SDNFV FINAL PROJECT

SDN Network as Virtual Router

助教：蔣汶儒

sdnta@win.cs.nycu.edu.tw

Deadline: 2025/12/18

OUTLINE

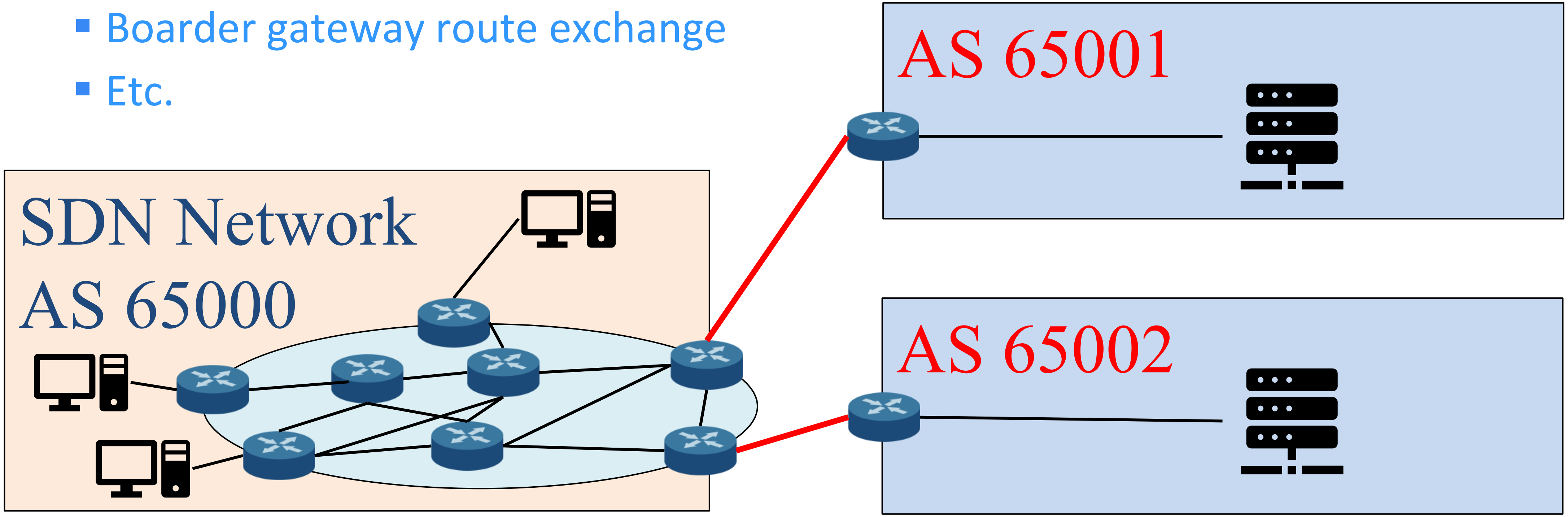
- Router Comparison
- Service in Use
- Router's activity
- Project Setup
- Final Project Topology
- Scoring Criteria and Verifications
- References

OUTLINE

- Router Comparison
- Service in Use
- Router’s activity
- Project Setup
- Final Project Topology
- Scoring Criteria and Verifications
- References

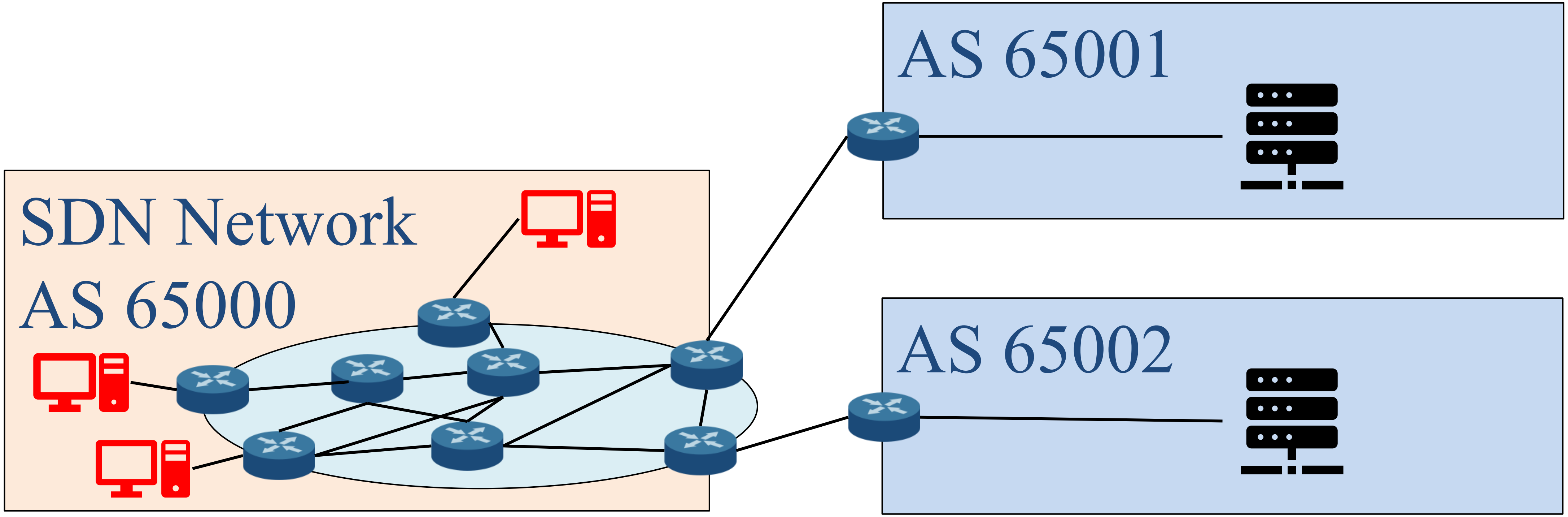
SDN-enabled Virtual Router

- SDN Network with virtual router
  - Use openflow switches and flowrules to simulate router behavior
  - For instance:
    - Layer2 forwarding for next hop communication
    - Boarder gateway route exchange
    - Etc.



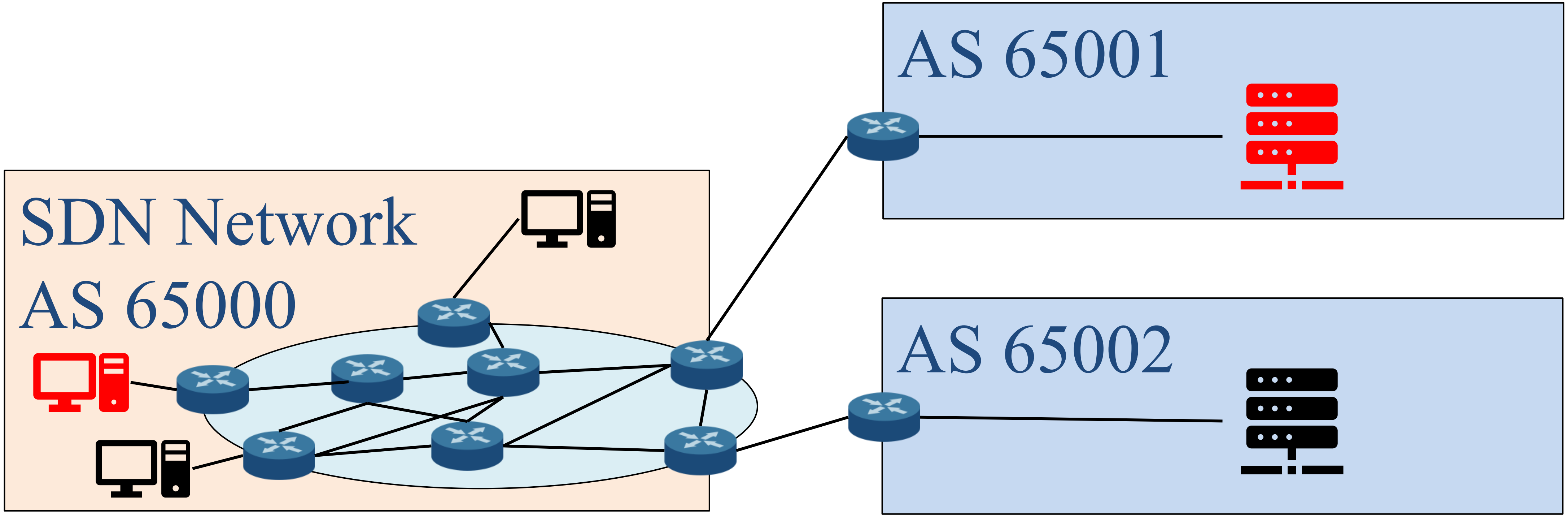
Traffic Types

- Intra-domain Traffics
  - Where **hosts** within the same AS communicates with each other.
  - SDN handles the traffic.



