SDNFV FINAL PROJECT

SDN Network as Virtual Router

Deadline: 2024/01/11

Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
- Supplements
- Scoring Criteria
- Reference

Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
- Supplements
- Scoring Criteria
- Reference

Review of Labs

- Lab3 - SDN-enabled Learning Bridge
  - Mac learning
- Lab4 – Unicast DHCP Application
  - Use intents to forward DHCP packets
  - Configure ONOS application via onos-netcfg
- Lab5 – Proxy ARP
  - Construct packets and packet-out to edge ports
- Lab6 – Network Function Virtualization
  - Use Quagga and Docker to simulate Autonomous Systems (AS)

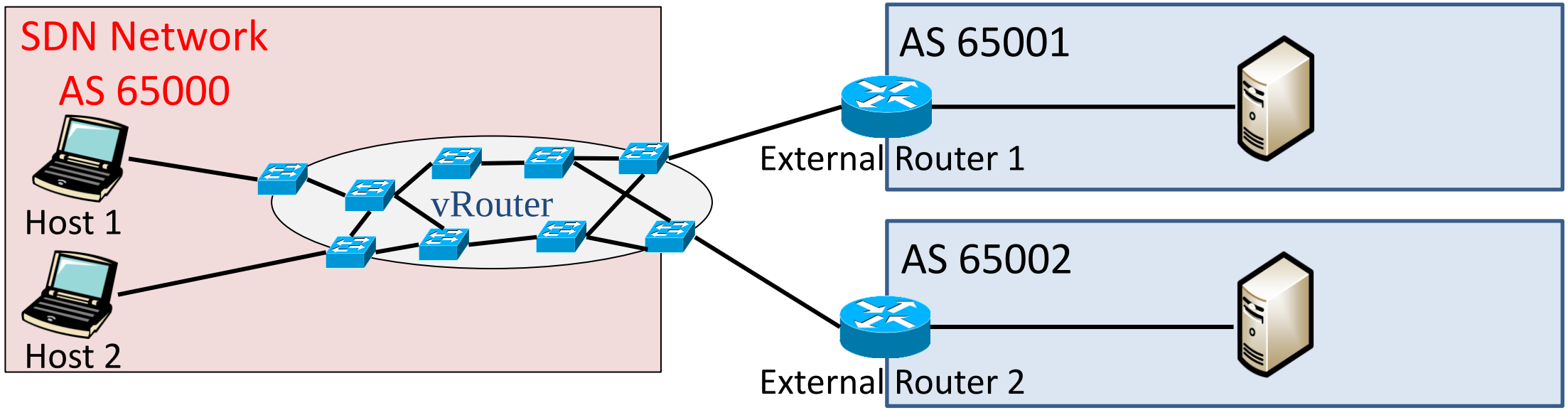
- Note: All of these labs would be used in final project

Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
- Supplements
- Scoring Criteria
- Reference

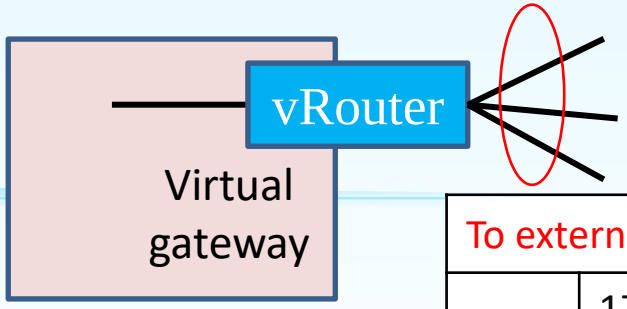
SDN Virtual Router

- SDN network as a router
  - Use OpenFlow switches and flow rules to simulate router behavior
  - For instance:
    - Route exchange
    - Layer 2 modification
    - Etc.

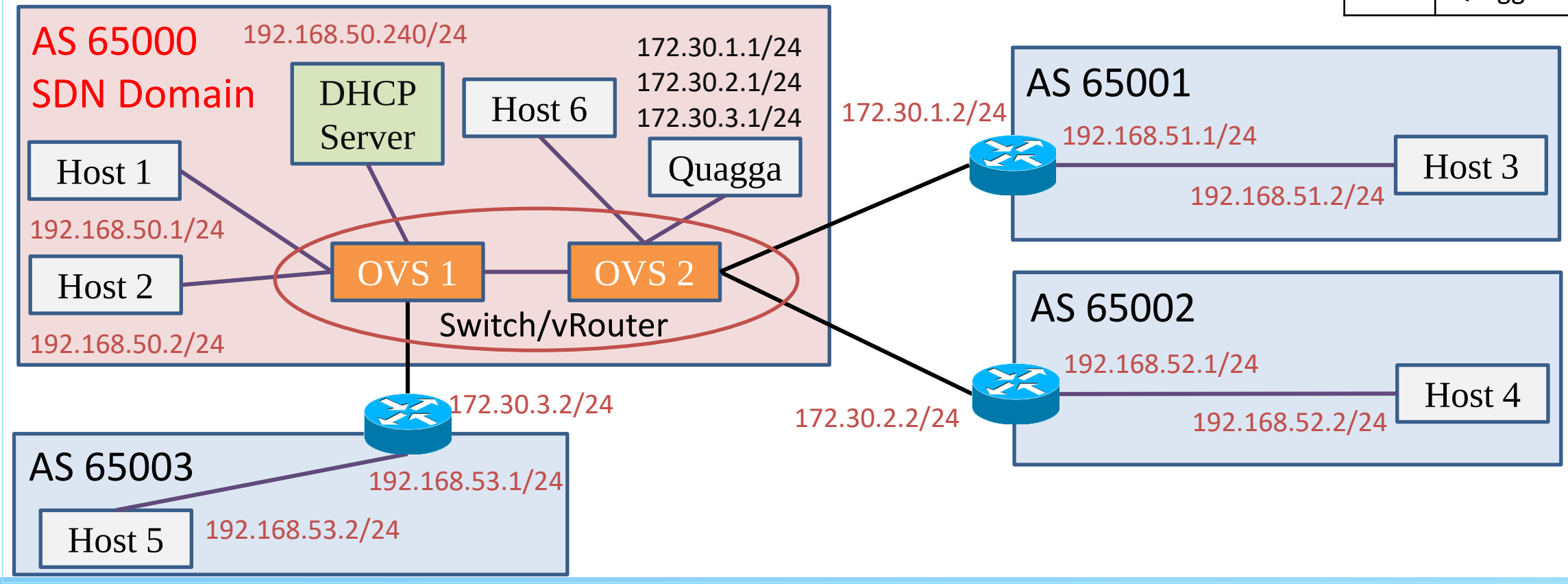


Sample Topology

- Virtual gateway IP: 192.168.50.254
- Virtual gateway MAC: 00:00:00:00:00:01
- Use Quagga MAC as external interface MAC of vRouter



To external routers	
IP:	172.30.1.1/24 172.30.2.1/24 172.30.3.1/24
MAC:	Quagga MAC

