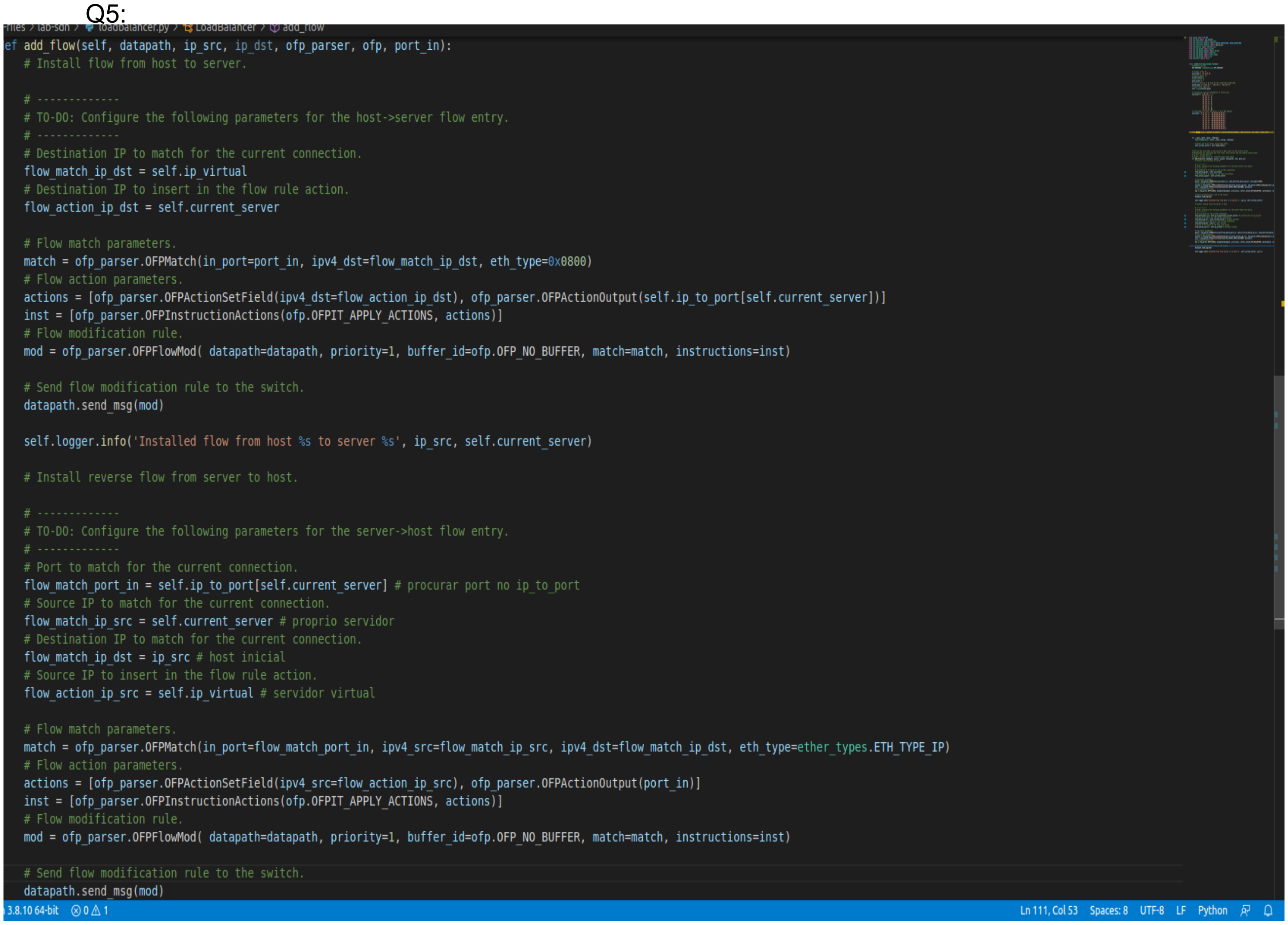
Q1: Because the switch needs to get the rules from the controller, hence making the time greater in the 1st attempt than in the following ones.

Q2: Pings don't get any replies, because there are no rules installed in the switches yet. Also, the controller is offline, so the switch won't have the rules installed to process the packet.

