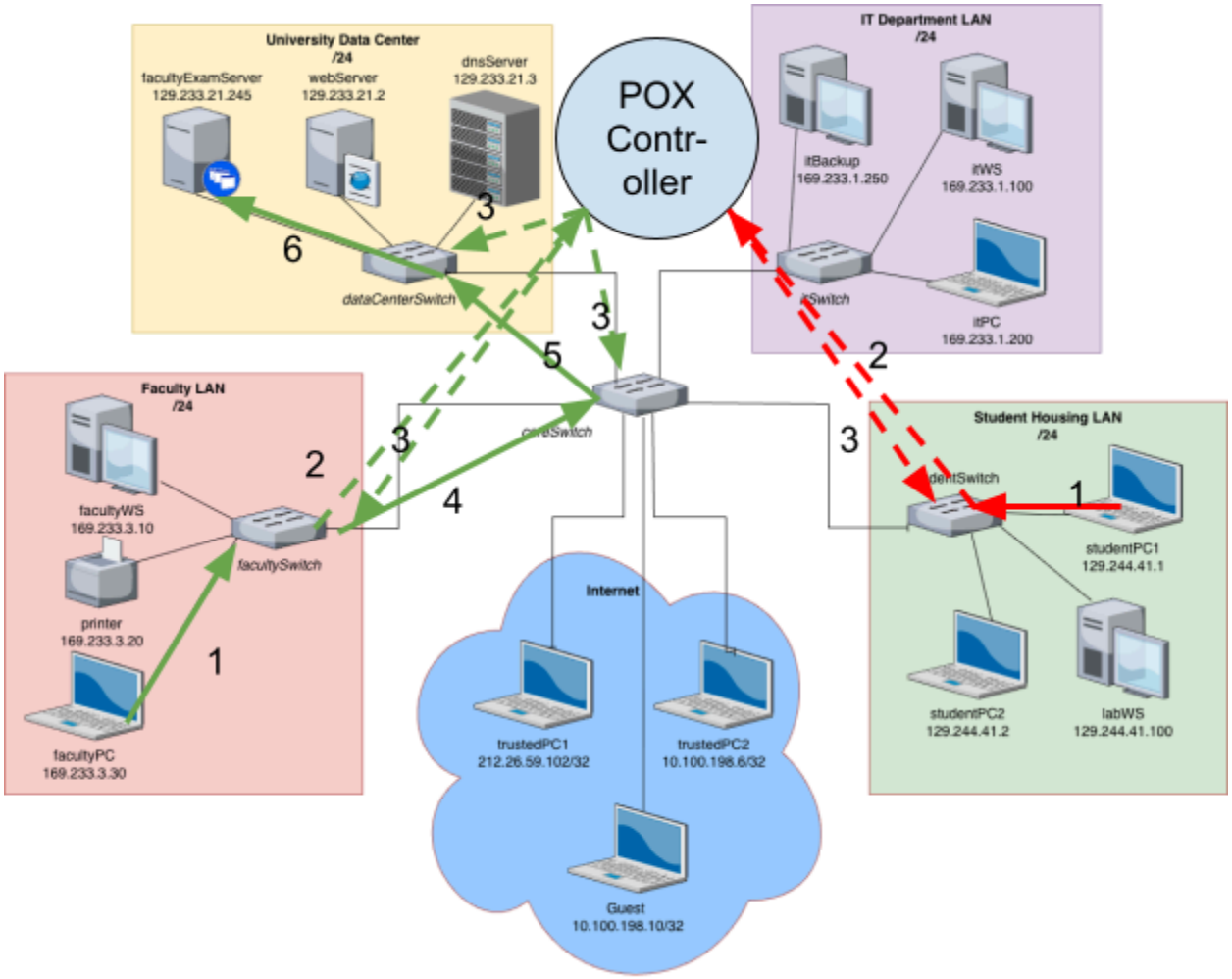