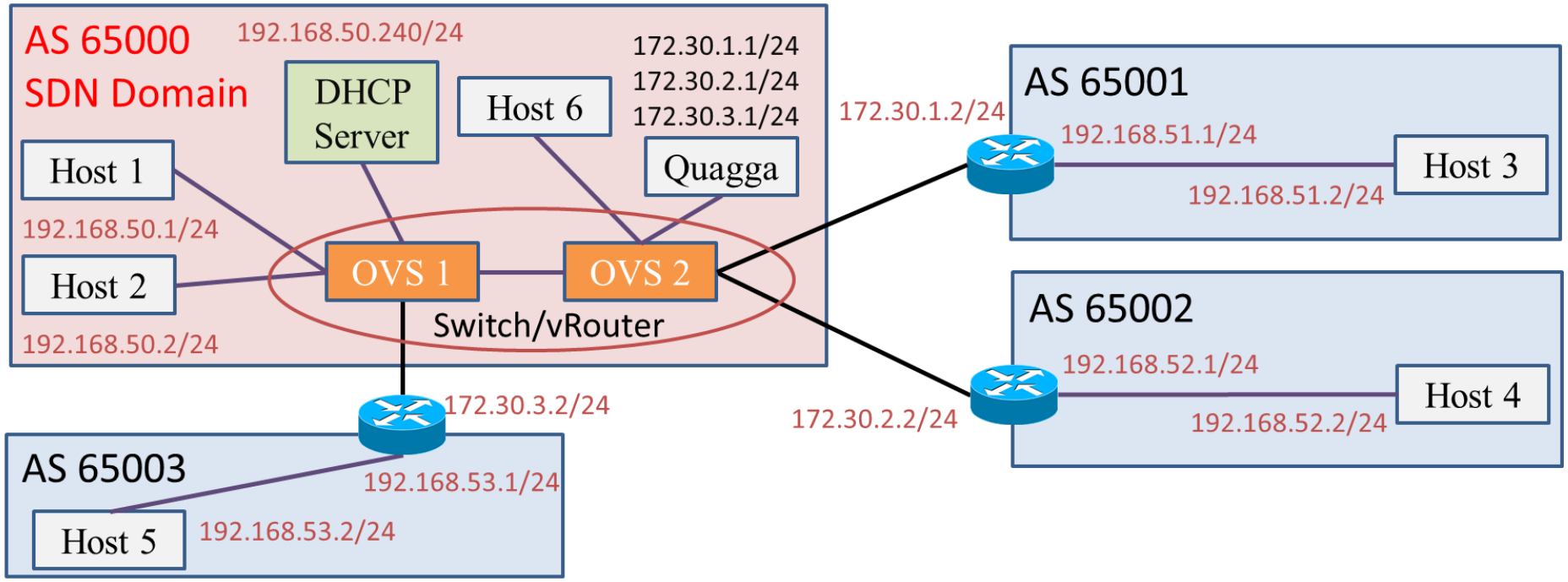


Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
- Supplements
- Scoring Criteria
- Reference

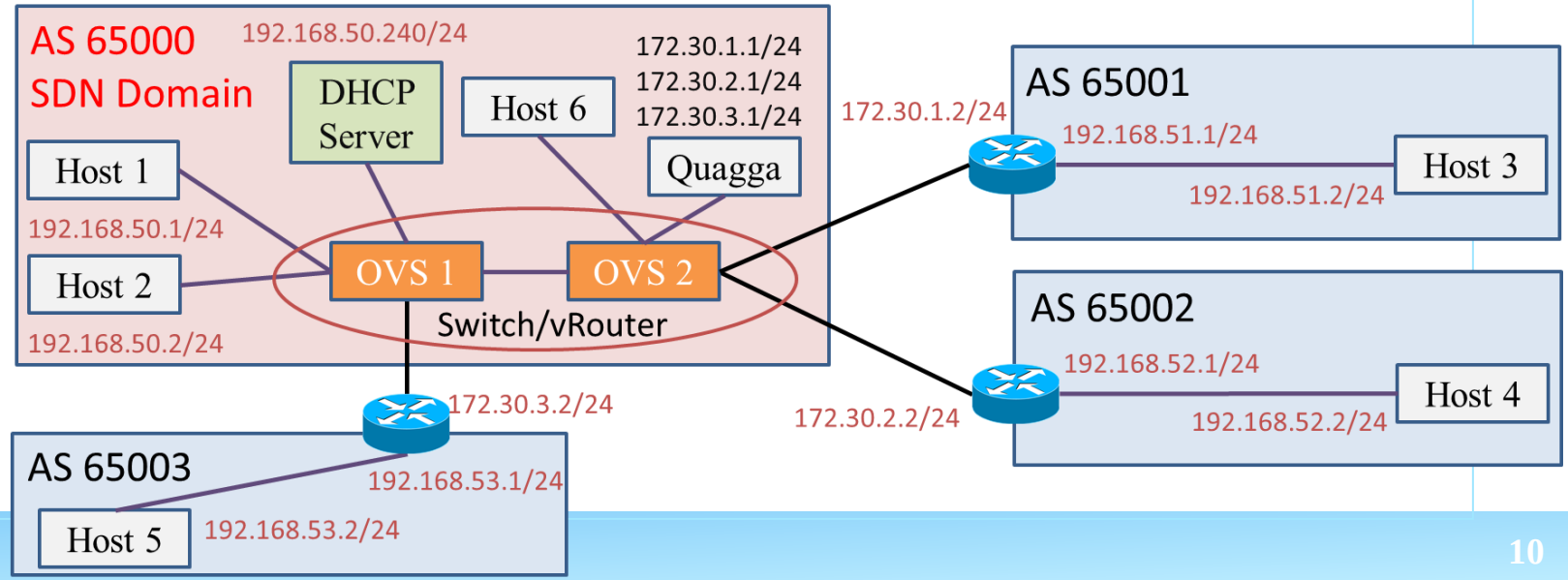
Goal

- Intra domain host communications
  - Handled by bridge APP
- Inter domain host communications
  - SDN domain <=> Other domains
  - One domain <=> SDN domain <=> Another domain