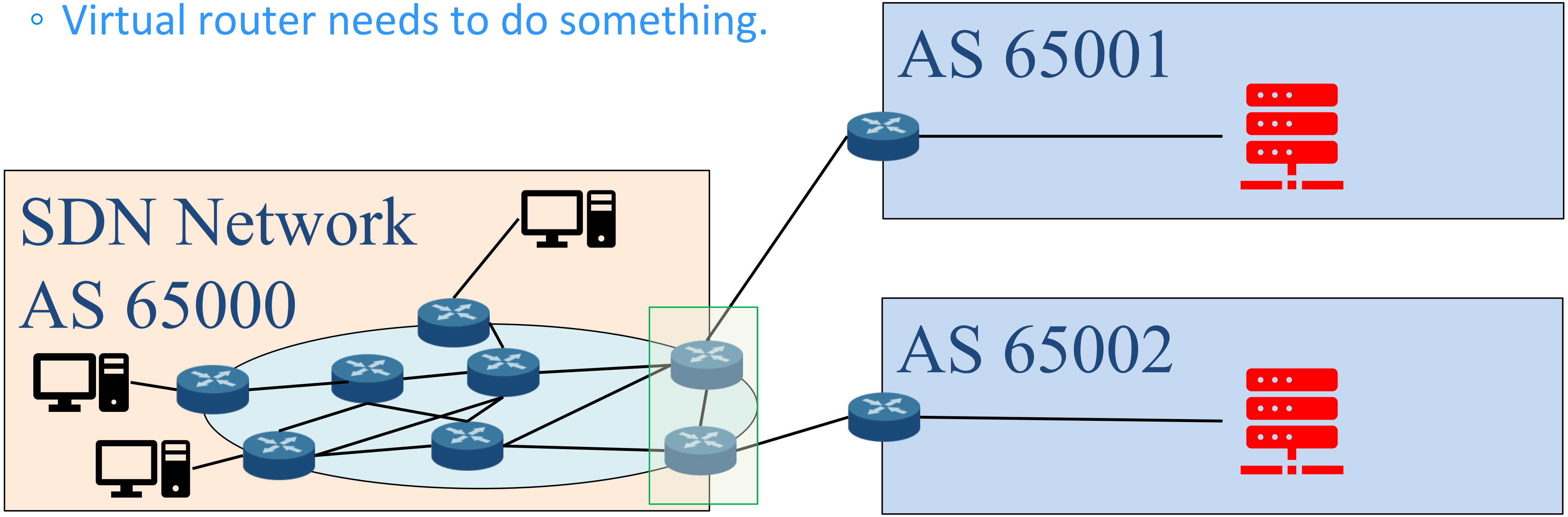
Traffic Types (cont.)

- Inter-domain Traffics
  - Where **an external host** from other domain communicates with **an internal host**.
  - The traffic pass through **gateways**.
  - Virtual router needs to do something.



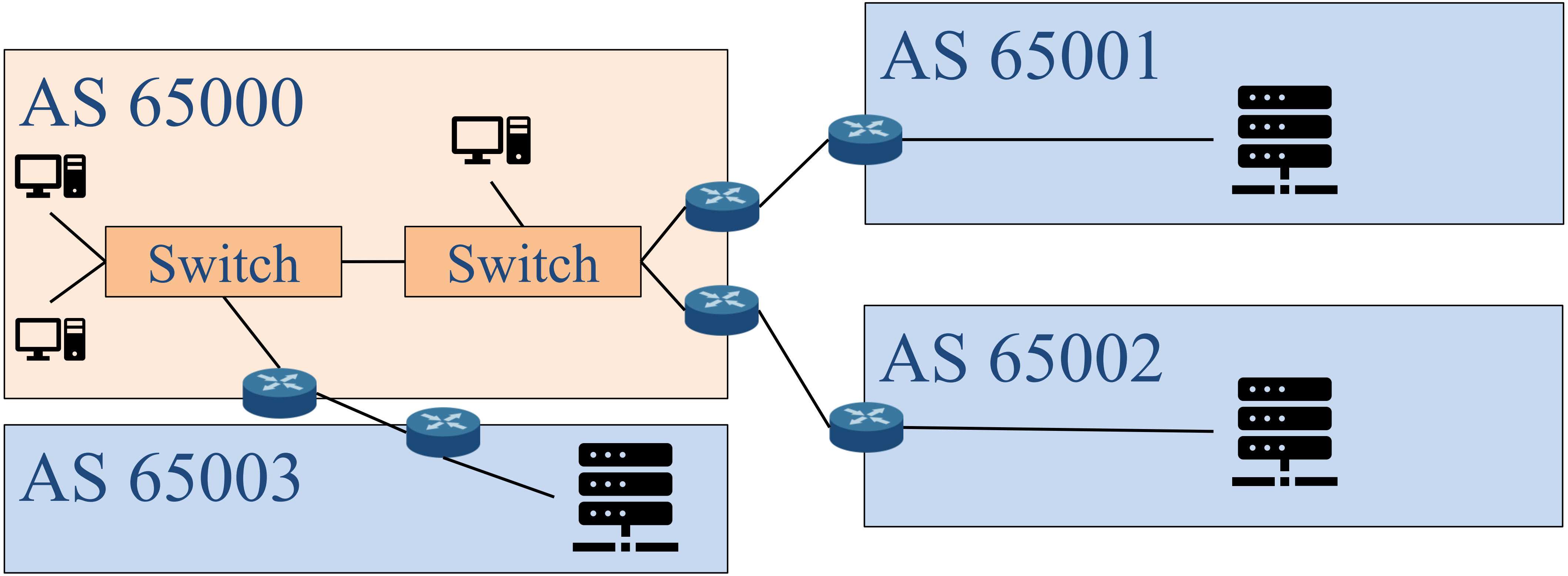
Traffic Types (cont.)

- Transit Traffics
  - Where **hosts** from different domains communicates with one another bypass the SDN network.
  - The traffic pass through virtual router.
  - Virtual router needs to do something.



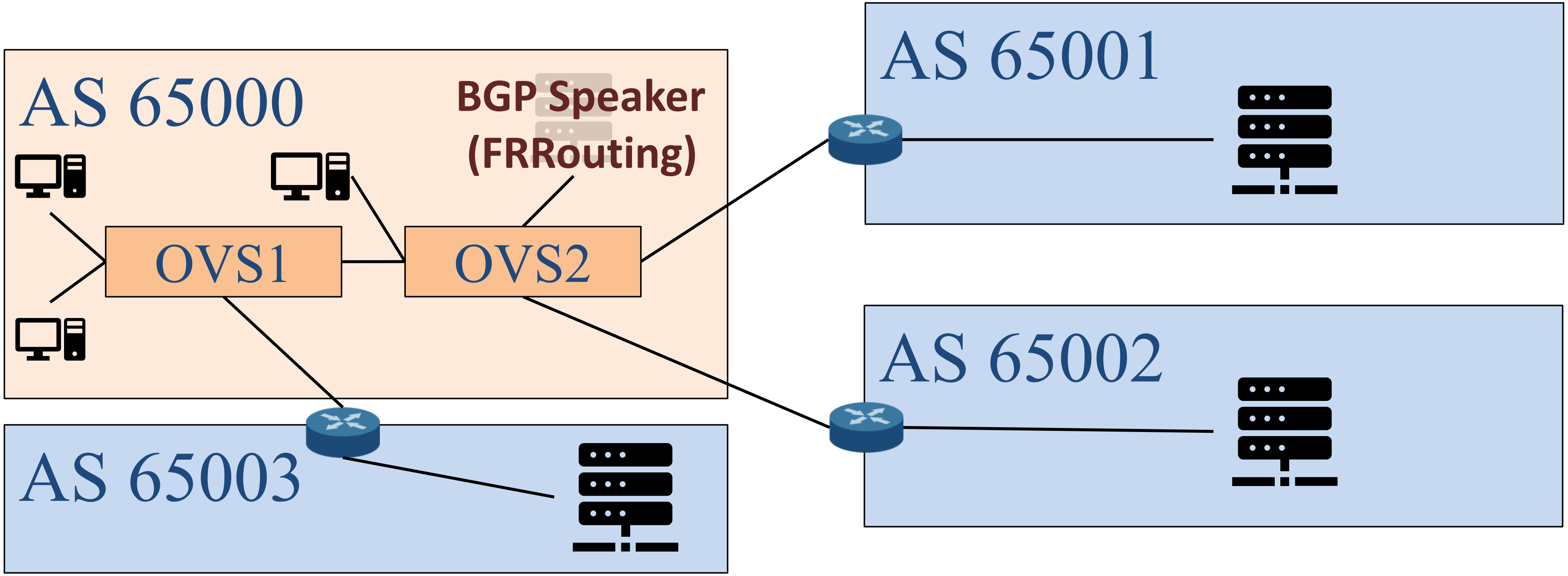
Networks with Physical Routers

- Physical routers
  - 1. Deal with routing decision.
  - 2. Deal with gateway exchange.
- Every edge requires a router, running eBGP and iBGP protocols.



SDN Networks with Virtual Routers

- SDN-enabled Virtual Routers
  - Doesn’t requires router connection to edge.
  - Only one BGP speaker is enough.
  - Doesn’t need a **real gateway.**