Q3: It doesn't work again. We need 2 more rules to be able to handle the reverse path: one in s2 to be able to forward from h2 to s1 and another one in s1 to be able to forward from s2 to h1. These 2 rules are similar to the last one we added: they would have to specify input port, output port (action), MAC address of source and MAC address of destination.

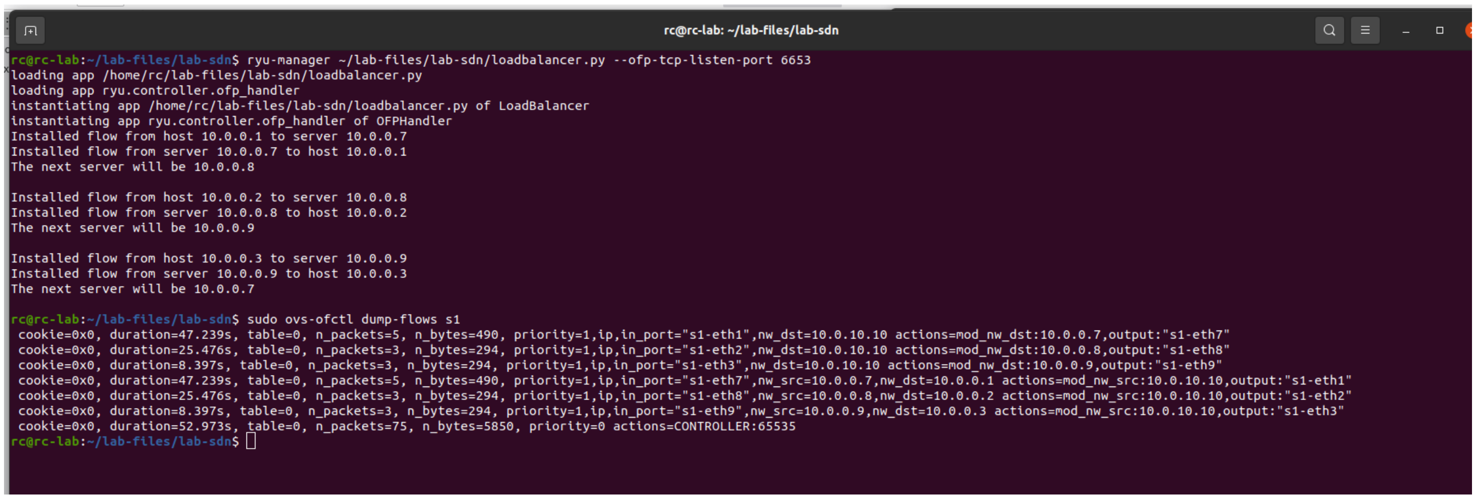
Q4: Yes, because the switch already has the rules installed. From h1 to h3, it isn't possible, because the switch doesn't have the necessary rules and the controller is not online to compute and install them.

Q5:



```
files > lab-sdn > loadbalancer.py > LoadBalancer > add_flow
ef add_flow(self, datapath, ip_src, ip_dst, ofp_parser, ofp, port_in):
   # Install flow from host to server.

   # ------------
   # TO-DO: Configure the following parameters for the host->server flow entry.
   # ------------
   # Destination IP to match for the current connection.
   flow_match_ip_dst = self.ip_virtual
   # Destination IP to insert in the flow rule action.
   flow_action_ip_dst = self.current_server

   # Flow match parameters.
   match = ofp_parser.OFPMatch(in_port=port_in, ipv4_dst=flow_match_ip_dst, eth_type=0x0800)
   # Flow action parameters.
   actions = [ofp_parser.OFPActionSetField(ipv4_dst=flow_action_ip_dst), ofp_parser.OFPActionOutput(self.ip_to_port[self.current_server])]
   inst = [ofp_parser.OFPInstructionActions(ofp.OFPIT_APPLY_ACTIONS, actions)]
   # Flow modification rule.
   mod = ofp_parser.OFPFlowMod( datapath=datapath, priority=1, buffer_id=ofp.OFP_NO_BUFFER, match=match, instructions=inst)

   # Send flow modification rule to the switch.
   datapath.send_msg(mod)

   self.logger.info('Installed flow from host %s to server %s', ip_src, self.current_server)

   # Install reverse flow from server to host.

   # ------------
   # TO-DO: Configure the following parameters for the server->host flow entry.
   # ------------
   # Port to match for the current connection.
   flow_match_port_in = self.ip_to_port[self.current_server] # procurar port no ip_to_port
   # Source IP to match for the current connection.
   flow_match_ip_src = self.current_server # proprio servidor
   # Destination IP to match for the current connection.
   flow_match_ip_dst = ip_src # host inicial
   # Source IP to insert in the flow rule action.
   flow_action_ip_src = self.ip_virtual # servidor virtual

   # Flow match parameters.
   match = ofp_parser.OFPMatch(in_port=flow_match_port_in, ipv4_src=flow_match_ip_src, ipv4_dst=flow_match_ip_dst, eth_type=ether_types.ETH_TYPE_IP)
   # Flow action parameters.
   actions = [ofp_parser.OFPActionSetField(ipv4_src=flow_action_ip_src), ofp_parser.OFPActionOutput(port_in)]
   inst = [ofp_parser.OFPInstructionActions(ofp.OFPIT_APPLY_ACTIONS, actions)]
   # Flow modification rule.
   mod = ofp_parser.OFPFlowMod( datapath=datapath, priority=1, buffer_id=ofp.OFP_NO_BUFFER, match=match, instructions=inst)

   # Send flow modification rule to the switch.
   datapath.send_msg(mod)
3.8.10 64-bit  ⊗ 0 ⚠ 1                                        Ln 111, Col 53  Spaces: 8  UTF-8  LF  Python
```