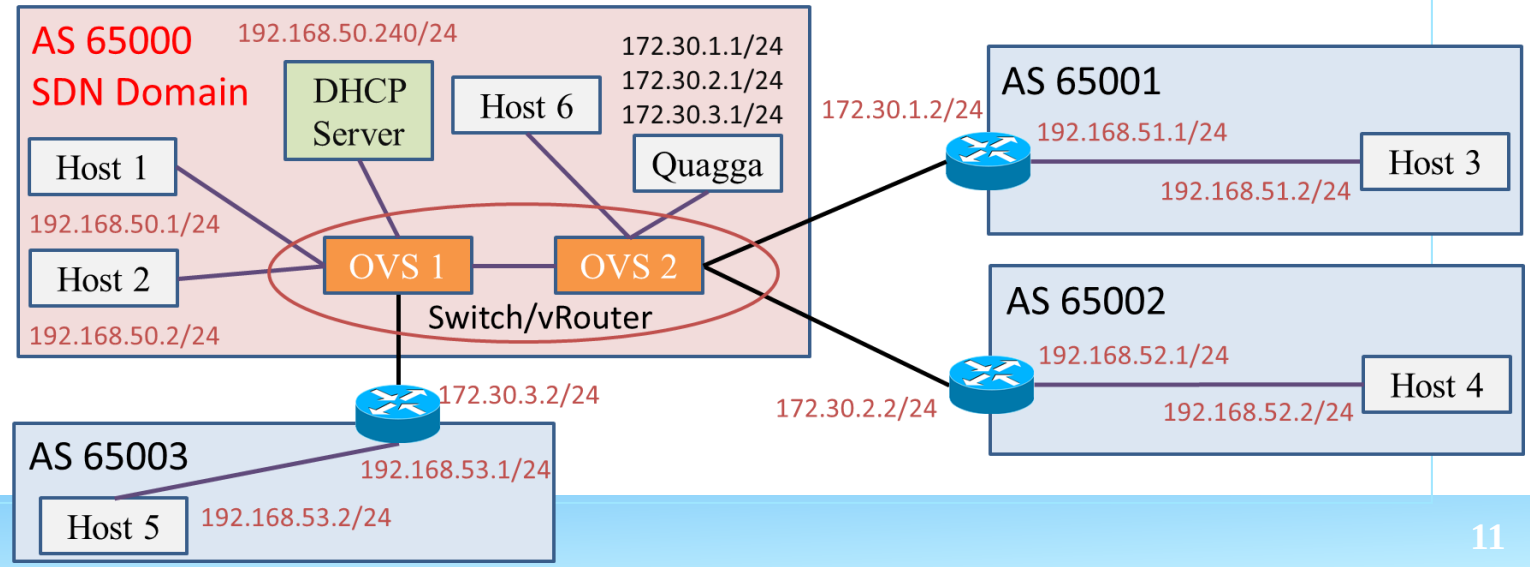
vRouter Specification

- Intra AS packet forwarding and packet-in request
  - Lab3
- DHCP support for devices in AS
  - Lab4
- ARP reply for all devices in SDN Domain
  - Lab5
- Routing table maintenance
  - Lab6
- Create flow rules for intra/inter domain traffic
  - vRouter app



Supplements

1. 4 scripts for topology construction/destruction
   - Built by Docker and OVS
   - All Dockers are configured, including IP addresses and quagga config files
2. A sample ONOS APP config file
   - Configurations
     - Interface Service config
     - Virtual gateway IP and MAC
     - BGP peers
     - DHCP server location
3. Compiled oar files, including
   - Bridge App
   - UnicastDHCP App
   - ProxyARP App



vRouter Project TODO List

- Configure vRouter using onos-netcfg
- Route exchange
  - Forward external router's eBGP packet to Quagga and vice versa (using intent)
- Route decision
  - Decide next hop using information collected from Quagga
- Gateway function
  - L2 modification for inter AS communication

Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
  - Zebra FIB
  - ONOS Interface Service
  - ONOS Route Service
- In Used App Configurations
- Virtual Router Workflow
- Supplements
- Scoring Criteria
- Reference

Zebra FIB Pushing

- Zebra supports a 'Forwarding Information Base (FIB) push' interface (FPI)
  - FPI allows an **external component** to **learn** the forwarding information
- Forwarding Plane Manager (FPM)
  - Receives FIB
  - Decode FIB into routes

✓ FIB pushing
  - FPM establishes a TCP connection with Zebra
  - Zebra pushes FIB to FPM

- In this project, we use ONOS built in FPM to collect routing information from zebra

```
mikoto@root > app activate org.onosproject.fpm
```

Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
  - Zebra FIB
  - ONOS Interface Service
  - ONOS Route Service
- In Used App Configurations
- Virtual Router Workflow
- Supplements
- Scoring Criteria
- Reference

ONOS Interface Service

- A storage service for APPs to decide how to select and forward network traffic
- Manually assign a subnet / connection point mapping for query

```
{
  "ports": {
    "of:0000000000000004/2": {
      "interfaces": [
        {
          "name": "intf1",
          "ips": [
            "172.30.1.1/24"
          ]
        }
      ]
    }
  }
}
```

Connection Point

Interface Config

ONOS Interface Service

- Query for interface info using Interface Service

```
import org.onosproject.net.intf.Interface;
import org.onosproject.net.intf.InterfaceService;

@Reference(cardinality = ReferenceCardinality.MANDATORY)
protected InterfaceService intfService;

Interface outIntf = intfService.getMatchingInterface(IP4Address("172.30.1.1/24"));
```

Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
  - Zebra FIB
  - ONOS Interface Service
  - ONOS Route Service
- In Used App Configurations
- Virtual Router Workflow
- Supplements
- Scoring Criteria
- Reference

ONOS Route Service

- Collect routing table via FPM APP
- Provide API to query routing table
- Each entry contains next hop info for target subnet

```
mikoto@root > routes
B: Best route, R: Resolved route

Table: ipv4
B R  Network          Next Hop      Source (Node)
> *  192.168.51.0/24   172.30.1.2    FPM (127.0.0.1)
> *  192.168.52.0/24   172.30.1.3    FPM (127.0.0.1)
> *  192.168.53.0/24   172.30.2.2    FPM (127.0.0.1)
  Total: 3
```

- Query next hop using RouteService

```
import org.onosproject.routeservice.RouteService;

@Reference(cardinality = ReferenceCardinality.MANDATORY)
protected RouteService routeService;

//getRouteTables() returns a set of iterable route entries
routeService.getRouteTables()
```

Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
- Supplements
- Scoring Criteria
- Reference

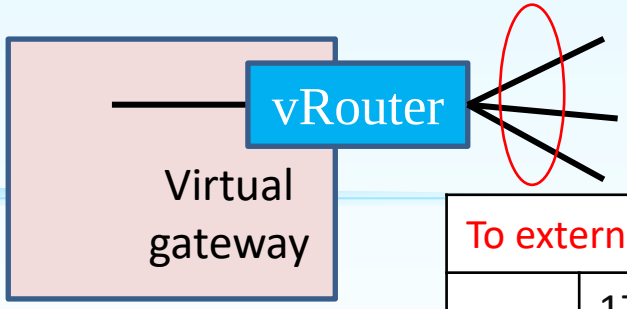
App Config File Attributes

- quagga:
  - Connection point of Quagga
- quagga-mac
  - MAC address of Quagga
- virtual-ip
  - Virtual gateway IP
- virtual-mac
  - Virtual gateway MAC
- peers:
  - BGP peering