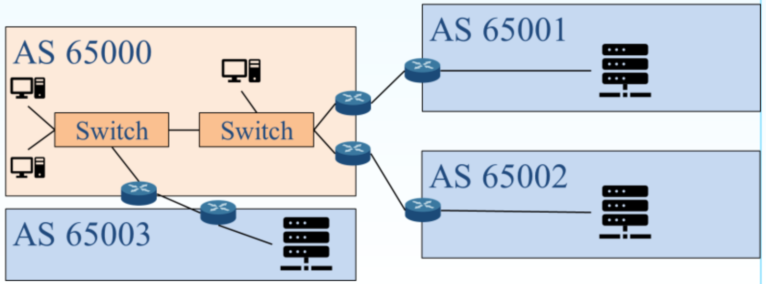


Goal

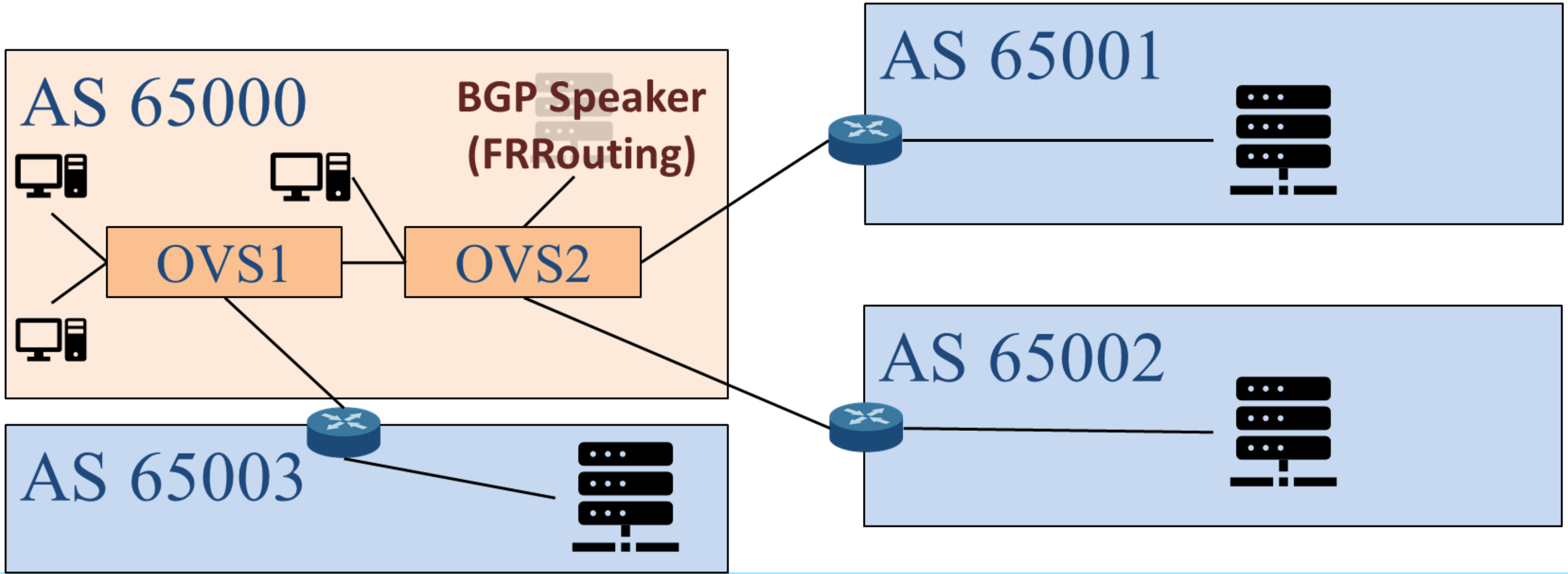
- Intra-domain host communication
  - Handled by Bridge APP
- Inter-domain host communication
  - SDN domain <-> Other domain
- Transit host communication
  - Other domain <-> SDN domain <-> Other domain

Virtual Router BGP Connection

- Physical router:
  - External routers connect with the boarder gateway.

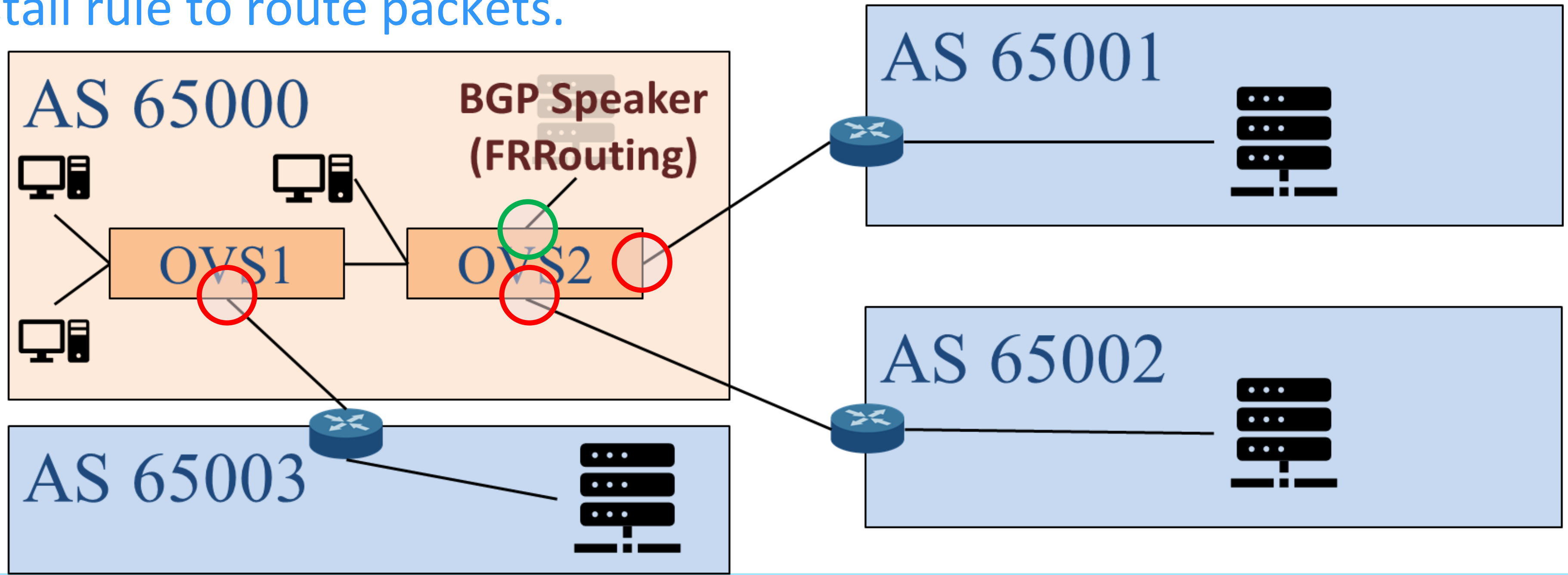


- Virtual router:
  - External routers connect with BGP Speaker.
  - ➢Need to delegate BGP Speaker IP to edge switch.



BGP Speaker IP Delegation and Routing

1. Delegate BGP speaker IP to the WAN Connect Point on edge switch.
   1) Determine WAN Connect Point.
   2) Config (via netcfg) WAN Connect Point interface.
2. Route packet between BGP speaker Connect Point and WAN Connect Point.
   1. Determine BGP speaker connect point.
   2. Install rule to route packets.

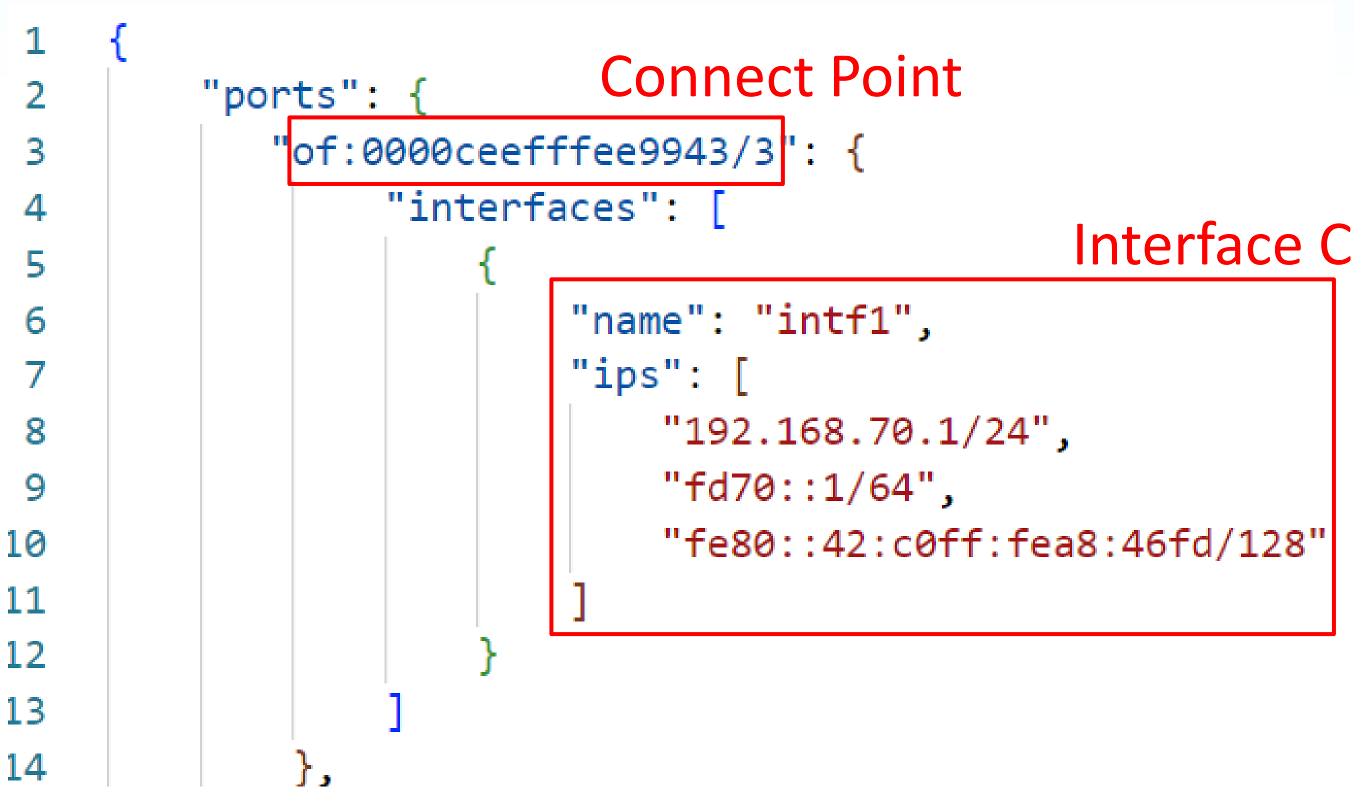


OUTLINE

- Router Comparison
- Service in Use
- Router’s activity
- Project Setup
- Final Project Topology
- Scoring Criteria and Verifications
- References

WAN Connect Point Configuration

- Create a configuration file for WAN Connect Point
    - Making external routers think that the BGP Speaker is at the Connect Point.



Connect Point

Interface Config

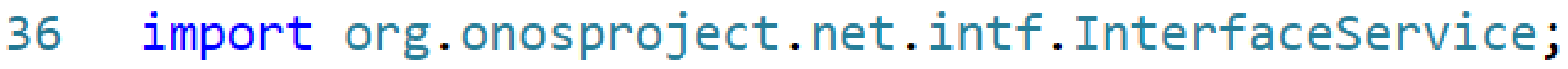
```
{
    "ports": {
        "of:0000ceefffee9943/3": {
            "interfaces": [
                {
                    "name": "intf1",
                    "ips": [
                        "192.168.70.1/24",
                        "fd70::1/64",
                        "fe80::42:c0ff:fea8:46fd/128"
                    ]
                }
            ]
        },
```

***NOTE**

This only gives ONOS controller information of the interface and its IPs. How to make BGP speaker receive packets designated to the IP is your work!

