**Lab#2**

Q1[50 points] Follow the instruction of FlowVisor Lab on Canvas. Capture your screen

to show the success of your lab. Answer the quizzes on the link at

https://github.com/onstutorial/onstutorial/wiki/Flowvisor-Exercise

[Quiz 1 ] Could you have implemented this kind of slicing in a traditional network?

Highlight the hidden text below for the answer.

[Quiz 2 ] Could this division of the network have been implemented on a more traditional

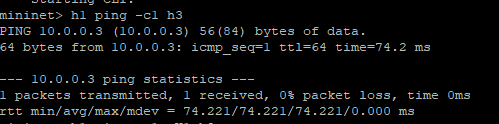
network?

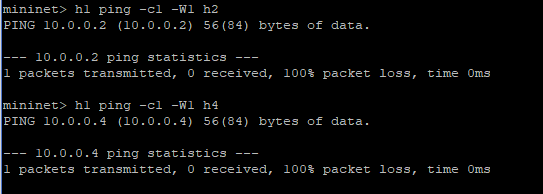
[Quiz 3] If static ARP entries were not set, which slice would have handled the ARP

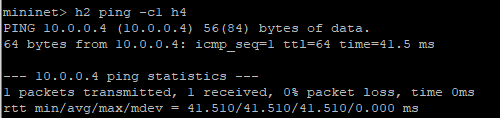
traffic? (Hint: The way the flowvisor handles ARPs is that it takes an ARP request "who

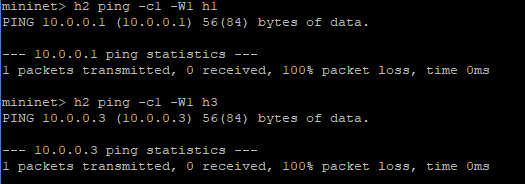
has $ip? tell $mac" and sends that to the slice that has flowspace for nw\_dst=$ip.)

**Part 1:**





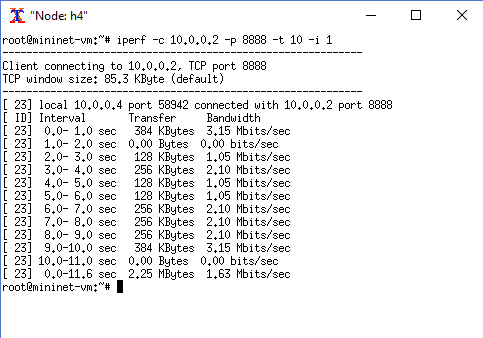


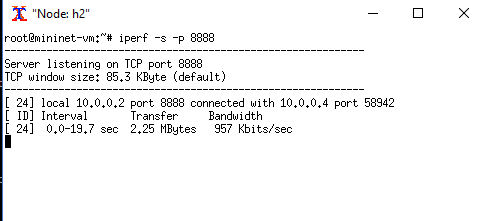


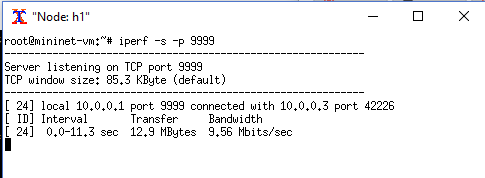
1. Could you have implemented this kind of slicing in a traditional network? Highlight the hidden text below for the answer.

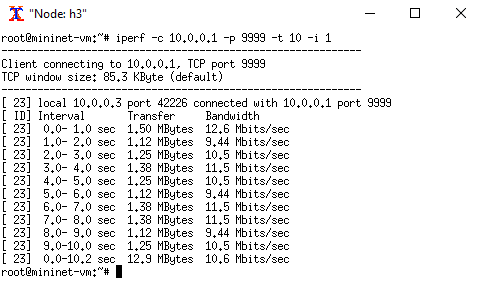
* Answer: Yes, with a VLAN! Port-based slicing is not so new.

**Part 2:**









2) Could this division of the network have been implemented on a more traditional network?

* Short answer: Of course, SDN doesn't do anything you couldn't do before.
* Long answer: In this case, an administrator might manually define access control rules to direct traffic to the video server or video port along a non-default path, logging into each switch. The difference is that with SDN, and in particular FlowVisor, those policies can be more flexible, as the controller for the video slice can make more dynamic traffic-routing and prioritization decisions for more dynamically defined slices of traffic.