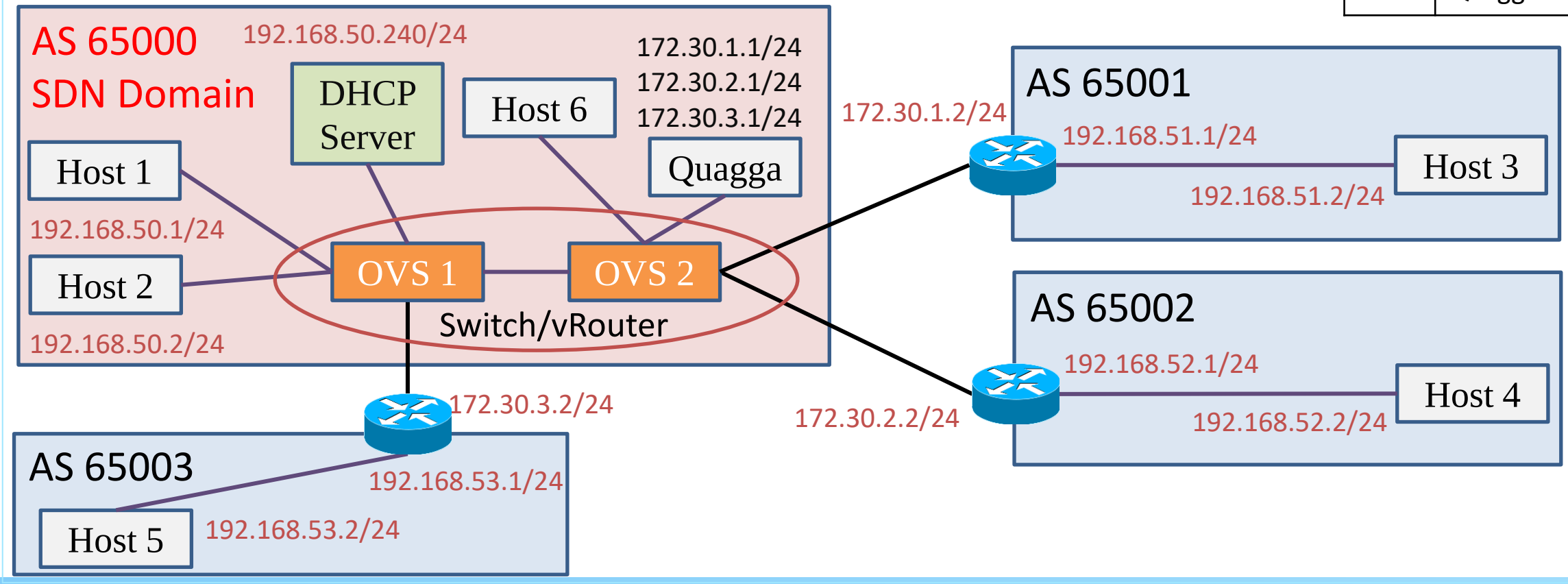
```
"apps": {
    "nycu.sdnfv.vrouter": {
        "router": {
            "quagga": "of:0000000000000002/4",
            "quagga-mac": "YOUR QUAGGA'S MAC",
            "virtual-ip": "192.168.50.254",
            "virtual-mac": "00:00:00:00:00:01",
            "peers": [
                "172.30.1.2",
                "172.30.2.2",
                "172.30.3.2"
            ]
        }
    }
}
```

Sample Topology

- Virtual gateway IP: 192.168.50.254
- Virtual gateway MAC: 00:00:00:00:00:01
- Use Quagga MAC as external interface MAC of vRouter



To external routers	
IP:	172.30.1.1/24 172.30.2.1/24 172.30.3.1/24
MAC:	Quagga MAC



Outline

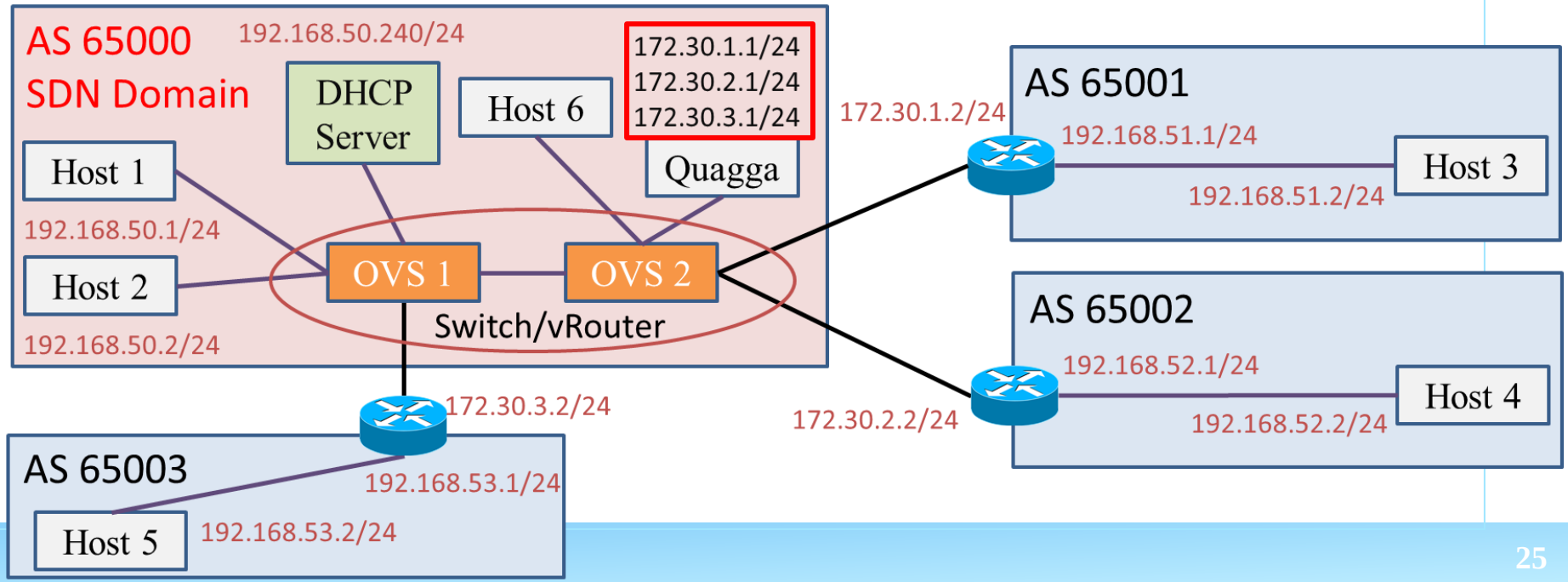
- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
  - Inter domain traffic
    - BGP traffic
    - SDN-external traffic
    - Transit traffic
- Supplements
- Scoring Criteria
- Reference

Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
  - Inter domain traffic
    - BGP traffic
    - SDN-external traffic
    - Transit traffic
- Supplements
- Scoring Criteria
- Reference

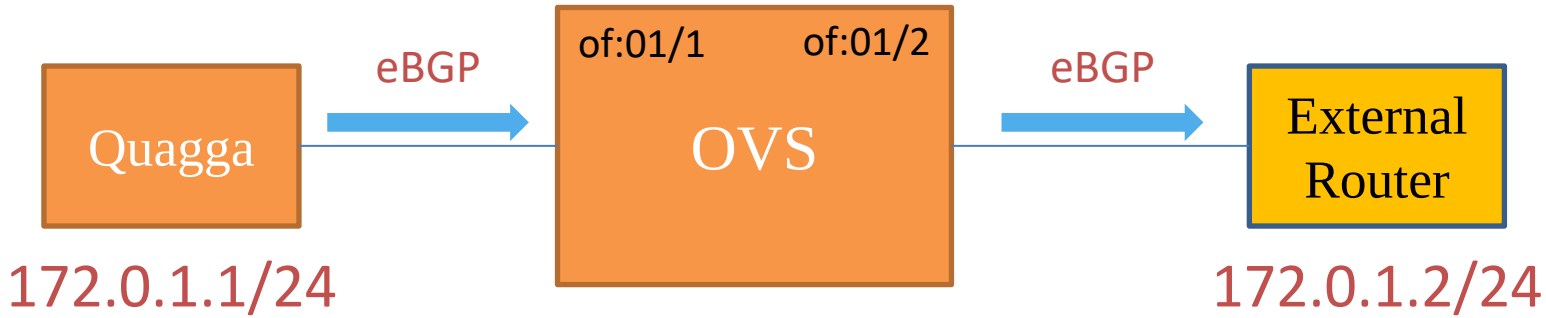
Virtual Router Configuration and BGP Peering

