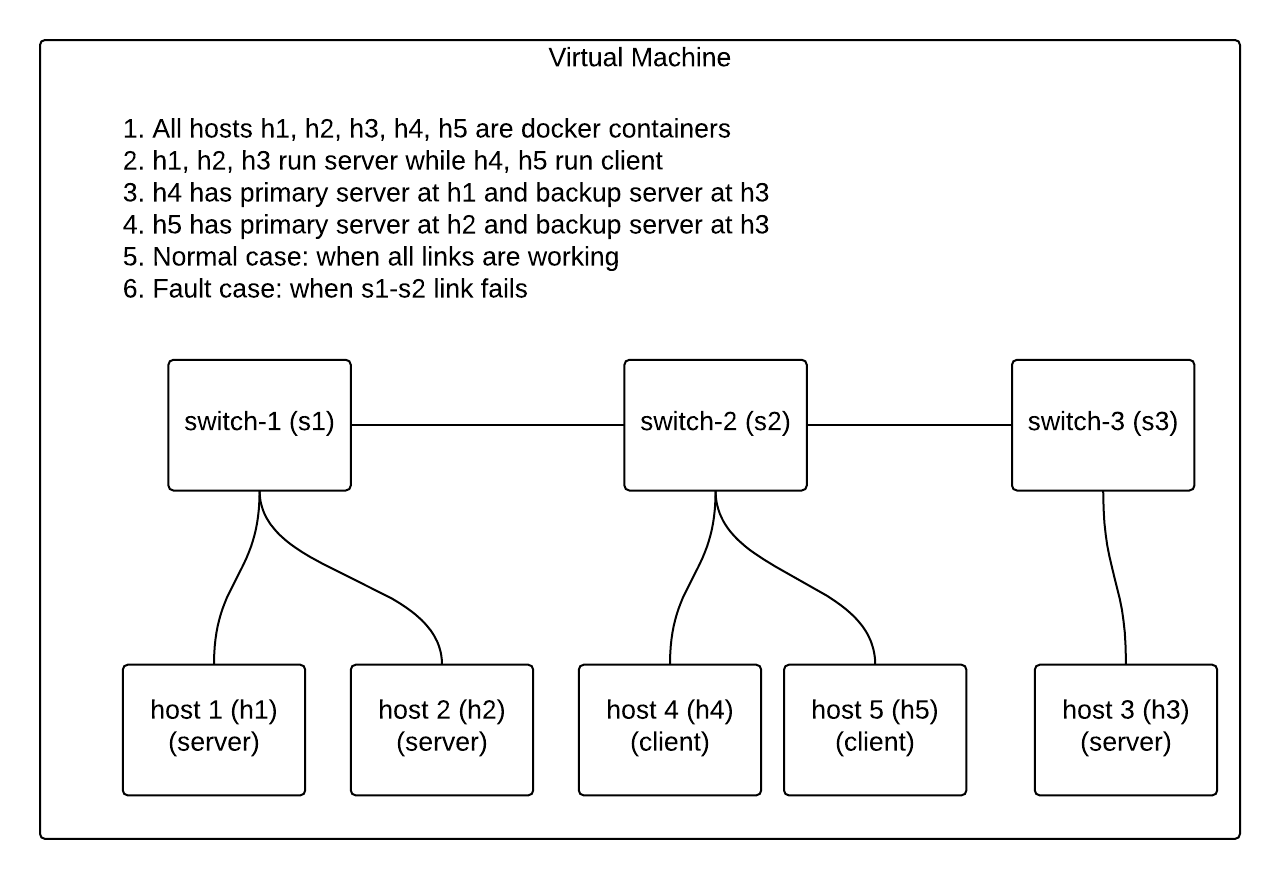
## Grading Rubric:

1. adding hosts: 20
2. add switches: 20
3. add links: 20
4. run application on hosts: 40

## Topology (undergraduates):



**Topology (graduates):**



**Groups are already created in Blackboard for assignment submission. One submission per team.**

**You will need to install vagrant. All you have to do to get started is the command**

**vagrant <supplied vagrantfile>**