3) If static ARP entries were not set, which slice would have handled the ARP traffic? (Hint: The way the flowvisor handles ARPs is that it takes an ARP request "who has $ip? tell $mac" and sends that to the slice that has flowspace for nw\_dst=$ip.)

* Answer: The ARP traffic would still be handled by the non-video slice.

Q2[50 points] Set up GRE & VxLAN Tunnels between two different Mininet

topologies. Please see the attached manual (Setting GRE Tunnels) to create Figure 1. To

be able to connect two topologies together (tunneling together) that are created using two

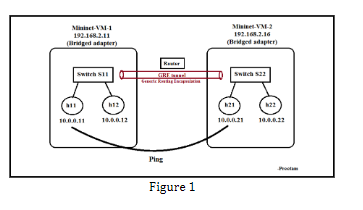
different Mininet-VMs. Essentially, be able to ping a host which is located in one

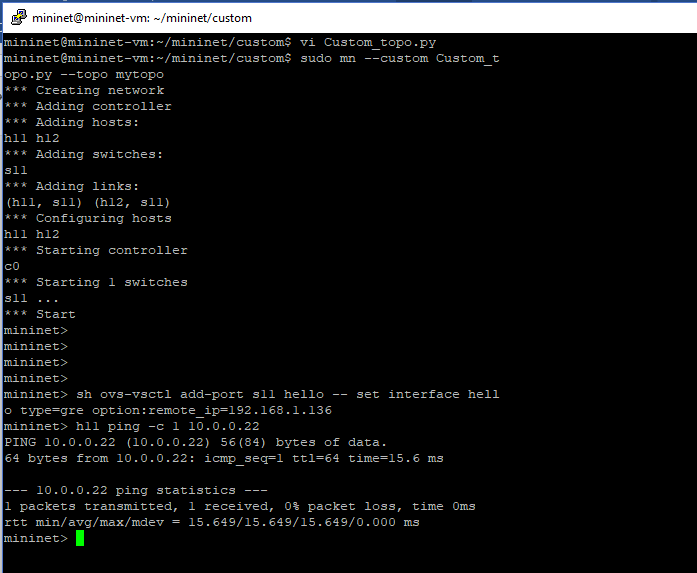
Mininet-VM using another host which is in a different Mininet-VM. A diagram to

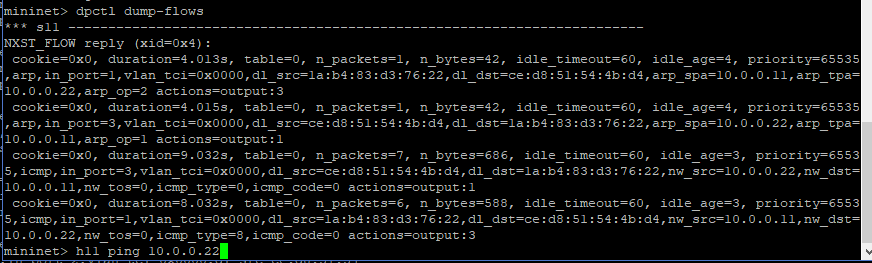
visualize this scenario is given in Figure 1. Please capture a screenshot for your setup and

dump the flow rules. Please print out and explain the flow rules when traffic is sent from

h11 to h22.







Explanation: In the above screenshot, that I have captured four flow rules.

In first flow rule:

cookie=0x0 is the opaque controller-issued identifier. This field is mainly used for MODIFY or DELETE flow rules, as this is a new flow value is set to 0 (OFPFC\_ADD).

Duration=4.013s is the duration time flow has been alive in seconds

table=0 This is the table id from where flow came from.

n\_packets=1 this is the number of packets in the flow.

n\_bytes=42 this is the number of bytes in the flow.

idle\_timeout=60 After this timeout, flow entry will be deleted if idle

idle\_age=4 After this age, flow entry will be deleted if idle

priority=65535 Priority of the entry. This is used when there is not an exact-match entry.