- Virtual router IP and MAC addresses:
  - IP addresses: one for each external interface
    - 172.0.1.1, 172.0.2.1, 172.0.3.1
  - MAC address: a single MAC for all external interfaces
    - Use quagga's MAC address
- Proxy ARP app handles ARP on behalf of vRouter



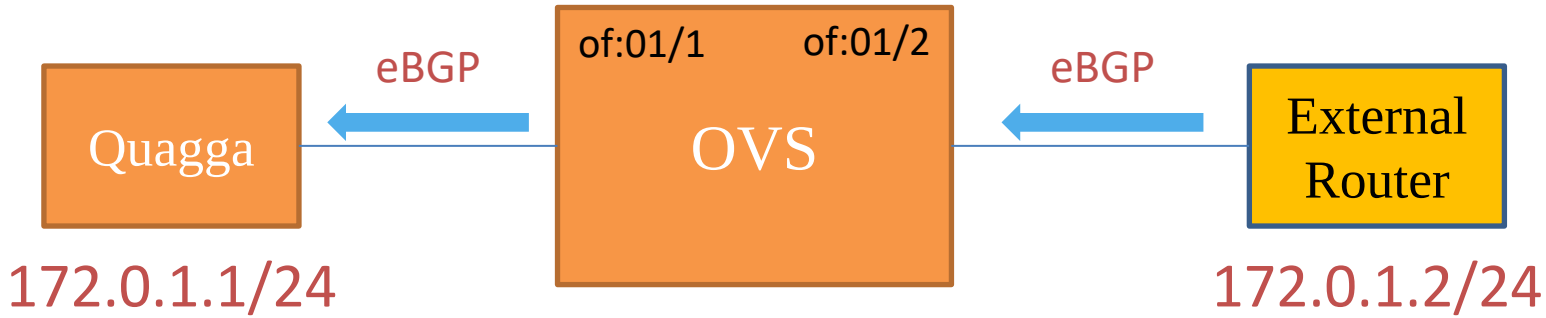
Flow rules for BGP Peering – Outgoing eBGP

- Install flow rules for eBGP via PointToPointIntent
  - Ingress point: Quagga's connect point
    - From configuration file
  - Filter: destination IP = external router's IP
    - From configuration file
  - Egress point: external interface
    - Via querying Interface Service



Flow rules for BGP Peering – Incoming eBGP

- Install flow rules for eBGP via PointToPointIntent
  - Ingress point: external interface
    - Via querying Interface Service
  - Filter: destination IP = Quagga's IP
  - Egress point: Quagga's connect point
    - From configuration file

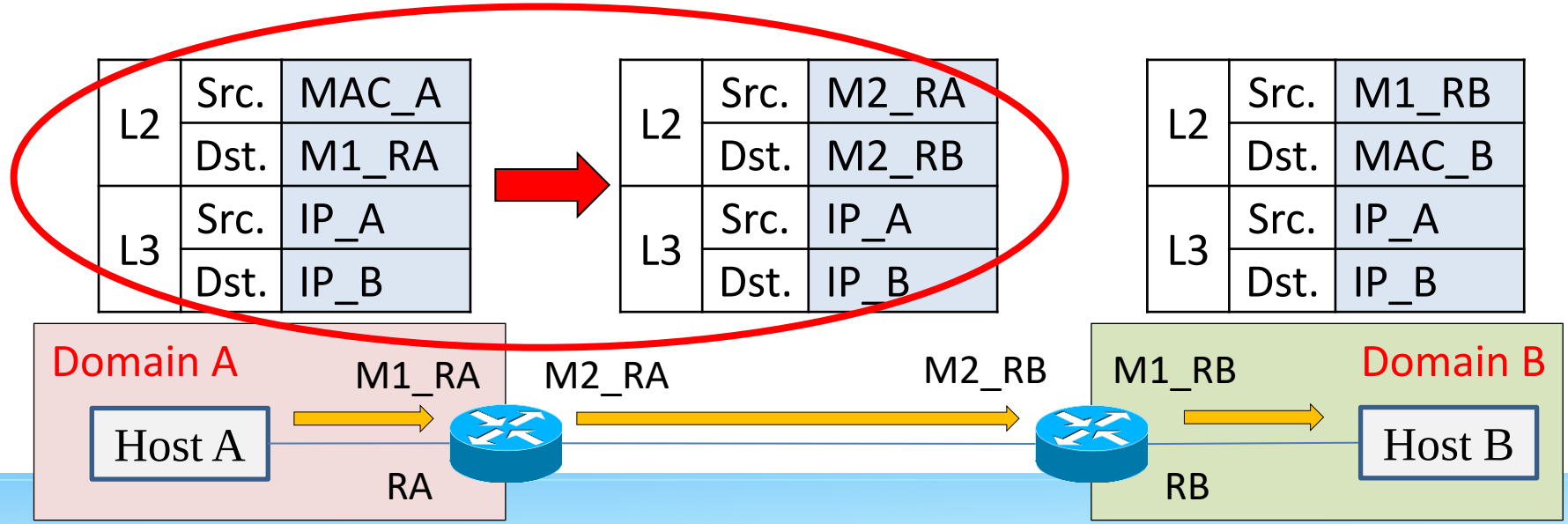


Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
  - Inter domain traffic
    - BGP traffic
    - SDN-external traffic
    - Transit traffic
- Supplements
- Scoring Criteria
- Reference

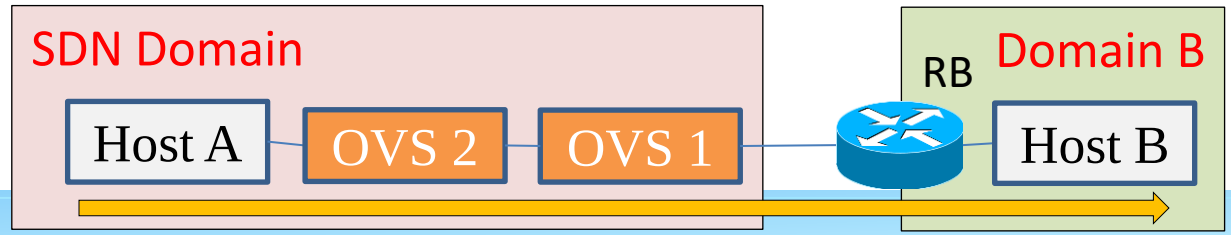
Legacy Router Workflow – Outgoing Packets

- Host A sends IPv4 packets to external Host B
  - Host A sends ARP request for gateway MAC
  - Host A construct packet, using
    - MAC_A as source MAC
    - M1_RA as destination MAC
    - IP_A as source IP
    - IP_B as destination IP
- RA performs **Layer 2 modification** on packet



Packet Workflow – SDN to External

- Host A in SDN domain sends pkts to external Host B, assuming flow rule not exists
  - Host A sends ARP request for gateway MAC
  - ProxyARP replies gateway MAC
- If vRouter knows route to destination IP, it installs flow rules using Intent Service with criteria
  - Ingress CP: packet-in port
  - Egress CP: next hop connect point (RB)
    - Query from Interface Service
  - Filter: destination IP = host B's IP Layer 2 header modification
  - Modify source MAC to Quagga's MAC
    - Query from configuration file
  - Modify destination MAC to next hop's MAC
    - Query from Host Service
- Otherwise
  - No-op



