5114 Network Programming languages Readme for Optimizations

Consistent Updates : POX style

Description of files:

updator.py : To test the general reach-ability and loop-freeness algorithm I used a component called Updator. The Updator uses a Timer call to obtain the flows to check for logical loops.

The updator function : core.updator.call\_updator() is called in the optimization component. This enables the optimization component to get the flows existing in the network. ( For testing purpose I just return the locally mainitained flows in the form of a hashtable from switch to list of flows. I do not install these rules on the switches )

The updator calls the update\_config (from the consistent\_update module )call at a time 60 seconds in one\_touch\_test. Here the optimization function to check\_if\_one\_touch is called and it returns 0 or 1 to let us know if it is safe to do a one touch update.

loop.py : I use the custom topology to test for loops. This topology has a loop. Although the code will work for any topology.

To create the topology :

sudo mn --custom ~/mininet/custom/loop.py --topo mytopo --arp --mac --controller=remote --switch=ovsk

To run the test :

/home/mininet/pox/pox.py openflow.topology topology openflow.discovery consistent\_update optimization updator

updator uses the consistent\_update function update config.

Based on the flows sent back by the updator, loop presence/absence is detected and one\_touch/ two phase is advocated to the updator.

Evaluation: In order to evaluate the efficiency of the optimizations in a real time scenario I used the load\_balancer as a Case Study.

Description of files :

The same files as above are used except lb\_updator is used to provide the network flows. optimization.py which checks for match equivalence.

The lbalancer.py file first sets 10.0.0.5 as the address where the server which is to be load balanced is, is present. The lbalancer calls the update\_server\_address function at 150 s which updates a flow in switch one. The update\_server\_address calls update\_config.

To run the load balancer application :

/home/mininet/pox/pox.py openflow.topology topology openflow.discovery lbalancer consistent\_update optimization lb\_updator

loadbalancer.py : Custom topology used to test loadbalancer application.

To create the topology :

sudo mn --custom ~/mininet/custom/loadbalancer.py --topo mytopo --arp --mac --controller=remote --switch=ovsk