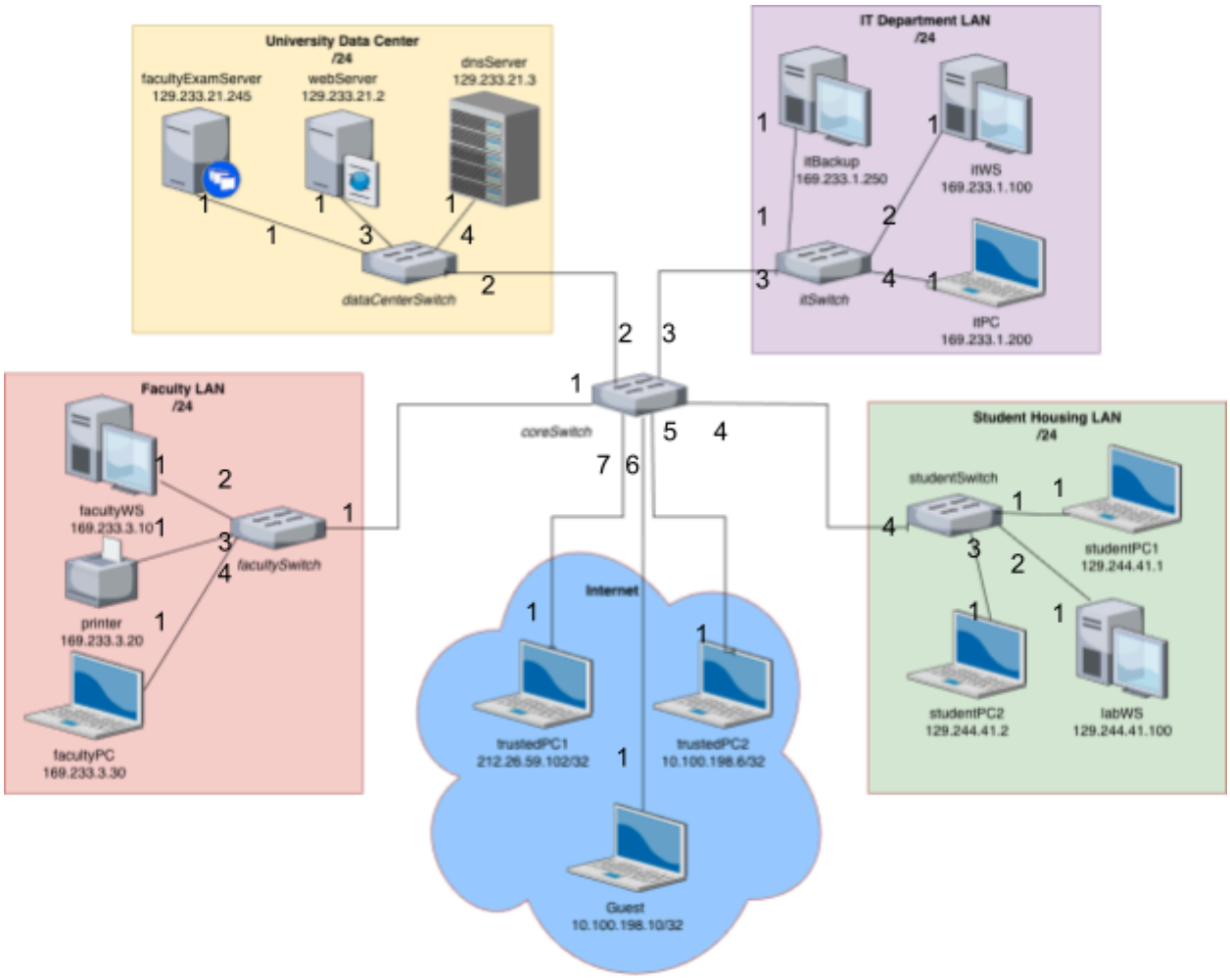
Q1