Legacy Router Workflow – Incoming Packets

- RA receives packets from RB
- RA performs Layer 2 modification on packets
- RA forward packets to Host A

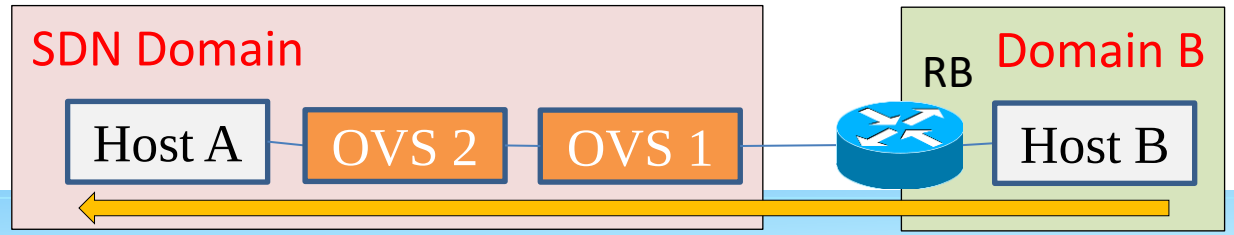
L2	Src.	M1_RA
	Dst.	MAC_A
L3	Src.	IP_B
	Dst.	IP_A

L2	Src.	M2_RB
	Dst.	M2_RA
L3	Src.	IP_B
	Dst.	IP_A

L2	Src.	MAC_B
	Dst.	M1_RB
L3	Src.	IP_B
	Dst.	IP_A

Packet Workflow – External to SDN

- External router (RB) forwards packets to SDN domain
  - RB sends ARP request for next hop's (Quagga) MAC
  - ProxyARP replies Quagga's MAC
- If vRouter knows Host A, it installs flow rules using Intent Service with criteria
  - Ingress CP: packet-in port
  - Egress CP: host connect point
    - Query from Host Service
  - Filter: destination IP = host A's IP Layer 2 header modification
  - Modify source MAC to virtual gateway's MAC
    - Query from configuration file
  - Modify destination MAC to host A's MAC
    - Query from Host Service
- Otherwise
  - No-op

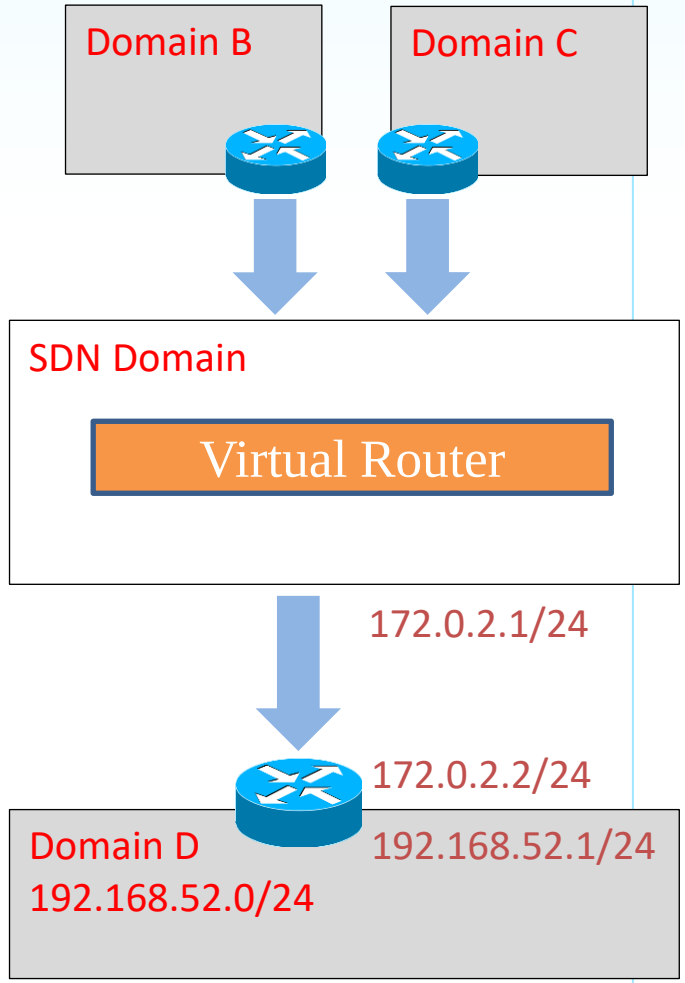


Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
  - Inter domain traffic
    - BGP traffic
    - SDN-external traffic
    - Transit traffic
- Supplements
- Scoring Criteria
- Reference

Packet Workflow – External to External

- Suppose vRouter already learned routes from FPM
  - E.g., route: 192.168.52.0/24 next hop 172.0.2.2
- vRouter installs rules for each transit **route entry**
  - Via MultiPointToSinglePoint intent, with criteria
    - Ingress CPs: all BGP peer connection points
      - Query from configuration file and Interface Service
    - Filter: destination subnet in transit **route entry**
    - Egress CP: next hop interface
      - Query from Interface Service
    - Modify source MAC to Quagga's MAC
      - Query from configuration file
    - Modify destination MAC to next hop router's MAC
      - Query from Host Service



Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
- Supplements
  - Provided Files
  - Building Docker Image
  - Build/Clean Up Topology
  - Start/Stop DHCP server
  - Testing/Debugging
- Scoring Criteria
- Reference

Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
- Supplements
  - Provided Files
  - Building Docker Image
  - Build/Clean Up Topology
  - Start/Stop DHCP server
  - Testing/Debugging
- Scoring Criteria
- Reference

Provided Files

- In a supplementary folder

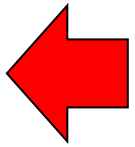
```
$ tree
sdnfv_final_supplement/
   ├── bgp_confs/       // quagga configuration files
   ├── build_topo.sh    // script to build demo topology
   ├── clean_topo.sh    // script to kill and clean up demo topology
   ├── dhcp_start.sh    // script to start DHCP server
   ├── dhcp_init.sh     // script only need to run once on init
   ├── dhcpd.conf       // DHCP server config
   ├── clone_quagga.sh // script to clone “quagga” from GitHub
   ├── host/            // host docker Dockerfile
   ├── config.json      // config for virtual router app
   └── target/          // provided oar files
```

Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
- Supplements
  - Provided Files
  - Building Docker Image
  - Build/Clean Up Topology
  - Start/Stop DHCP server
  - Testing/Debugging
- Scoring Criteria
- Reference

Build Docker Image from Dockerfile

- Dockerfile contains workflows to create a Docker image
- First, clone “quagga” from GitHub with the script
  - `$ ./clone_quagga.sh`
- In “host” and “quagga” folder
  - Contains Dockerfile to build Docker image
- To build Docker images:
  - In “host” folder:
    - `$ docker build -t host-mano .`
  - In “quagga” folder:
    - `$ docker build -t quagga-fpm .`



Don’t forget this dot!

Host Docker

- Host Docker image provides basic debugging tools
  - ping
  - arping
  - etc.