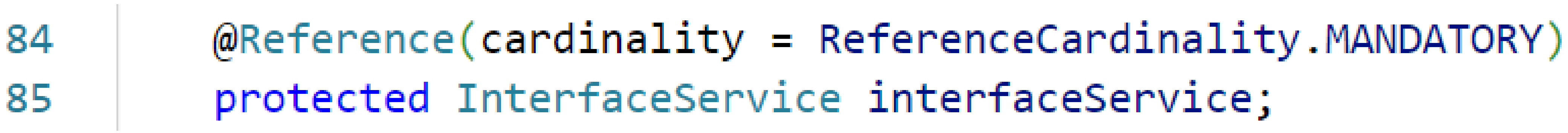
WAN Connect Point Information Retrieval

- Use ONOS Interface Service to retrieve WAN Connect Point.
  - Import Interface service



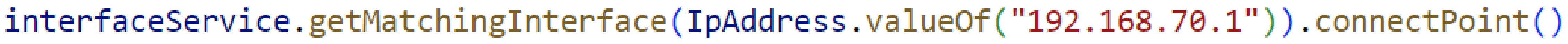
```
import org.onosproject.net.intf.InterfaceService;
```

  - Reference interface service



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

  - Query WAN Connect Point information from the interface service



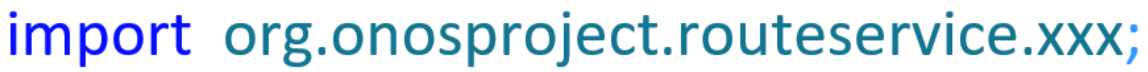
```
interfaceService.getMatchingInterface(IpAddress.valueOf("192.168.70.1")).connectPoint()
```

ONOS Route Service Usage

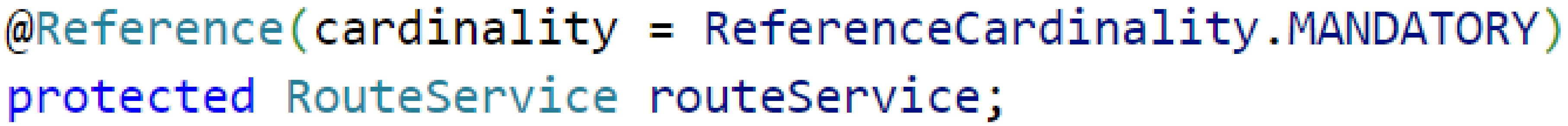
- Update dependencies in **pom.xml** file.

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

- Import methods.



```
import org.onosproject.routeservice.xxx;
```



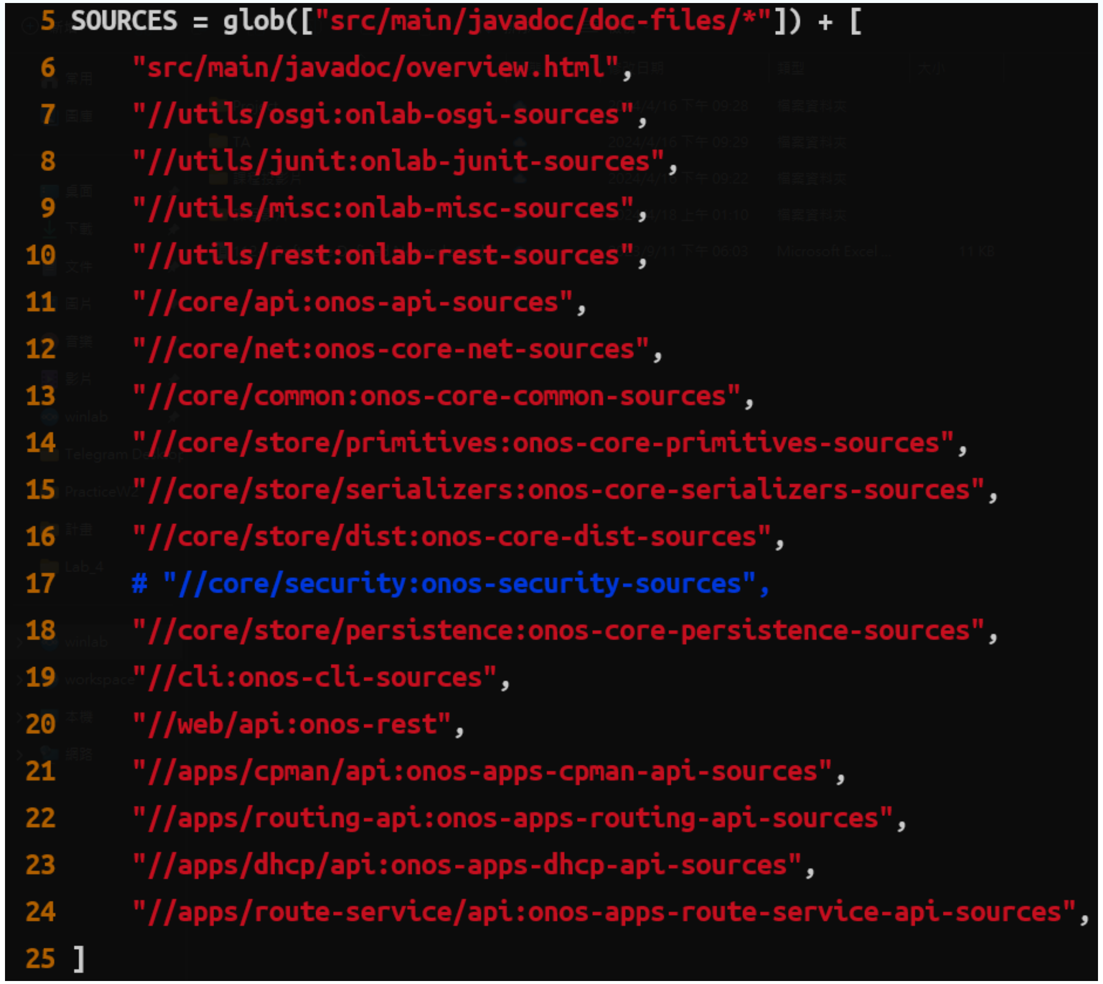
```
@Reference(cardinality = ReferenceCardinality.MANDATORY)
protected RouteService routeService;
```

- Read the docs

https://javadoc.io/doc/org.onosproject/onos-cli/1.8.1/org/onosproject/incubator/net/routing/package-summary.html

How to Read The Friendly Manual

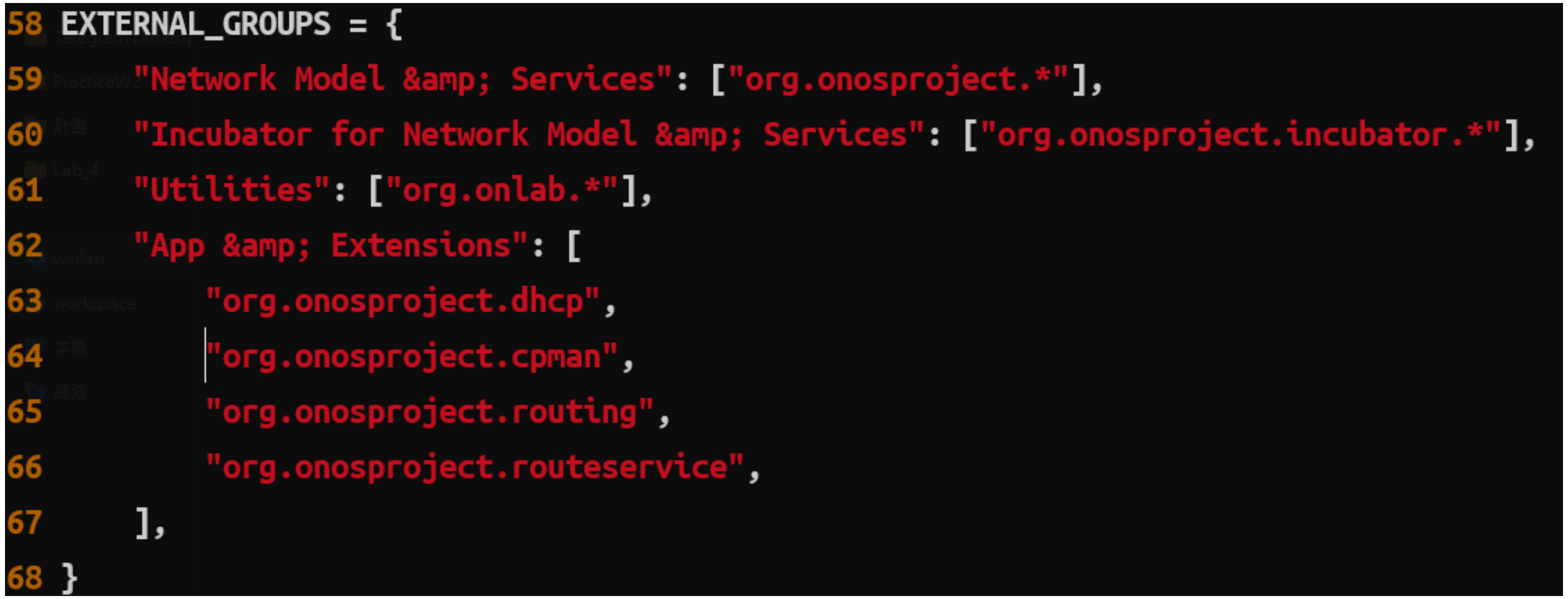
- Steps to build and view onos docs
  1. Goto ONOS_HOME (often ~/onos)
  2. Open docs/BUILD
  3. Modify SOURCES add `//apps/route-service/api:onos-apps-route-service-api-sources`



```
SOURCES = glob(["src/main/javadoc/doc-files/*"]) + [
    "src/main/javadoc/overview.html",
    "//utils/osgi:onlab-osgi-sources",
    "//utils/junit:onlab-junit-sources",
    "//utils/misc:onlab-misc-sources",
    "//utils/rest:onlab-rest-sources",
    "//core/api:onos-api-sources",
    "//core/net:onos-core-net-sources",
    "//core/common:onos-core-common-sources",
    "//core/store/primitives:onos-core-primitives-sources",
    "//core/store/serializers:onos-core-serializers-sources",
    "//core/store/dist:onos-core-dist-sources",
    # "//core/security:onos-security-sources",
    "//core/store/persistence:onos-core-persistence-sources",
    "//cli:onos-cli-sources",
    "//web/api:onos-rest",
    "//apps/cpman/api:onos-apps-cpman-api-sources",
    "//apps/routing-api:onos-apps-routing-api-sources",
    "//apps/dhcp/api:onos-apps-dhcp-api-sources",
    "//apps/route-service/api:onos-apps-route-service-api-sources",
]
```