a) In lab 5 packets were forwarded through a firewall using POX and used flooding to send traffic through the network. This lab instead uses POX as a router and only uses flood for ARP packets. This means the POX has to find specific output ports and forward packets based on subnets before being able to get to the hosts. The accept() function will no longer be a catchall that forwards passed packets with flooding. Instead, it will be used to set specific forwarding rules for routing between different subnets.

b)



Q2


University Data Center
/24
facultyExamServer
129.233.21.245
webServer
129.233.21.2
dnsServer
129.233.21.3
dataCenterSwitch
IT Department LAN
/24
itBackup
169.233.1.250
itWS
169.233.1.100
itSwitch
itPC
169.233.1.200
coreSwitch
Faculty LAN
/24
facultyWS
169.233.3.10
facultySwitch
printer
169.233.3.20
facultyPC
169.233.3.30
Internet
trustedPC1
212.26.59.102/32
trustedPC2
10.100.196.6/32
Guest
10.100.198.10/32
Student Housing LAN
/24
studentSwitch
studentPC1
129.244.41.1
studentPC2
129.244.41.2
labWS
129.244.41.100

Q3

```
mininet> pingall
*** Ping: testing ping reachability
dnsServer -> examServer X X X X X X X X X X X webServer
examServer -> dnsServer X X X X X X X X X X X webServer
facultyPC -> X X facultyWS X itBackup itPC itWS X printer X X X X X
facultyWS -> X X facultyPC X itBackup itPC itWS X printer X X X X X
guest -> X X X X X X X X X X X X X
itBackup -> X X facultyPC facultyWS X itPC itWS labWS printer studentPC1 studentPC2 X X X
itPC -> X X facultyPC facultyWS X itBackup itWS labWS printer studentPC1 studentPC2 X X X
itWS -> X X facultyPC facultyWS X itBackup itPC labWS printer studentPC1 studentPC2 X X X
labWS -> X X X X X itBackup itPC itWS X studentPC1 studentPC2 X X X
printer -> X X facultyPC facultyWS X itBackup itPC itWS X X X X X X
studentPC1 -> X X X X X itBackup itPC itWS labWS X studentPC2 X X X
studentPC2 -> X X X X X itBackup itPC itWS labWS X studentPC1 X X X
trustedPC1 -> X X X X X X X X X X X X X
trustedPC2 -> X X X X X X X X X X X X X
webServer -> dnsServer examServer X X X X X X X X X X X
*** Results: 71% dropped (60/210 received)
mininet>
```

Yes, pingall is working as expected. We are meant to block all ICMP traffic except for traffic from one device to another within the same subnet, as well as specific traffic between IT and Faculty and between IT and Student Housing. As we can see in the printer example, faculty devices from its own subnet can be accessed, as well as IT devices from the IT subnet.

Q4

```
mininet> iperf trustedPC2 guest
*** Iperf: testing TCP bandwidth between trustedPC2 and guest
*** Results: ['70.8 Gbits/sec', '70.5 Gbits/sec']
mininet> iperf guest printer
*** Iperf: testing TCP bandwidth between guest and printer
*** Results: ['68.2 Gbits/sec', '68.3 Gbits/sec']
mininet> iperf trustedPC1 printer
*** Iperf: testing TCP bandwidth between trustedPC1 and printer
*** Results: ['70.2 Gbits/sec', '70.0 Gbits/sec']
mininet> iperf trustedPC1 webServer
*** Iperf: testing TCP bandwidth between trustedPC1 and webServer
*** Results: ['68.0 Gbits/sec', '67.8 Gbits/sec']
mininet> iperf examServer studentPC1
*** Iperf: testing TCP bandwidth between examServer and studentPC1
^C
Interrupt
mininet> iperf examServer facultyPC
*** Iperf: testing TCP bandwidth between examServer and facultyPC
*** Results: ['66.0 Gbits/sec', '65.7 Gbits/sec']
mininet> iperf examServer facultyWS
*** Iperf: testing TCP bandwidth between examServer and facultyWS
*** Results: ['69.9 Gbits/sec', '69.8 Gbits/sec']
mininet> iperf studentPC1 facultyWS
*** Iperf: testing TCP bandwidth between studentPC1 and facultyWS
*** Results: ['67.8 Gbits/sec', '67.8 Gbits/sec']
mininet> iperf studentPC2 labWS
*** Iperf: testing TCP bandwidth between studentPC2 and labWS
*** Results: ['69.4 Gbits/sec', '69.2 Gbits/sec']
mininet> iperf facultyWS labWS
*** Iperf: testing TCP bandwidth between facultyWS and labWS
*** Results: ['63.8 Gbits/sec', '63.8 Gbits/sec']
mininet> iperf itPC itBackup
*** Iperf: testing TCP bandwidth between itPC and itBackup
*** Results: ['67.7 Gbits/sec', '66.9 Gbits/sec']
```

The individual results were as expected. The only failure we saw was between examServer and studentPC1. This was expected to fail because Rule 2 states that only Faculty subnet devices may access the Exam Server using TCP. Even though there are a lot more success messages than failures, this is because Rule 2 explicitly allows traffic between large groups of devices. This includes all non-Internet devices with each other, as well as Internet devices with the University Data Center.

