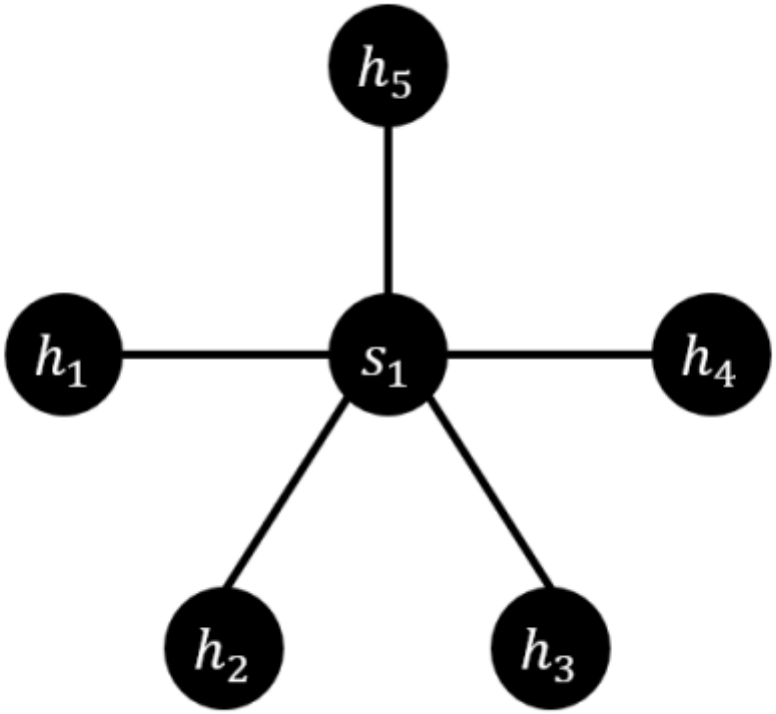
1. Topology

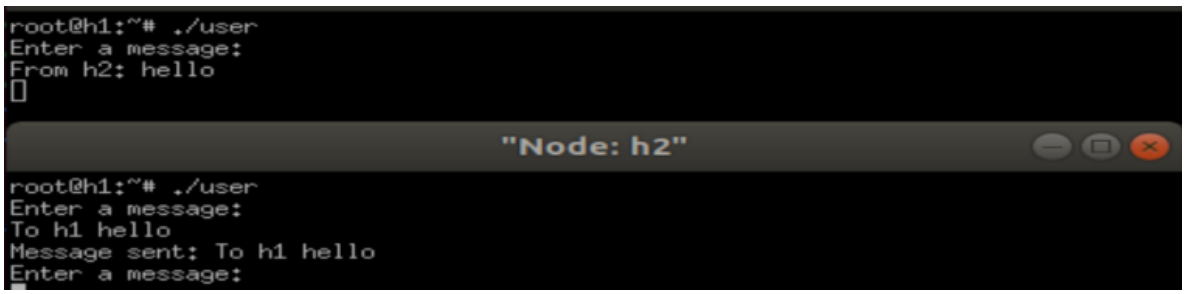


```python
class SingleSwitchTopo( Topo ):
    "Single switch connected to n hosts."
    def build( self ):
        switch1 = self.addSwitch('s1')

        host1 = self.addHost('h1')
        host2 = self.addHost('h2')
        host3 = self.addHost('h3')
        host4 = self.addHost('h4')
        host5 = self.addHost('h5')

        self.addLink( switch1, host1)
        self.addLink(switch1, host2)
        self.addLink(switch1, host3)
        self.addLink(switch1, host4)
        self.addLink(switch1, host5)



def perfTest():
    "Create network and run simple performance test"
    topo = SingleSwitchTopo()
    net = Mininet( topo=topo,
                   link=TCLink )
    net.start()

    CLI(net)

    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
```

2. Chat room with TCP

The center h5 is created in center.cpp, and h1,2,3,4 are created in user.cpp. Because h1,2,3,4 actually only interact with h5, we use lpthread so that we don't need to create a new socket every time. To compile them, following commands are need instead.

```
g++ center.cpp -o center -lpthread
g++ user.cpp -o user -lpthread
```



3. Chat room with UDP

   The message is broadcasted, the broadcast address is "10.255.255.255"

   My udpclient.cpp file is based on the examples given on the website. In the for loop, there is a if-statement: if strcmp(rv_addr, local_addr). If the sender's IP is the same as mine, it means I'm exactly the sender, then I will not display it. Otherwise, I print the message.

.