How to Read The Friendly Manual

- Steps to build and view onos docs
  1. Goto ONOS_HOME (often ~/onos)
  2. Open docs/BUILD
  3. Modify SOURCES, add `//apps/route-service/api:onos-apps-route-service-api-sources`
  4. Modify EXTERNAL_GROUPS, add `org.onosproject.routeservice`



```
EXTERNAL_GROUPS = {
    "Network Model &amp; Services": ["org.onosproject.*"],
    "Incubator for Network Model &amp; Services": ["org.onosproject.incubator.*"],
    "Utilities": ["org.onlab.*"],
    "App &amp; Extensions": [
        "org.onosproject.dhcp",
        "org.onosproject.cpman",
        "org.onosproject.routing",
        "org.onosproject.routeservice",
    ],
}
```

How to Read The Friendly Manual

- Steps to build and view onos docs
  1. Goto ONOS_HOME (often ~/onos)
  2. Open docs/BUILD
  3. Modify SOURCES, add `//apps/route-service/api:onos-apps-route-service-api-sources`
  4. Modify EXTERNAL_GROUPS, add `org.onosproject.routeservice`
  5. Run `onos-build-docs`
  6. Run `unzip bazel-bin/docs/external.jar`
     - You shall find `apidocs` directory been generated
  7. Goto `apidocs` and run `python3 –m http.server`
  8. You will find the documentation been serve at localhost:8000

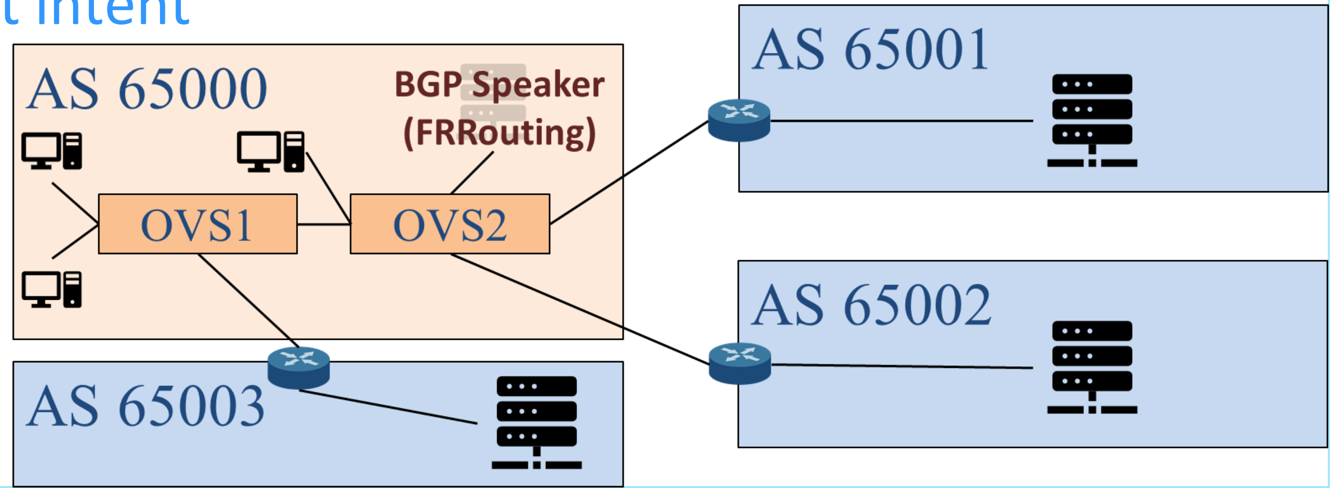
OUTLINE

- Router Comparison
- Service in Use
- Router’s activity
- Project Setup
- Final Project Topology
- Scoring Criteria and Verifications
- References

BGP Message Exchange

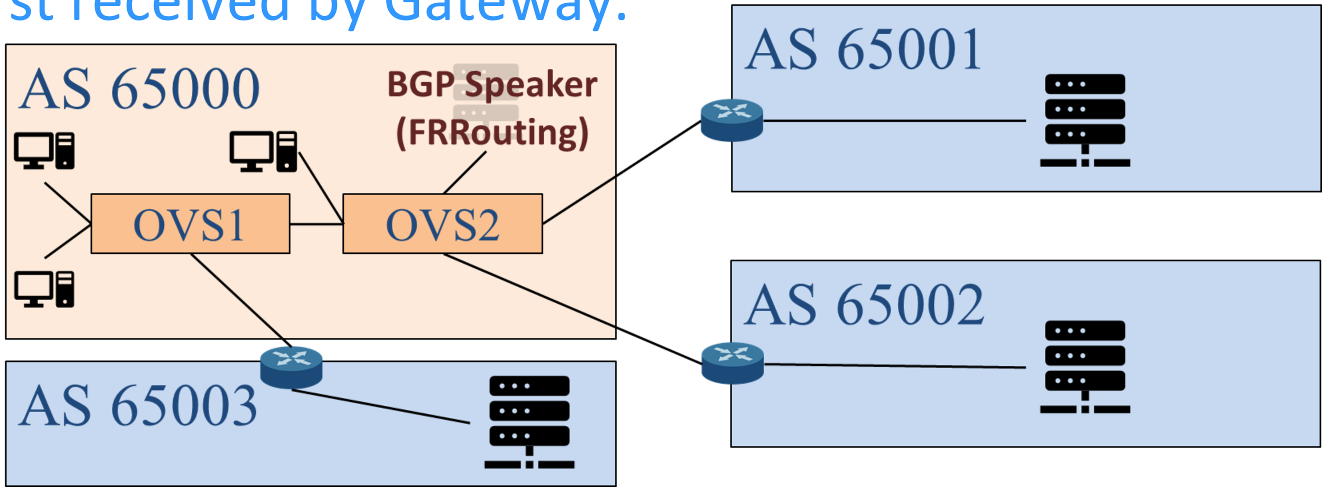
- In order to exchange BGP message with neighbor router
  - Neighbor discovery for L2 connectivity
    - Proxy ARP APP handles ARPs and NDPs on behalf of BGP Speaker.
  - L3 forwarding for BGP Messages

- **L3 forwarding for BGP Messages?**
  - **Incoming**
    - Hint: MultiPointToSinglePoint intent
  - **Outgoing**
    - ????



Virtual Gateway and Inter-domain Routing

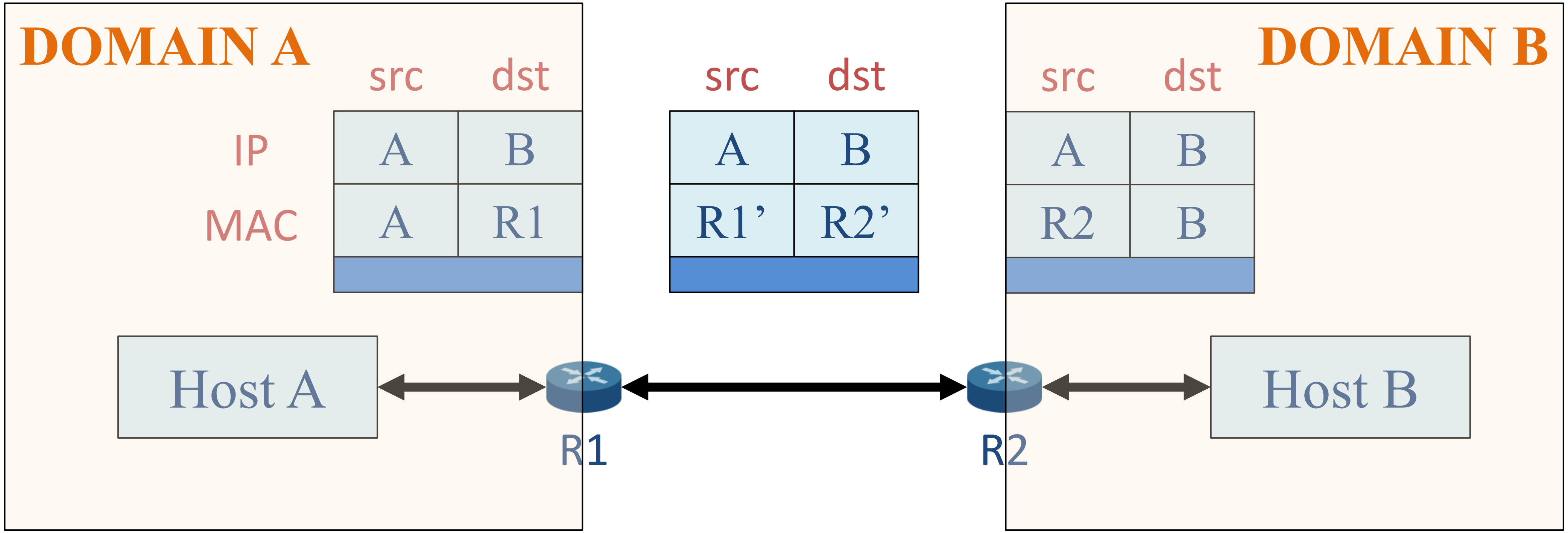
- Gateway and Routing
  - Assume Gateway IP: 192.168.1.254/24
    - Packets originated from 192.168.1.0/24 towards other networks
      - Packet first sent to Gateway.
    - Packet coming from other networks destined 192.168.1.0/24
      - Packet first received by Gateway.