Q5

a) itWS and facultyPC's results were ['10M', '10.5Mbits/sec', '10.5Mbits/sec']. This success was expected because the IT and Faculty LANs were both included in the list of subnets that were allowed to use UDP connections.

b) **trustedPC1 and dnsServer** fail because **rule 3 only allows UDP traffic between** "The University Data Center, IT Department, Faculty LAN and the Student Housing LAN". This doesn't include our Internet devices.
trustedPC2 and guest fail for the same reason as trustedPC1 and dnsServer. Additionally, despite them having IP addresses that share the same first many numbers, they are explicitly not part of the same subnet due to being /32 subnet masked. This means they **cannot use the "devices that are on the same subnet" part of rule 3.**

c)

```
mininet> iperfudp 10M itWS facultyPC
*** Iperf: testing UDP bandwidth between itWS and facultyPC
*** Results: ['10M', '10.5 Mbits/sec', '10.5 Mbits/sec']
mininet> iperfudp 10M dnsServer itWS
*** Iperf: testing UDP bandwidth between dnsServer and itWS
*** Results: ['10M', '10.5 Mbits/sec', '10.5 Mbits/sec']
mininet> iperfudp 10M facultyWS labWS
*** Iperf: testing UDP bandwidth between facultyWS and labWS
*** Results: ['10M', '10.6 Mbits/sec', '10.5 Mbits/sec']
mininet> iperfudp 10M itWS itPC
*** Iperf: testing UDP bandwidth between itWS and itPC
*** Results: ['10M', '10.5 Mbits/sec', '10.5 Mbits/sec']
mininet> iperfudp 10M trustedPC1 dnsServer
*** Iperf: testing UDP bandwidth between trustedPC1 and dnsServer
^C
Interrupt
mininet> iperfudp 10M trustedPC2 guest
*** Iperf: testing UDP bandwidth between trustedPC2 and guest
^C
Interrupt
mininet>
```

Q6

a) Describe and discuss your test

My test to verify rule 4 consisted of:

- three iperf requests to prove each device can reach the printer using TCP,
- a ping request to prove that they do not bypass Rule 2, which states that the printer only uses TCP,
- as well as another three iperf requests from the devices to prove they do not have TCP access to other devices on the Faculty LAN.

This test successfully proves access to the Printer using only TCP, while also proving that the rule does not allow further access than the printer device.

b)

```
mininet> iperf guest printer
*** Iperf: testing TCP bandwidth between guest and printer
*** Results: ['82.0 Gbits/sec', '81.6 Gbits/sec']
mininet> iperf trustedPC1 printer
*** Iperf: testing TCP bandwidth between trustedPC1 and printer
*** Results: ['82.9 Gbits/sec', '82.6 Gbits/sec']
mininet> iperf trustedPC2 printer
*** Iperf: testing TCP bandwidth between trustedPC2 and printer
*** Results: ['74.0 Gbits/sec', '73.9 Gbits/sec']
mininet> trustedPC1 ping printer
PING 169.233.3.20 (169.233.3.20) 56(84) bytes of data.
^C
--- 169.233.3.20 ping statistics ---
62 packets transmitted, 0 received, 100% packet loss, time 63672ms

mininet> iperf guest facultyWS
*** Iperf: testing TCP bandwidth between guest and facultyWS
^C
Interrupt
mininet> iperf trustedPC1 facultyWS
*** Iperf: testing TCP bandwidth between trustedPC1 and facultyWS
^C
Interrupt
mininet> iperf trustedPC2 facultyWS
*** Iperf: testing TCP bandwidth between trustedPC2 and facultyWS
^C
Interrupt
mininet>
```

My test for Rule 4 successfully proved that the rule functioned as intended. Printer access for Guest and TrustedPC devices is normally not allowed, but as we can see from the iperf commands, the TCP connections between the Internet devices and the Printer were successful. At the same time, the non-TCP requests, such as the ping between trustedPC1 and printer,

were properly blocked to avoid contradicting Rule 2’s statement that the printer only prints using TCP. Additionally, we can prove that Rule 4 doesn’t grant the Internet devices any extra access to the Faculty subnet by showing iperf attempts, such as between guest and facultyWS.