- If you need any other tools
  - Modify Dockerfile
  - Rebuild Docker image

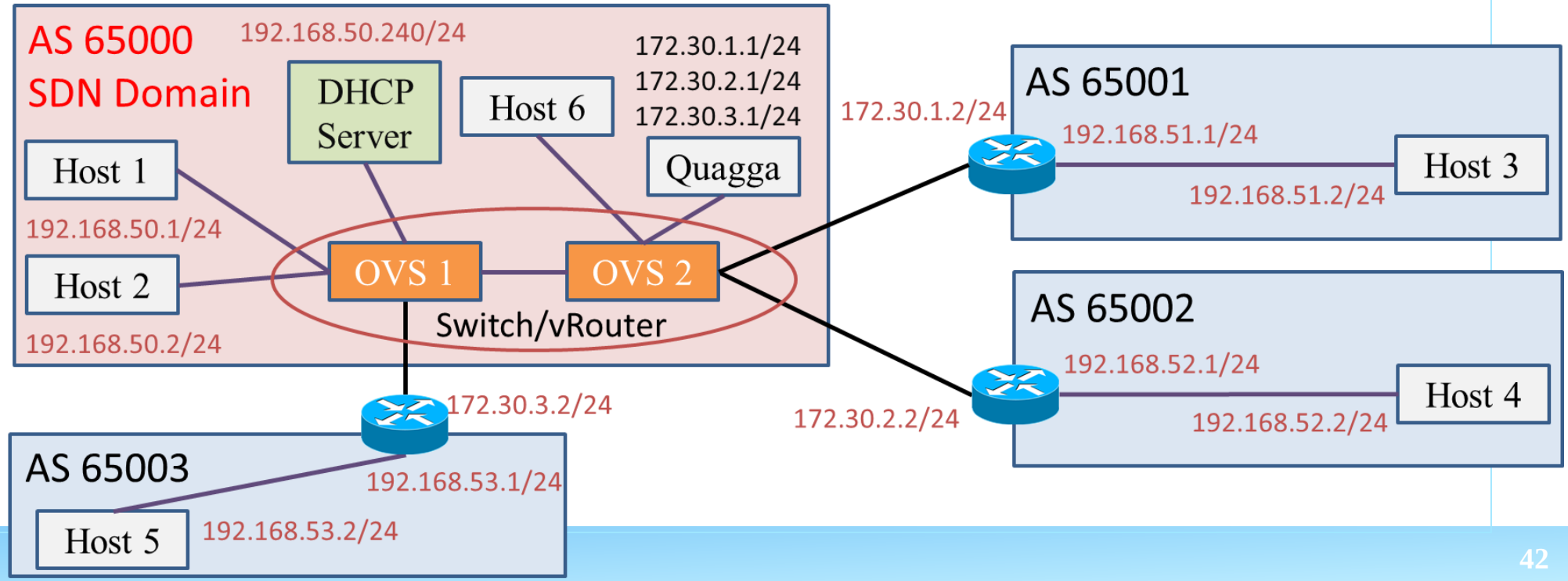
Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
- Supplements
  - Provided Files
  - Building Docker Image
  - Build/Clean Up Topology
  - Start/Stop DHCP server
  - Testing/Debugging
- Scoring Criteria
- Reference

Build/Clean Up Topology

- `$ sudo ./build_topo.sh`
  - This script helps you build the topology shown in page 7

- `$ sudo ./clean_topo.sh`
  - Use this script to clean up the topology



Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
- Supplements
  - Provided Files
  - Building Docker Image
  - Build/Clean Up Topology
  - Start/Stop DHCP server
  - Testing/Debugging
- Scoring Criteria
- Reference

Start/Stop DHCP Server

- This script only need to execute once
  - `$ sudo ./dhcp_init.sh`
- Start DHCP Server
  - `$ sudo ./dhcp_start.sh`
- Kill DHCP Server
  - `$ sudo killall dhcpd`

Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
- Supplements
  - Provided Files
  - Building Docker Image
  - Build/Clean Up Topology
  - Start/Stop DHCP server
  - Testing/Debugging
- Scoring Criteria
- Reference

Enabling FPM and Route Service API

- Use FPM app for route rule collection

- Modify pom.xml
  - Enable FPM APP before vRouter APP start

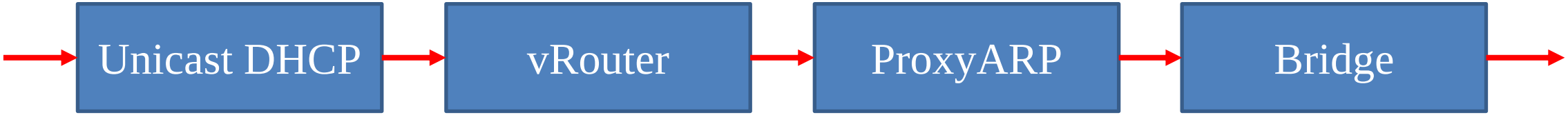
```
<properties>
        <onos.app.requires>org.onosproject.fpm</onos.app.requires>
</properties>
```

  - Setup Route Service API dependencies

```
<dependency>
        <groupId>org.onosproject</groupId>
        <artifactId>onos-apps-route-service-api</artifactId>
        <version>2.7.0</version>
</dependency>
```

Packet Processor Priority

- Packets will pass through each processor
  - From low priority to high priority
- Packet “handled” manifest
  - Call context.send() or context.block() to mark a packet context as “handled”
  - context.send() can only be called once
  - Subsequent packet processors may check “handled” mark and process packet accordingly
- To avoid bridge APP handling all traffic
  - Call context.block() on all processed packets



Packet Processor Priority

- Use 6 as your app's packet processor priority

```
packetService.addProcessor(processor, PacketProcessor.director(6));
```

Procedure to Setup Demo Environment

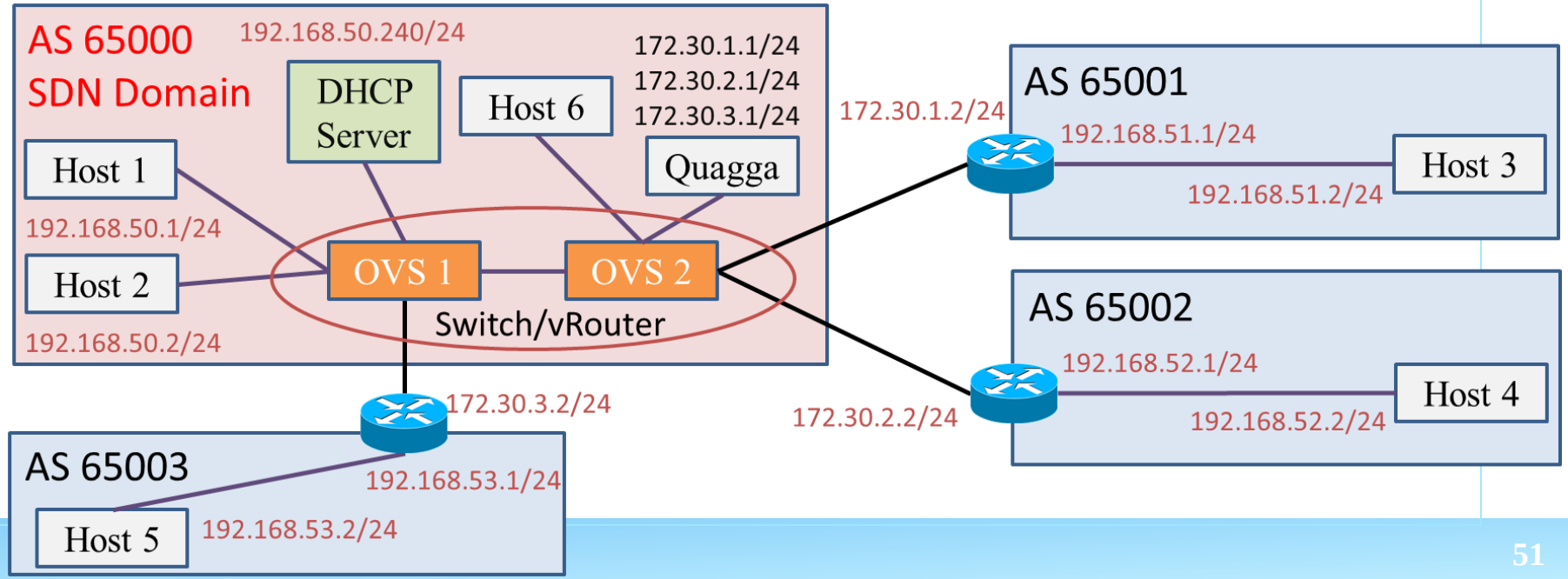
- Start ONOS
  - `$ ok clean`
- Build topology
  - `$ sudo ./build_topo.sh`
- Upload json config
  - Update quagga-mac before starting!
  - `$ onos-netcfg localhost config.json`
- Start DHCP server
  - `$ sudo ./dhcp_start.sh`