Q6:



rc@rc-lab: ~/lab-files/lab-sdn

```
rc@rc-lab:~/lab-files/lab-sdn$ ryu-manager ~/lab-files/lab-sdn/loadbalancer.py --ofp-tcp-listen-port 6653
loading app /home/rc/lab-files/lab-sdn/loadbalancer.py
loading app ryu.controller.ofp_handler
instantiating app /home/rc/lab-files/lab-sdn/loadbalancer.py of LoadBalancer
instantiating app ryu.controller.ofp_handler of OFPHandler
Installed flow from host 10.0.0.1 to server 10.0.0.7
Installed flow from server 10.0.0.7 to host 10.0.0.1
The next server will be 10.0.0.8

Installed flow from host 10.0.0.2 to server 10.0.0.8
Installed flow from server 10.0.0.8 to host 10.0.0.2
The next server will be 10.0.0.9

Installed flow from host 10.0.0.3 to server 10.0.0.9
Installed flow from server 10.0.0.9 to host 10.0.0.3
The next server will be 10.0.0.7

rc@rc-lab:~/lab-files/lab-sdn$ sudo ovs-ofctl dump-flows s1
 cookie=0x0, duration=47.239s, table=0, n_packets=5, n_bytes=490, priority=1,ip,in_port="s1-eth1",nw_dst=10.0.10.10 actions=mod_nw_dst:10.0.0.7,output:"s1-eth7"
 cookie=0x0, duration=25.476s, table=0, n_packets=3, n_bytes=294, priority=1,ip,in_port="s1-eth2",nw_dst=10.0.10.10 actions=mod_nw_dst:10.0.0.8,output:"s1-eth8"
 cookie=0x0, duration=8.397s, table=0, n_packets=3, n_bytes=294, priority=1,ip,in_port="s1-eth3",nw_dst=10.0.10.10 actions=mod_nw_dst:10.0.0.9,output:"s1-eth9"
 cookie=0x0, duration=47.239s, table=0, n_packets=5, n_bytes=490, priority=1,ip,in_port="s1-eth7",nw_src=10.0.0.7,nw_dst=10.0.0.1 actions=mod_nw_src:10.0.10.10,output:"s1-eth1"
 cookie=0x0, duration=25.476s, table=0, n_packets=3, n_bytes=294, priority=1,ip,in_port="s1-eth8",nw_src=10.0.0.8,nw_dst=10.0.0.2 actions=mod_nw_src:10.0.10.10,output:"s1-eth2"
 cookie=0x0, duration=8.397s, table=0, n_packets=3, n_bytes=294, priority=1,ip,in_port="s1-eth9",nw_src=10.0.0.9,nw_dst=10.0.0.3 actions=mod_nw_src:10.0.10.10,output:"s1-eth3"
 cookie=0x0, duration=52.973s, table=0, n_packets=75, n_bytes=5850, priority=0 actions=CONTROLLER:65535
rc@rc-lab:~/lab-files/lab-sdn$
```