- IP is the logical address of ultimate destination.
  - But, MAC is the physical address of the next hop.

Gateway Traffic Handling Example

- Any packets within the domain only knows about the gateway’s MAC.
- After analyzing the information (IP), it will change the according MAC and sends the packet out.



DOMAIN A

	src	dst
IP	A	B
MAC	A	R1

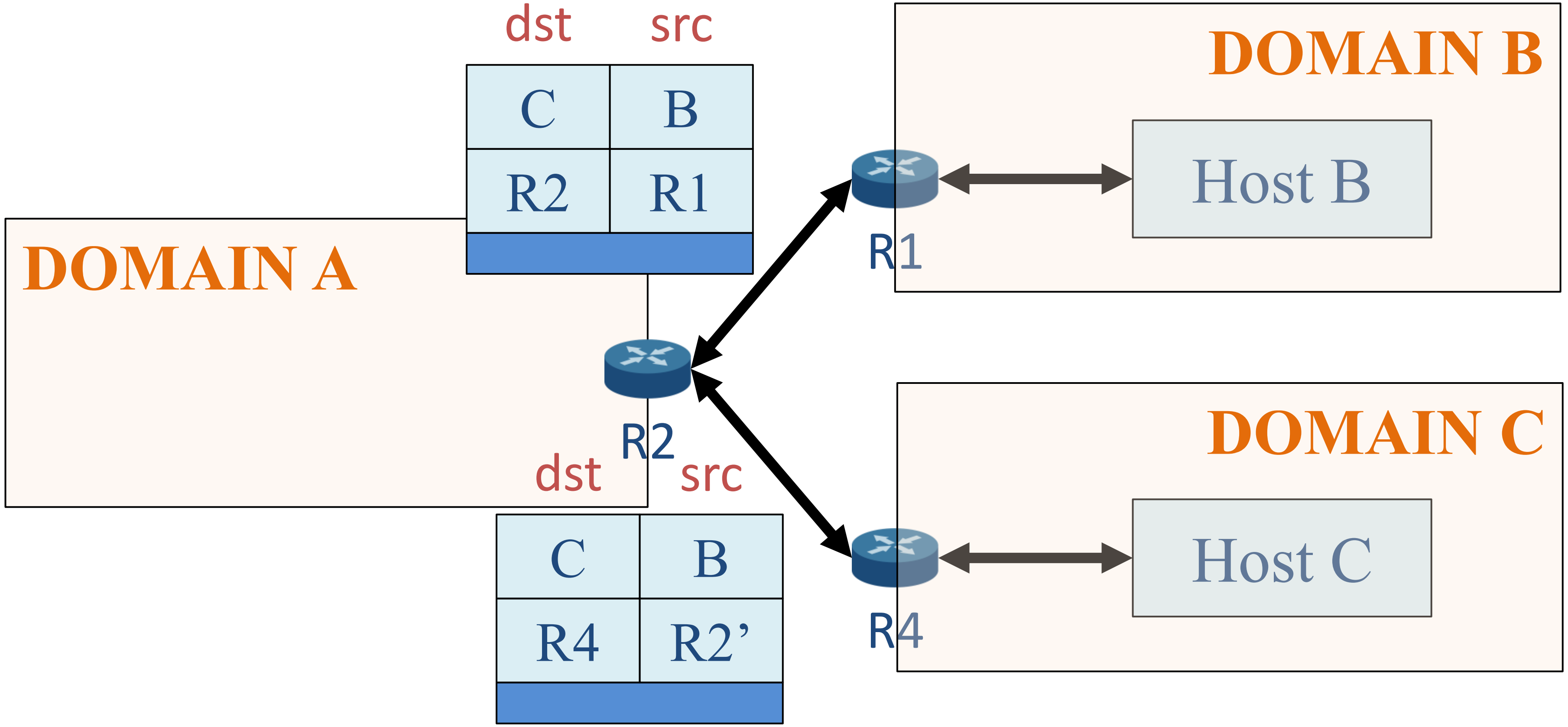
src	dst
A	B
R1’	R2’

DOMAIN B

src	dst
A	B
R2	B

Transit Traffics

- Transit traffics are in fact two interdomain traffics.



OUTLINE

- Router Comparison
- Service in Use
- Router’s activity
- Project Setup
- Final Project Topology
- Scoring Criteria and Verifications
- References

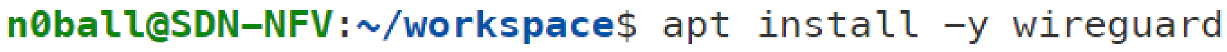
Tunnel and VXLAN

- A Tunnel send data over a network by encapsulating packets within other packets.
  - Commonly used to connect different networks, provide secure communication, bypass firewalls.
  - Applications
    - Virtual Private Networks (VPN)
    - Generic Routing Encapsulation (GRE)
    - Virtual Extensible LAN (VXLAN)
  - Types
    - Layer 2: Transmit data that is higher or equal to Layer 2
    - Layer 3: Transmit data that is higher or equal to Layer 3

- VXLAN (L2 Tunnel)
  - Designed to help build large, scalable L2 networks over existing L3 networks

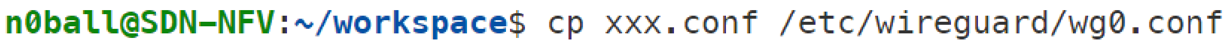
Wireguard

- TA will provide a Wireguard configuration file
  - **Remember, the last number of your IP is your ID (x)**
- Run the follow command to install wireguard



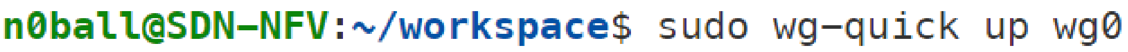
```
n0ball@SDN-NFV:~/workspace$ apt install -y wireguard
```

- Copy the configuration file to wg0



```
n0ball@SDN-NFV:~/workspace$ cp xxx.conf /etc/wireguard/wg0.conf
```

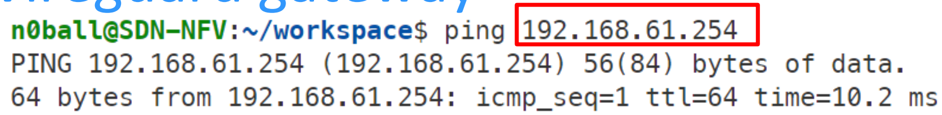
- Bring up the Wireguard interface



```
n0ball@SDN-NFV:~/workspace$ sudo wg-quick up wg0
```

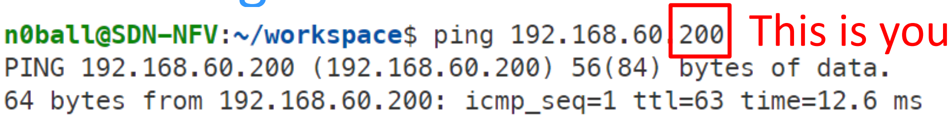
- Check if Wireguard is good
  - Wireguard gateway

This is Wireguard gateway



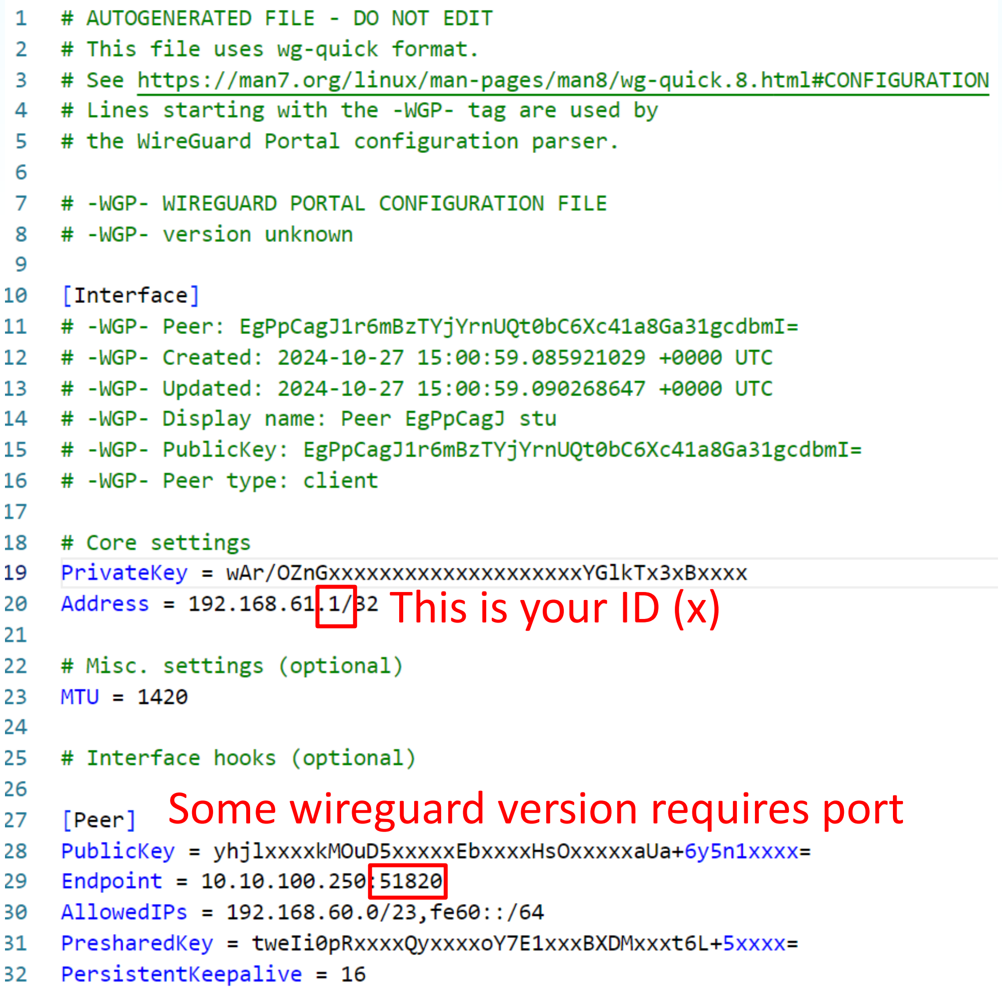
```
n0ball@SDN-NFV:~/workspace$ ping 192.168.61.254
PING 192.168.61.254 (192.168.61.254) 56(84) bytes of data.
64 bytes from 192.168.61.254: icmp_seq=1 ttl=64 time=10.2 ms
```

