

Q1) We had to build a topology consisting of two hosts and one switch(h1,h2,s1). To create the topology we need to make a class overriding the method 'build(self)'.

```
class TwoNodes(Topo):  
    switch_s1 = None  
    host_h1 = None  
    host_h2 = None  
  
    def build(self):
```

- To add switches in the network use 'self.addSwitch("switch\_name")'.
- To add hosts use 'self.addHost('host\_name')'
- To make a link between them use 'self.addLink(var1,var2)'

To make the topology instantiate the above class and pass it as a param of Mininet. i.e

```
topo = TwoNodes()  
net = Mininet(topo = topo)
```

- To start the simulation use

```
net.start()
```

- To get the components from the topology use(i.e net.get('name\_of\_switch/host'))

```
net.get()
```

- To get the ip of a host use host\_name.IP()
- To run a command in the host use host\_var\_name.cmd('the\_command')
- To start the server use iperf -s -p 5111 -i 2 > /tmp/server\_log.txt &
  - Here -p represents the port number
  - To monitor the server at some time interval use -i time\_interval
  - Use -s to start the server
- To start the client use iperf -c ' + server\_ip+' -p 5111 -t 20 > /tmp/throughput.txt
  - To send packets for t time use -t time\_interval
  - To start the the client use -c with the server\_ip you want to listen

Q2) This uses all the commands as the previous one except for the following -

- To define the bandwidth of the link use `bw= x` (where x is in Mbps)
- To define the delay in the link use `delay='xms'`
- To define the packet drop % use `loss = x` (where x is percentage packet drop)
- To define the queue size at switch use `max_queue_size=x` (where x is some number)
- To ping all the hosts use `net.pingAll()`
- To dump the host connections

```
dumpNodeConnections( net.hosts )
```

Note -

Use the following function to assign dpid to switches

```
def int2dpid( self,dpid ):
    try:
        dpid = hex( dpid )[ 2: ]
        dpid = '0' * ( 16 - len( dpid ) ) + dpid
        return dpid
    except IndexError:
        raise Exception( 'Unable to derive default datapath ID - '
                        'please either specify a dpid or use
a '
                        'canonical switch name such as s23.' )
```