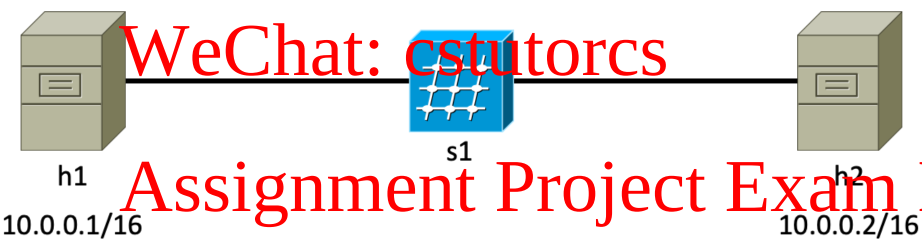
The Actual Labs

LAB 1 (20 points)

Mininet allows us t[illegible]rs from the command line. If we specify a link of type `tc`, the softwa[illegible]dwidth and the delay for each link. We can also run bandwidth test[illegible]rf command.

The default minine[illegible]osts’ connected to one virtual switch:



- Start mininet without any arguments:

  ```
  sudo mn
  ```

  Then in the mininet terminal, run:

  ```
  h2 iperf -s &
  ```

  ```
  h1 iperf -t 10 -c 10.0.0.2
  ```

  After iperf finishes, run:

  ```
  h1 ping -c 3 h2
  ```

Question 1: What is the bandwidth and the average delay reported between the two hosts?

Now let’s limit the bandwidth, and increase the delay, and see what happens.

Exit out of mininet with ‘quit’.

Start mininet with these arguments:

```
sudo mn --link tc,bw=10,delay=10ms
```


Then in the mininet terminal, run iperf:

```
h2 iperf -s &
h1 iper... 0.0.2
```

After iperf finishes, ...

```
h1 ping ...
```

Question 2: What is the bandwidth and the average delay reported between the two hosts?

By default, Mininet hosts start with randomly assigned MAC addresses. Every time the Mininet is created, the MAC addresses change, so correlating traffic with specific hosts can be a challenge. We can use `--mac` to set MAC and IP addresses to small, unique, easy to read values. This will come in handy in future labs.

Start mininet:

```
sudo mn
```


Then in the mininet terminal, run:

```
h1 ifconfig
```


Question 3: What is IP address and the MAC address of the host?

Now try the --mac option

- Start mininet with these arguments:

```
sudo mn --mac
```

Then in the mininet terminal, run

```
h1 ifconfig
```

Question 4. Now what is the IP address and the MAC address of the host?
Hint: IP address may not change but the MAC address will change to a simpler one.

LAB 2 (20 points)

Step 1. Start ipmini ... ple topology for Layer 2 Spanning Tree:

```
sudo pyth ... et.examples \
--topo=sp ... twork
```

Step 2. Type 'net' t ... ted to what. You'll see something similar to this:

```
mininet> net
hs1 hs1-eth0:s1-eth3
hs2 hs2-eth0:s2-eth3
hs3 hs3-eth0:s3-eth3
s1 lo:  s1-eth1:s2-eth1 s1-eth2:s3-eth1 s1-eth3:hs1-eth0
s2 lo:  s2-eth1:s1-eth1 s2-eth2:s3-eth2 s2-eth3:hs2-eth0
s3 lo:  s3-eth1:s1-eth2 s3-eth2:s2-eth2 s3-eth3:hs3-eth0
mininet>
```


Q1: Using your favorite drawing program, or just a pencil and paper, draw a network diagram that includes the three hosts (hs1, hs2, hs3) and the three switches (s1, s2, s3). Be sure to show all interconnections between hosts and switches. Label the interfaces.

Step 3: Using 'ifconfig' find the MAC address of the three hosts. For example:

```
hs1 ifconfig
hs2 ifconfig
hs3 ifconfig
```

Q2: Add the MAC address of each host's interface to your diagram from #2.

Step 4: This lab uses Linux bridges to implement the 'switch'. What it is actually doing is implementing three different bridges on a single container. In the mininet cli, s1, s2 and s3 "switches" are actually the same container. But they are then logically separated inside the container as three bridges, also called s1, s2 and s3. It's the logically separated ones we care about in this lab.

Execute the command:

```
s1 brctl show
```

Q3: How many switches/bridges do you see?

Note you could typ ... get the same output (reason given above).