  - VXLAN Target



```
n0ball@SDN-NFV:~/workspace$ ping 192.168.60.200
PING 192.168.60.200 (192.168.60.200) 56(84) bytes of data.
64 bytes from 192.168.60.200: icmp_seq=1 ttl=63 time=12.6 ms
```

This is your x



```
# AUTOGENERATED FILE - DO NOT EDIT
# This file uses wg-quick format.
# See https://man7.org/linux/man-pages/man8/wg-quick.8.html#CONFIGURATION
# Lines starting with the -WGP- tag are used by
# the WireGuard Portal configuration parser.

# -WGP- WIREGUARD PORTAL CONFIGURATION FILE
# -WGP- version unknown

[Interface]
# -WGP- Peer: EgPpCagJ1r6mBzTYjYrnUQt0bC6Xc41a8Ga31gcdbmI=
# -WGP- Created: 2024-10-27 15:00:59.085921029 +0000 UTC
# -WGP- Updated: 2024-10-27 15:00:59.090268647 +0000 UTC
# -WGP- Display name: Peer EgPpCagJ stu
# -WGP- PublicKey: EgPpCagJ1r6mBzTYjYrnUQt0bC6Xc41a8Ga31gcdbmI=
# -WGP- Peer type: client

# Core settings
PrivateKey = wAr/OZnGxxxxxxxxxxxxxxxxxxxxYGlkTx3xBxxxx
Address = 192.168.61.1/32

# Misc. settings (optional)
MTU = 1420

# Interface hooks (optional)

[Peer]
PublicKey = yhjlxxxxkMOuD5xxxxxEbxxxxHsOxxxxxaUa+6y5n1xxxx=
Endpoint = 10.10.100.250:51820
AllowedIPs = 192.168.60.0/23,fe60::/64
PresharedKey = tweIi0pRxxxxQyxxxxoY7E1xxxBXDMxxxt6L+5xxxx=
PersistentKeepalive = 16
```

This is your ID (x)

Some wireguard version requires port

Maximum Transmission Unit (MTU)

- The largest size (in bytes) of a network packet that can be transmitted over a particular interface or network medium without fragmentation.
- Default to 1500 Bytes.

- What Happens if Packet Size Exceeds the MTU?
  - Fragmentation
  - Drop (Especially IPv6)

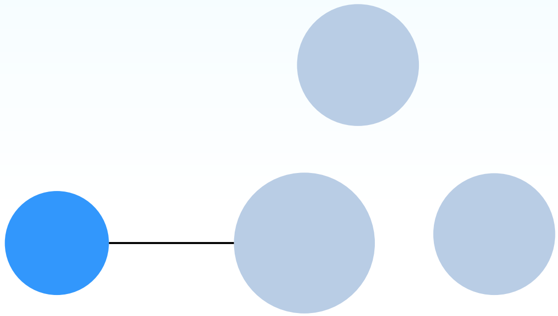
- Suggested subtraction of MTU due to encapsulation
  - VXLAN: 50 Bytes
  - Wireguard: 80 Bytes

OUTLINE

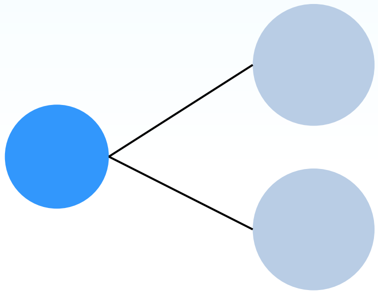
- Router Comparison
- Service in Use
- Router's activity
- Project Setup
- Final Project Topology
- Scoring Criteria and Verifications
- References

Autonomous System (AS)

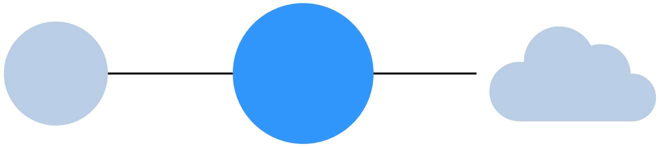
- AS Types



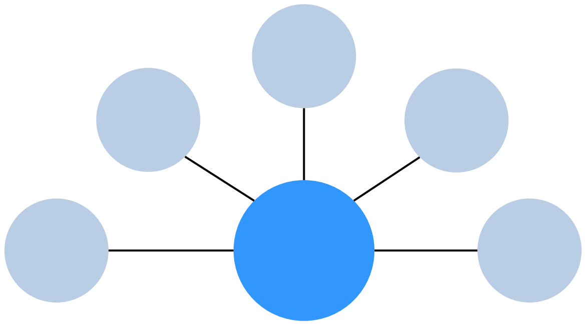
Stub



Multi Homed



Transit



Internet Exchange Point

ISP Characteristic

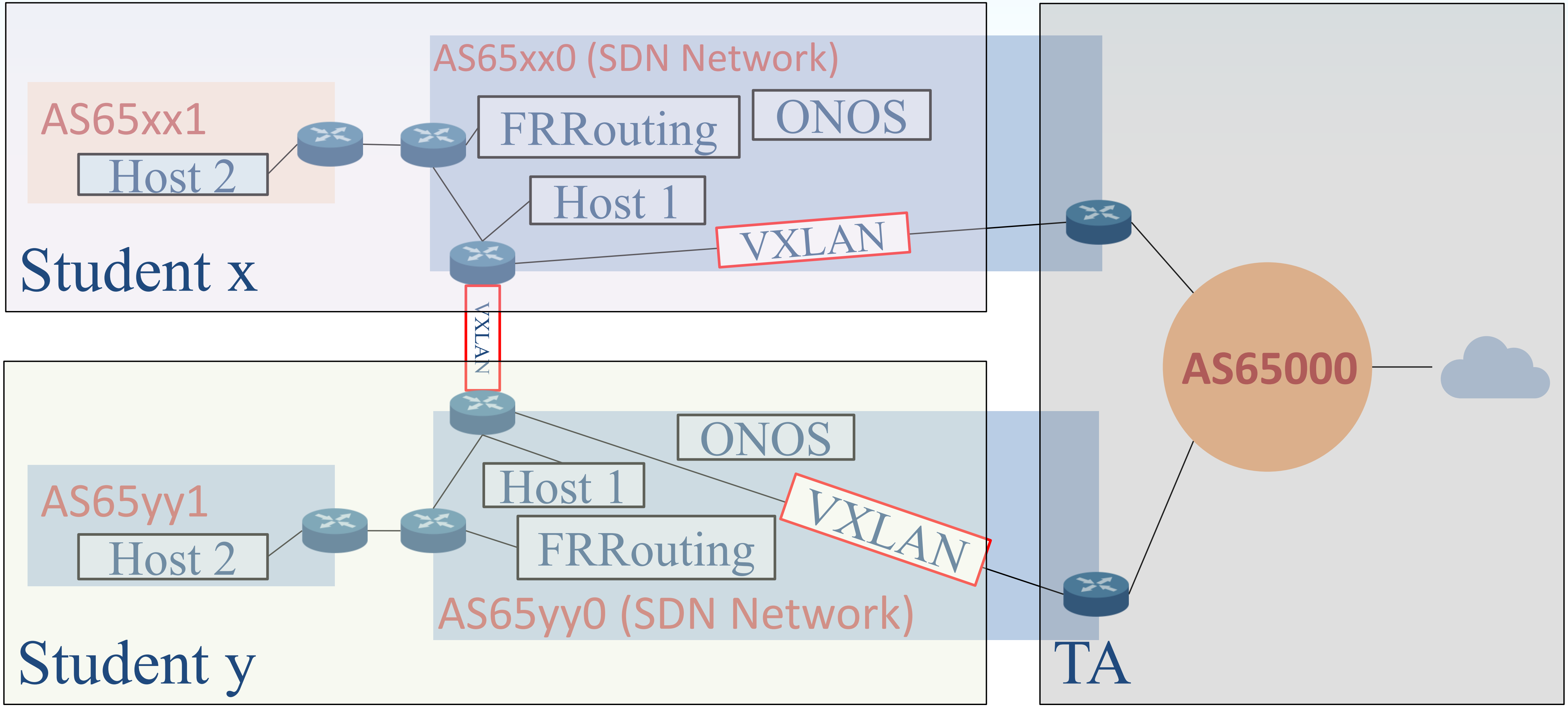
- Internet Service Provider (ISP)
  - A company or organization that provides individuals, enterprises, and other entities access to the internet.
- Key Concepts
  - Internet Access
  - Internet Service
  - IP Addresses
- If you have been assigned an IP from NYCU
  - How does the Internet world routes packets to your IP’s location?
  - How does others know the way to find you in the Internet world?
- Goal: A vrouter in a small ISP and that manages internet resources.

