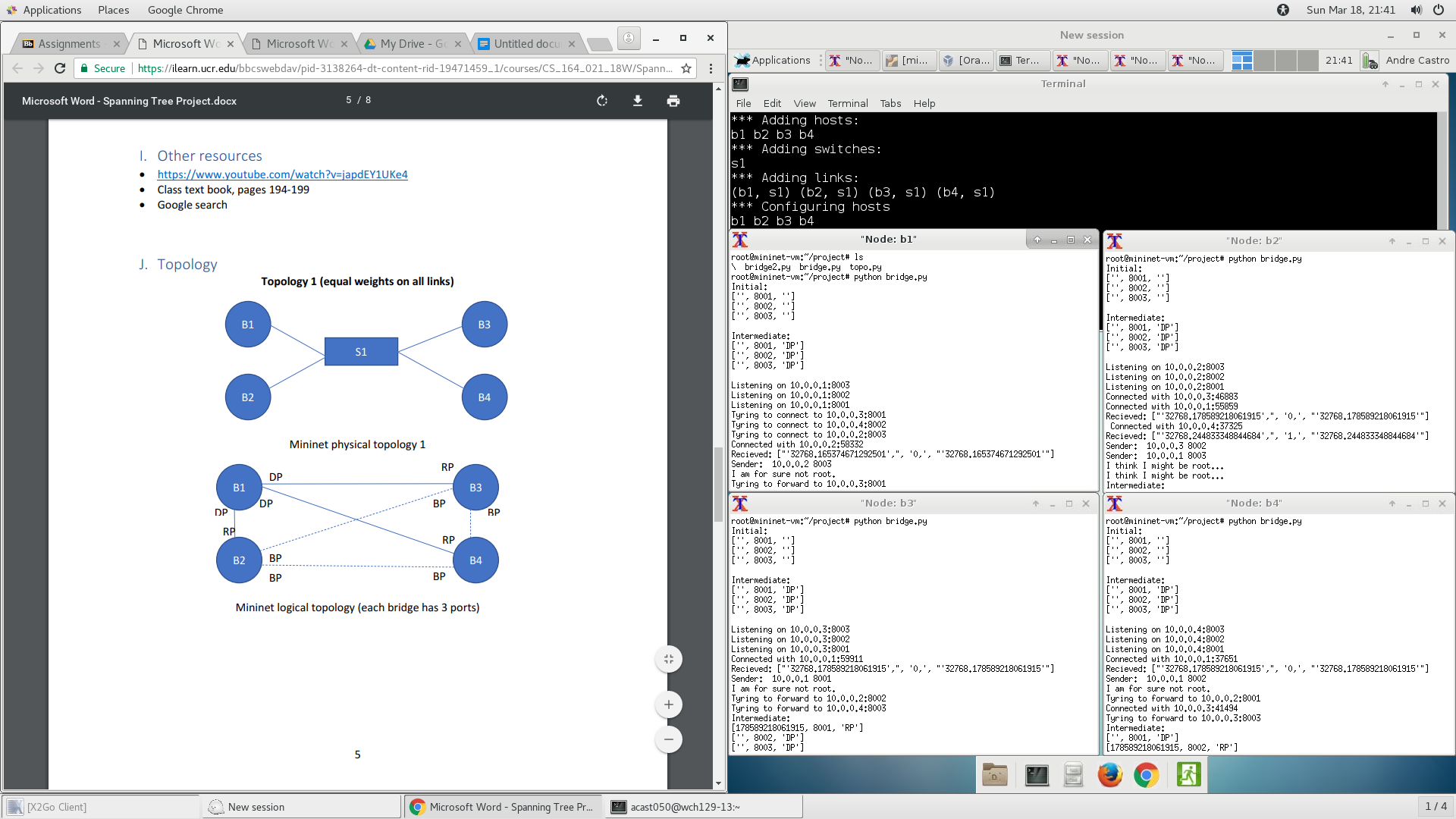
CS164 Project

Andre Castro

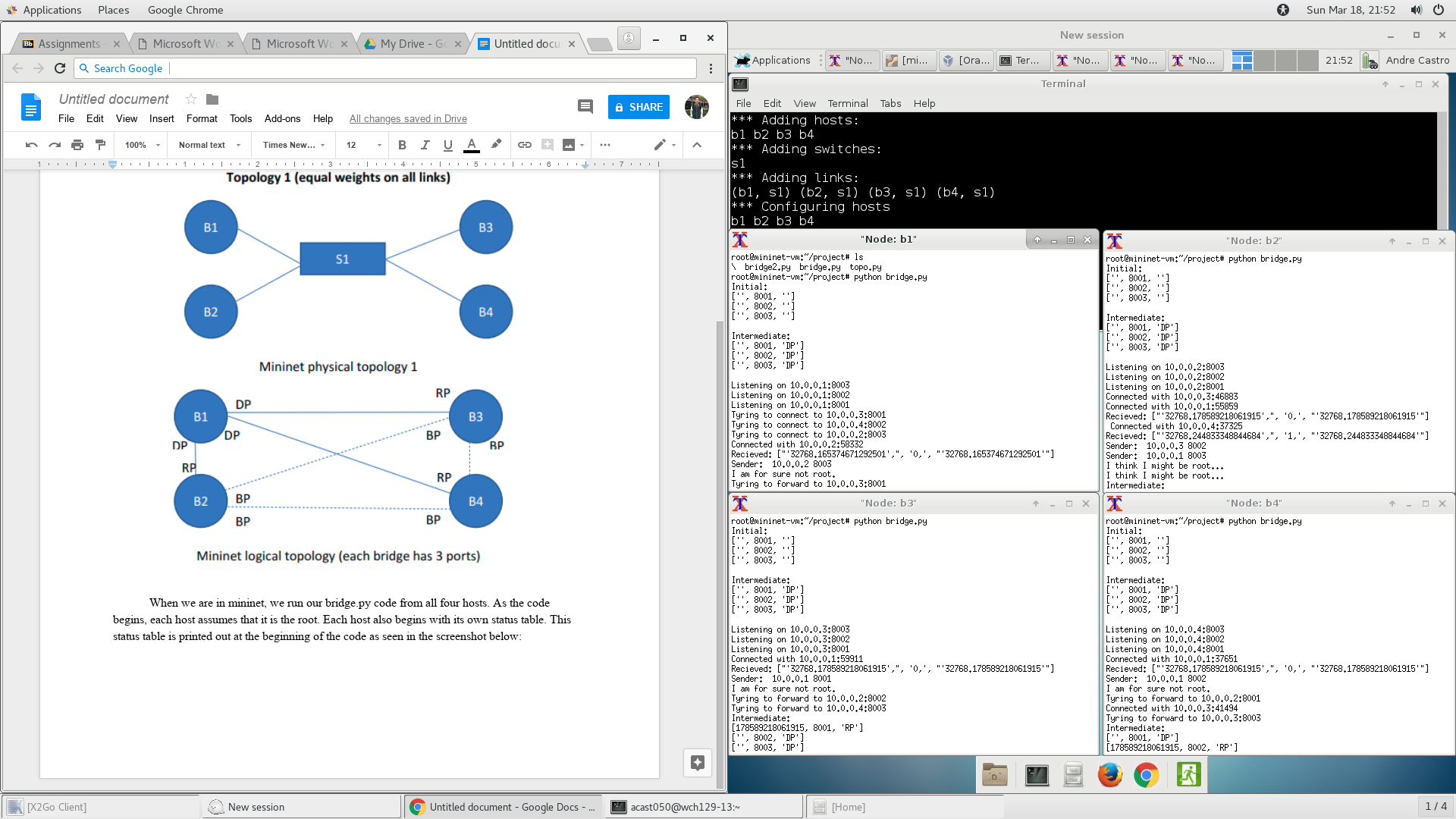
SID: 861170524

For this project we implemented the spanning tree algorithm. It was completed in python using mininet to simulate a network.

To begin, we opened mininet using our topology topo.py. This file described a topology where there were four hosts connected to one switch. The diagram below details the topology:



When we are in mininet, we run our bridge.py code from all four hosts. As the code begins, each host assumes that it is the root. Each host also begins with its own status table. This status table is printed out at the beginning of the code as seen in the screenshot below:

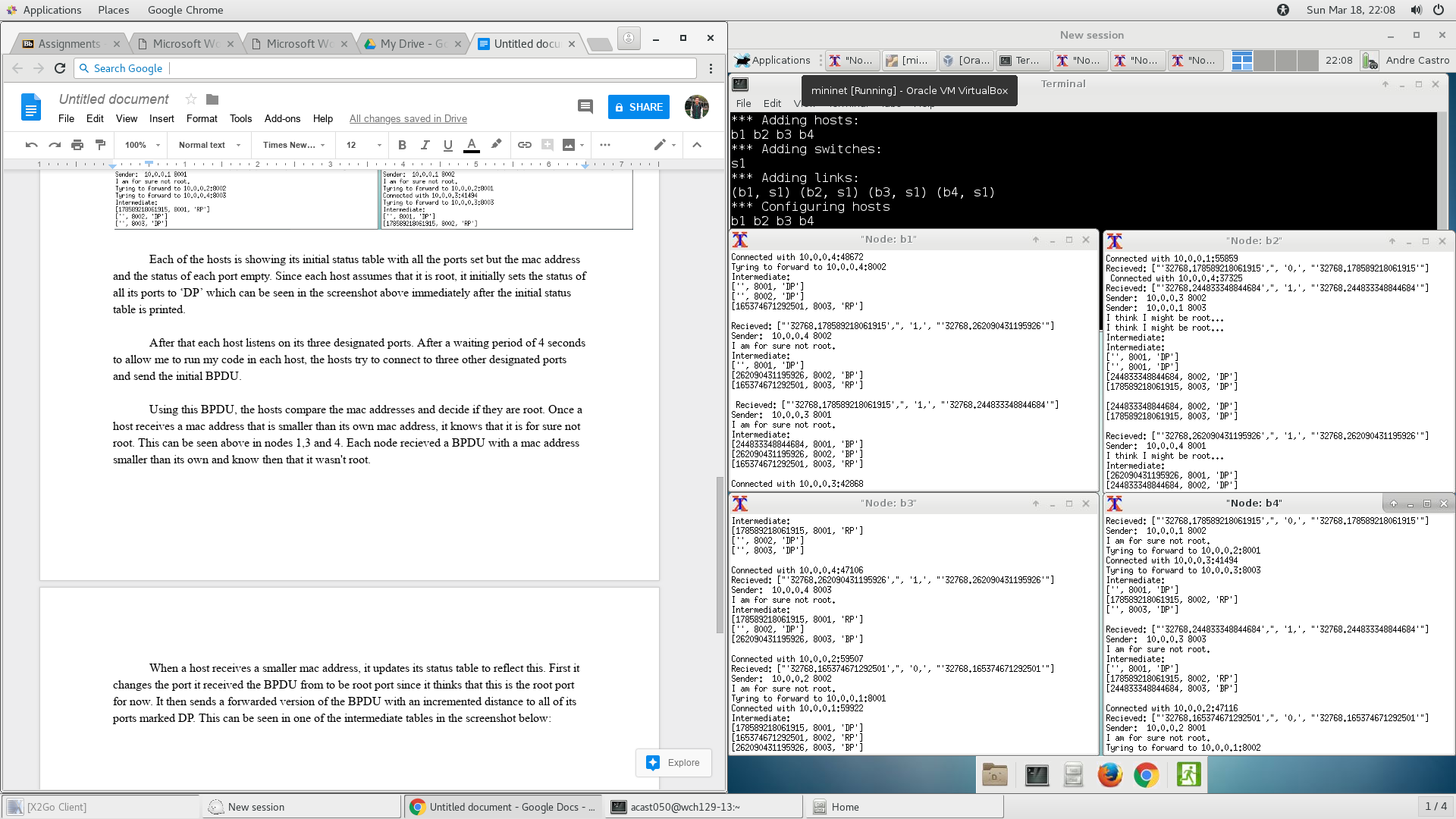


Each of the hosts is showing its initial status table with all the ports set but the mac address and the status of each port empty. Since each host assumes that it is root, it initially sets the status of all its ports to ‘DP’ which can be seen in the screenshot above immediately after the initial status table is printed.

After that each host listens on its three designated ports. After a waiting period of 4 seconds to allow me to run my code in each host, the hosts try to connect to three other designated ports and send the initial BPDU.

Using this BPDU, the hosts compare the mac addresses and decide if they are root. Once a host receives a mac address that is smaller than its own mac address, it knows that it is for sure not root. This can be seen above in nodes 1,3 and 4. Each node recieved a BPDU with a mac address smaller than its own and know then that it wasn't root.