Arp

in\_port=1 This flow rule will be applied to ingress traffic from port-1

vlan\_tci=0x0000 It is a vlan tag control identifier. It is set to zero.

dl\_src=la:b4:83:d3:76:22 This field shows source mac address.

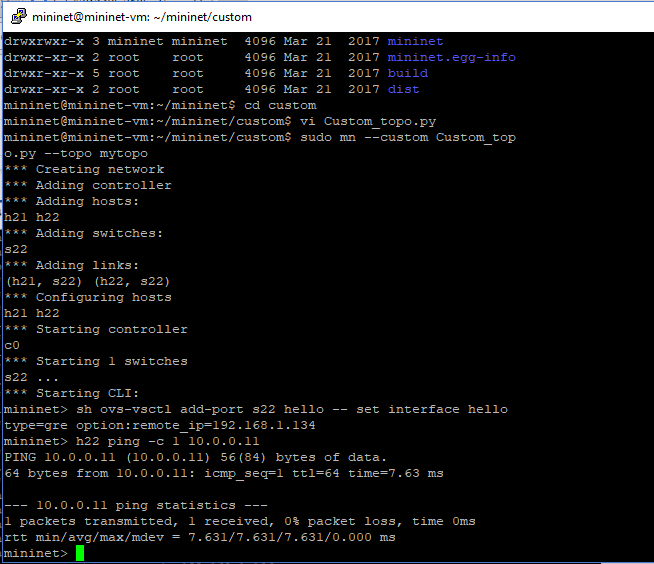
dl\_dst=ce:d8:51:54:4b:d4 This field shows destination mac address.

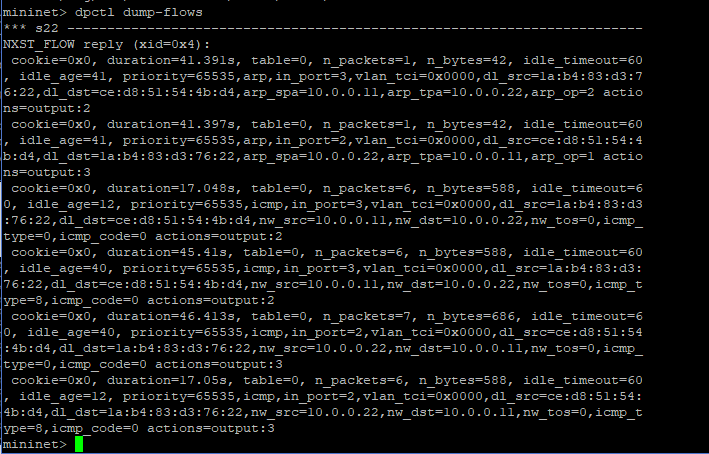
arp\_spa=10.0.0.11 Source IP address of arp

arp\_tpa=10.0.0.22 Target IP address of arp

arp\_op=2 Opcode for arp.

actions=output:3 All traffic will be sent to port 3





Explanation: In the above screenshot, that I have captured six flow rules.

In first flow rule:

cookie=0x0 is the opaque controller-issued identifier. This field is mainly used for MODIFY or DELETE flow rules, as this is a new flow value is set to 0 (OFPFC\_ADD).

Duration=41.391s is the duration time flow has been alive in seconds

table=0 This is the table id from where flow came from.

n\_packets=1 this is the number of packets in the flow.

n\_bytes=42 this is the number of bytes in the flow.

idle\_timeout=60 After this timeout, flow entry will be deleted if idle

idle\_age=41 After this age, flow entry will be deleted if idle

priority=65535 Priority of the entry. This is used when there is not an exact-match entry.

Arp

in\_port=3 This flow rule will be applied to ingress traffic from port-1

vlan\_tci=0x0000 It is a vlan tag control identifier. It is set to zero.

dl\_src=la:b4:83:d3:76:22 This field shows source mac address.

dl\_dst=ce:d8:51:54:4b:d4 This field shows destination mac address.

arp\_spa=10.0.0.11 Source IP address of arp

arp\_tpa=10.0.0.22 Target IP address of arp

arp\_op=1 Opcode for arp.

actions=output:3 All traffic will be sent to port 3