Procedure to Setup Demo Environment

- Install ONOS applications
  - `$ onos-app localhost install! target/unicastdhcp-1.0...`
  - `$ onos-app localhost install! path/to/your/routerAPP`
  - `$ onos-app localhost install! target/proxyarp-1.0...`
  - `$ onos-app localhost install! target/bridge-1.0...`

How To Test Your App

- Check your app
  - Use ONOS CLI to show routing table and check rules for eBGP traffic
  - Host 1's pings Host 2 (Intra domain traffic)
  - Host 1's pings Host 3 (Inter domain traffic)
  - Host 3's pings Host 4/5 (Transit traffic)
  - Host 6 can obtain DHCP offer and ping Host 5 (DHCP + Inter domain traffic)



Debugging

- Check current routing table
  - In ONOS cli

```
mikoto@root > routes
B: Best route, R: Resolved route

Table: ipv4
B R  Network          Next Hop       Source (Node)
> *  192.168.51.0/24   172.30.1.2     FPM (127.0.0.1)
> *  192.168.52.0/24   172.30.1.3     FPM (127.0.0.1)
> *  192.168.53.0/24   172.30.2.2     FPM (127.0.0.1)
   Total: 3
```

Debugging

- Check current interface settings
  - In ONOS cli

```
mikoto@root > interfaces
intf2: port=of:0000000000000002/4 ips=[172.30.2.1/24]
intf1: port=of:0000000000000004/2 ips=[172.30.1.1/24]
```

Packet Lost in Inter Domain Communication

- Packet processor only implement “SetOutPort” method
- Modify src, dst MAC address and packet out by context.send() won’t work

- Inter domain communication will encounter packet loss

```
root@h08:/# ping 192.168.50.2
PING 192.168.50.2 (192.168.50.2) 56(84) bytes of data.
64 bytes from 192.168.50.2: icmp_seq=2 ttl=63 time=6.27 ms
64 bytes from 192.168.50.2: icmp_seq=3 ttl=63 time=0.177 ms
^C
--- 192.168.50.2 ping statistics ---
```

- No need to handle this problem

Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
- Supplements
- Scoring Criteria
- Reference

Scoring Criteria (1/4)

- Create Makefile
  - You MUST provide your Makefile, which creates an application with a make command
  - The Makefile should be placed at the top layer of the project directory
  - TA will use your Makefile to build your application
  - If you do not provide Makefile, you will not earn any credit
- Use provided APPs
  - TA will test your vRouter with the APPs in the supplement

Scoring Criteria (2/4)

- (10%) Project naming convention
  - <groupId>: nycu.sdnfv
  - <artifactId>: vrouter
  - <version>: <use default> (1.0-SNAPSHOT)
  - <package>: nycu.sdnfv.vrouter
- (15%) Config parsing
  - Your app should parse config file correctly
- (15%) BGP traffic
  - Quagga should be able to communicate with external routers
  - We will test this item without the Bridge APP

Scoring Criteria (3/4)

- (15%) Intra domain traffic
  - Ping should work in SDN domain
- (15%) Inter domain traffic
  - Ping should work from SDN domain to external AS
- (15%) Transit traffic
  - Ping from AS 65001/65002 to AS 65003 should be forwarded correctly
- (15%) DHCP client
  - Host 6 should be able to obtain DHCP offer and able to ping AS 65001 - AS 65003

Scoring Criteria (4/4)

- Activate these apps only

```
mikoto@root > apps -a -s
*   3 org.onosproject.route-service          2.7.0    Route Service Server
*   6 org.onosproject.optical-model          2.7.0    Optical Network Model
*  31 org.onosproject.hostprovider           2.7.0    Host Location Provider
*  32 org.onosproject.lldpprovider           2.7.0    LLDP Link Provider
*  33 org.onosproject.openflow-base          2.7.0    OpenFlow Base Provider
*  34 org.onosproject.openflow               2.7.0    OpenFlow Provider Suite
*  42 org.onosproject.drivers                2.7.0    Default Drivers
*  65 org.onosproject.fpm                    2.7.0    FIB Push Manager (FPM) Route
Receiver
* 171 org.onosproject.gui2                   2.7.0    ONOS GUI2
* 176 nctu.winlab.router                    1.0.SNAPSHOT Router app
* 177 nctu.winlab.unicastdhcp               1.0.SNAPSHOT Unicast DHCP app
* 178 nctu.winlab.proxyarp                  1.0.SNAPSHOT Proxy arp app
* 179 nctu.winlab.bridge                    1.0.SNAPSHOT Bridge app
```

Submission Naming Convention

- Rename your router app directory as final_project_<StudentID>
- Compress the directory into a zip file named as final_project_<StudentID>.zip
- Upload your zip file to E3
- Wrong file name or format will result in 10 points deduction
- 20% deduction for late submission in one week
  - Won't accept submissions over one week

Demo

- TA will open a demo time reservation sheet one week before demo
- The dates will be chosen after the deadline
- Demo questions will appear at the start of the demo
- The score of demo will occupy 40% total score of this project
  - For example:
    - You earn 100% of the credits for submission
    - You earn 80% of the credits for demo
    - Then your total score of this project will be:

      100 x 60% + 80 x 40% = 92

About help!

- For any project problem
  - Ask at the E3 forum
  - TAs will help to clarify project contents instead of giving answers!
  - Please describe your questions with sufficient context
    - E.g., Environment setup, Input/Output, Screenshots, …
- For personal problems, mail to sdnta@win.cs.nctu.edu.tw
  - You have problems so that you can't meet the deadline
  - You got a weird score with the project
- No Fixed TA hour

Outline

- Review of Labs
- Virtual Router Explained
- Virtual Router Specification
- ONOS App and Services In Use
- In Used App Configurations
- Virtual Router Workflow
- Supplements
- Scoring Criteria
- Reference

Reference

- ONOS JAVA API 2.7.0