When a host receives a smaller mac address, it updates its status table to reflect this. First it changes the port it received the BPDU from to be root port since it thinks that this is the root port for now. It then sends a forwarded version of the BPDU with an incremented distance to all of its ports marked DP. This can be seen in one of the intermediate tables in the screenshot below:

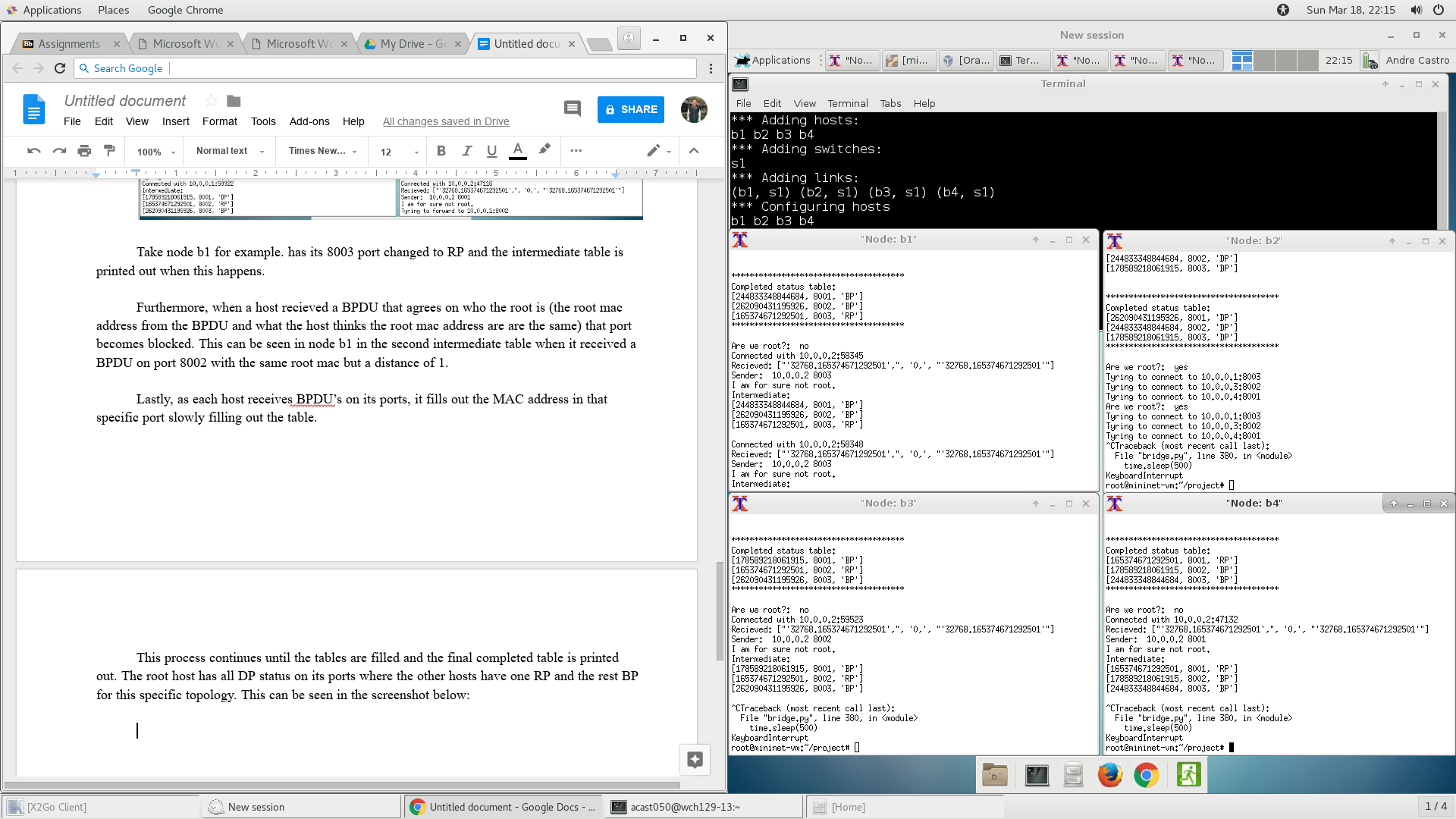


Take node b1 for example. has its 8003 port changed to RP and the intermediate table is printed out when this happens.

Furthermore, when a host recieved a BPDU that agrees on who the root is (the root mac address from the BPDU and what the host thinks the root mac address are are the same) that port becomes blocked. This can be seen in node b1 in the second intermediate table when it received a BPDU on port 8002 with the same root mac but a distance of 1.

Lastly, as each host receives BPDU’s on its ports, it fills out the MAC address in that specific port slowly filling out the table.

This process continues until the tables are filled and the final completed table is printed out. The root host has all DP status on its ports where the other hosts have one RP and the rest BP for this specific topology. This can be seen in the screenshot below:



After the completed status table is printed, the system is in its stable state where the root is the only host sending BPDU’s. This can be seen as the window in the top right, node b2 is the only node printing the the “Trying to connect” messages. The rest of the windows are just receiving BPDU’s and updating their status tables if necessary.