Step 5: Now execut

```
s1 brctl
s1 brctl showmacs s2
s1 brctl showmacs s3
```

Q4: What do you see? Can you figure out which MAC addresses belong to what devices? Using 'ifconfig' find and document where each MAC came from.

Step 6: Run the command:

```
ping4all
```


Q5: This will send some traffic (an ICMP ECHO request) to and from each host. How does this change the output of 'brctl showmacs' for all three bridges? Why is this? Explain your answer.

Step 7: Using the mininet command:

s1 brctl showstp s1
s1 brctl showstp s2
s1 brctl showstp s3

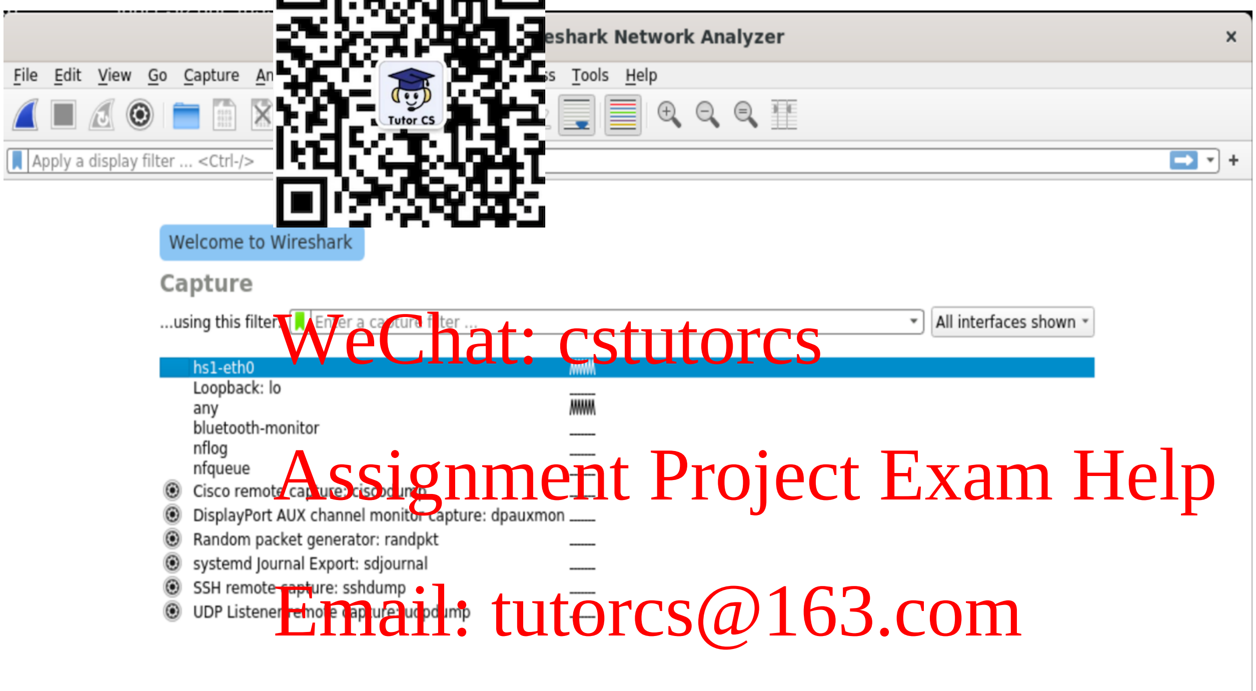
Q6: Find which interface in the spanning tree topology is in the blocked state. Indicate which interface this is in your diagram above.

Q7: Find which bridge is the spanning tree root and document why it was elected as such.

Step 8: Start up Wireshark on hs1. At the mininet prompt:


```
hs1 wireshark &
```

You should see this pop up on your screen:



Choose the hs1-eth0 interface to dump packets (double click on it).

You will start to see packets being captured in the Wireshark interface:

In the top window, pick one of the STP packets (spanning tree protocol).

In the middle window (underneath) start to dissect the packet. In this case there are four constituent parts – Frame, Ethernet, Link Layer and Spanning Tree itself.

Answer these questions about the captured STP packet and show some annotated screen shots to validate your work:

Q8: How many bits on the wire is the total STP packet?

Q9: What is the source MAC address of the STP packet, and what is the destination MAC address? What is special about that destination MAC?

Q10: Dig into the spanning tree portion of the packet capture. Find validation that the STP root you found in the previous lab exercise is correct. Show your work (annotated screen shot) and explain.