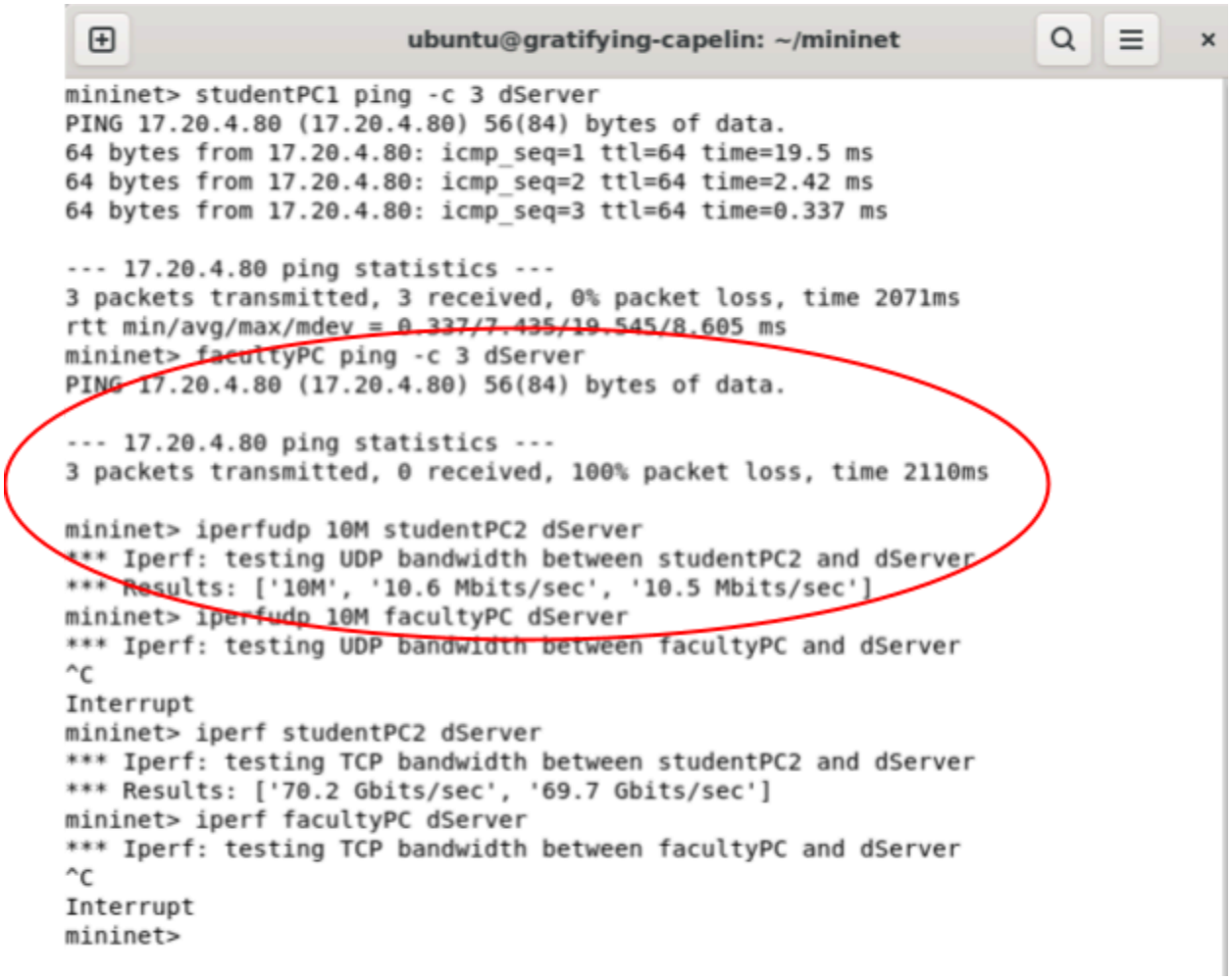
Q7

Verify Rule 5 – Describe the importance of Rule 5 and why it is needed

I verified rule 5 by simply running any of the failed tests above. Any request that wasn't explicitly allowed was dropped as a result of Rule 5. Rule 5 is needed because it ensures that only traffic that is explicitly permitted into the network is allowed. Without rule 5, we would need to instead make case-by-case rules. This would have to cover many more options than our current model which instead blocks all traffic that isn't permitted.

Q8

The Discord Server is isolated from the other subnets by only explicitly allowing traffic from the Student webserver. This is different from the other 5 rules' implementation because, rather than checking the type of traffic and filtering from there, we simply use the source and destination IP addresses to determine whether or not to allow the traffic.



As we can clearly see in the results of our tests, the firewall successfully blocked any traffic between the dServer and any non-Student subnets. When attempting to ping the dServer as a facultyPC, we had 100% packet loss, as shown in the red circle above. On the other hand, any traffic between a student PC and the dServer was successful, be it the ping or either version of iperf.