ISP Service Requirements

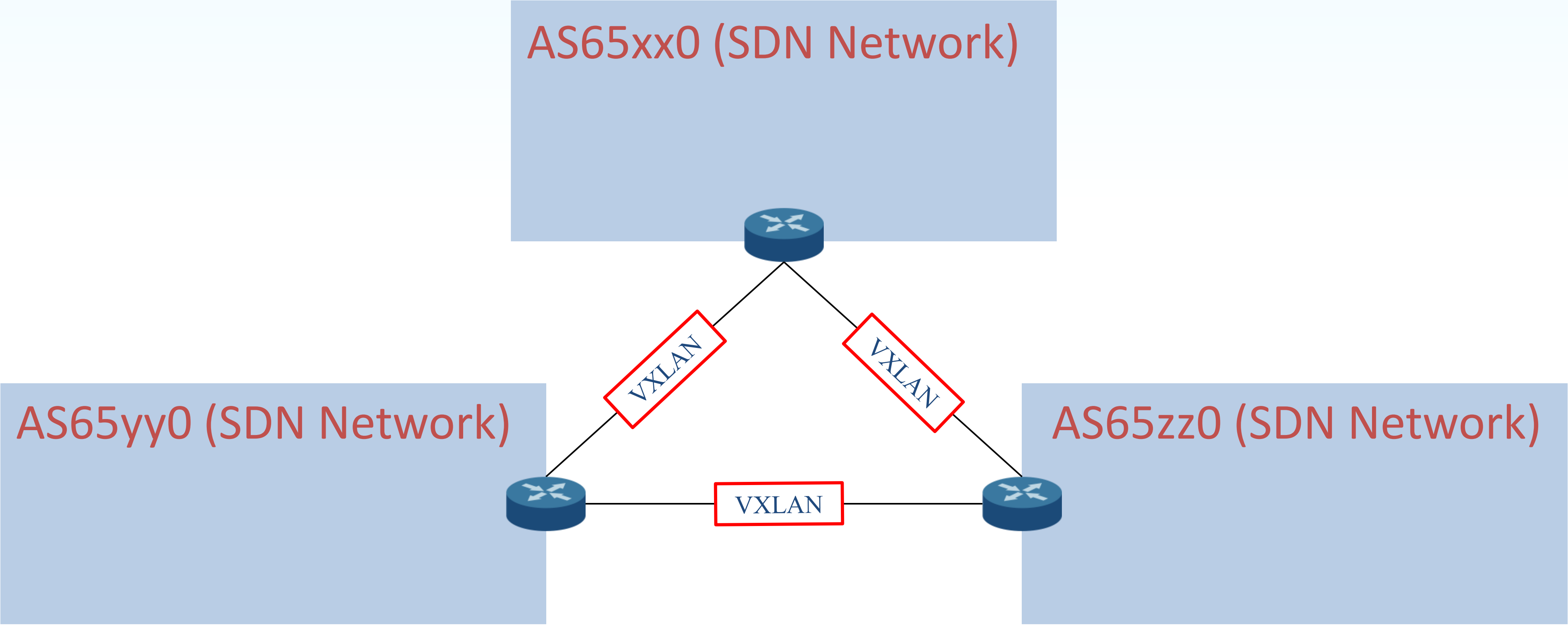
- Service requirements for customers.
  - Routers to exchange other AS's route.
  - Layer 2 modification
  - Packet handling

- Service requirements for other ISPs.
  - Packet handling
  - Quality of Service (QoS)

Topology


AS65xx0 (SDN Network)
AS65xx1
FRRouting
ONOS
Host 2
Host 1
VXLAN
Student x
VXLAN
AS65000
ONOS
Host 1
AS65yy1
Host 2
FRRouting
VXLAN
AS65yy0 (SDN Network)
Student y
TA

Topology for 2/3 Students



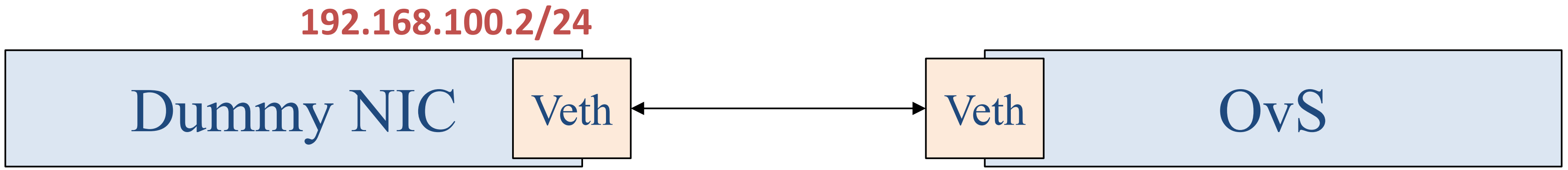
Configuration Requirements

- You are running an AS65xx0 and announcing prefixes
  - 172.16.x.0/24.
  - 2a0b:4e07:c4:xx::/64
- You help to transit prefixes announced by AS65xx1
  - 172.17.x.0/24.
  - 2a0b:4e07:c4:1xx::/64
- IXP is AS65000 at 192.168.70.253/24 and fd70::fe/64
  - You have to announce the prefixes you know to the IXP
  - You can connect the IXP via
    - 192.168.70.x/24
    - fd70::x/64
  - BGP Password is winlab.nycu

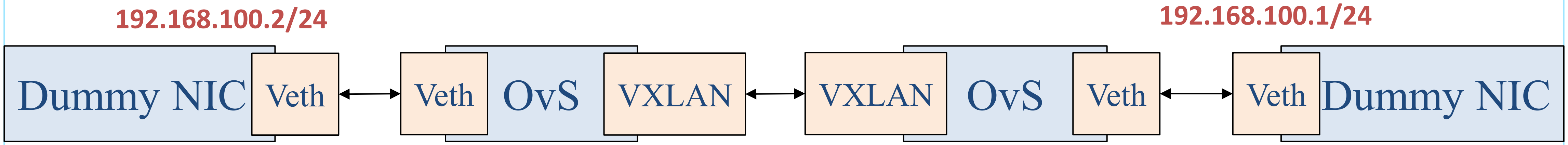
OVS and VXLAN

- You can consider OVS as a bridge.
  - You can create veth pairs and connect your container
  - You can create VXLAN Tunnel

- Let’s see if we can create a tunnel using `ovs-vsctl` and `ip` commands.



- TA Server have already opened a VXLAN connection with IP 192.168.100.2/24

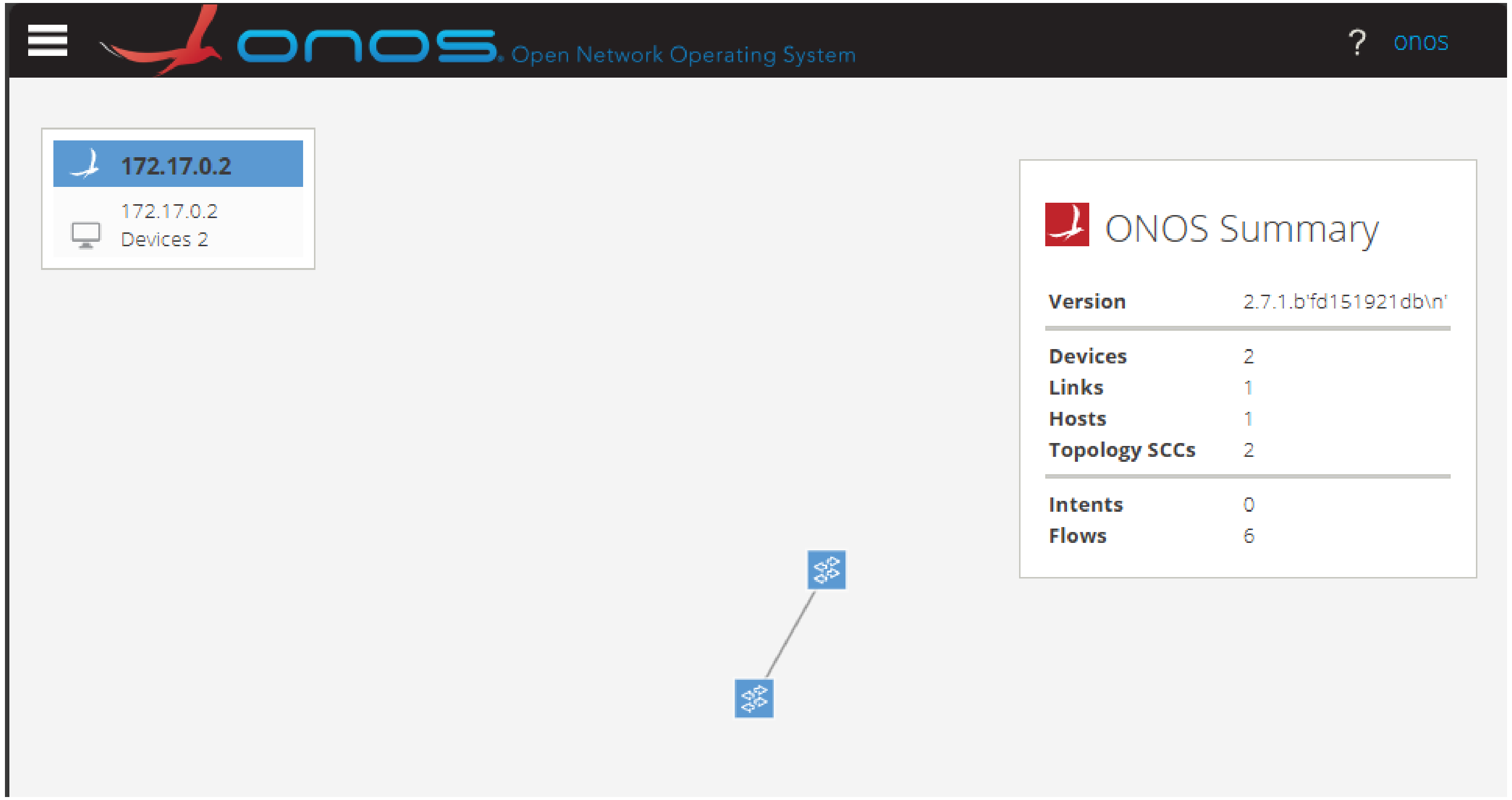


OVS and VXLAN (cont.)

- You can use docker to create an onos controller

```
n0ball@SDN-NFV:~/workspace$ docker run --rm --name onos -d -p 8181:8181 -p 6653:6653 -p 8101:8101 onosproject/onos:2.7-latest
```

- If you have finish lest step, you shall see two switches connected



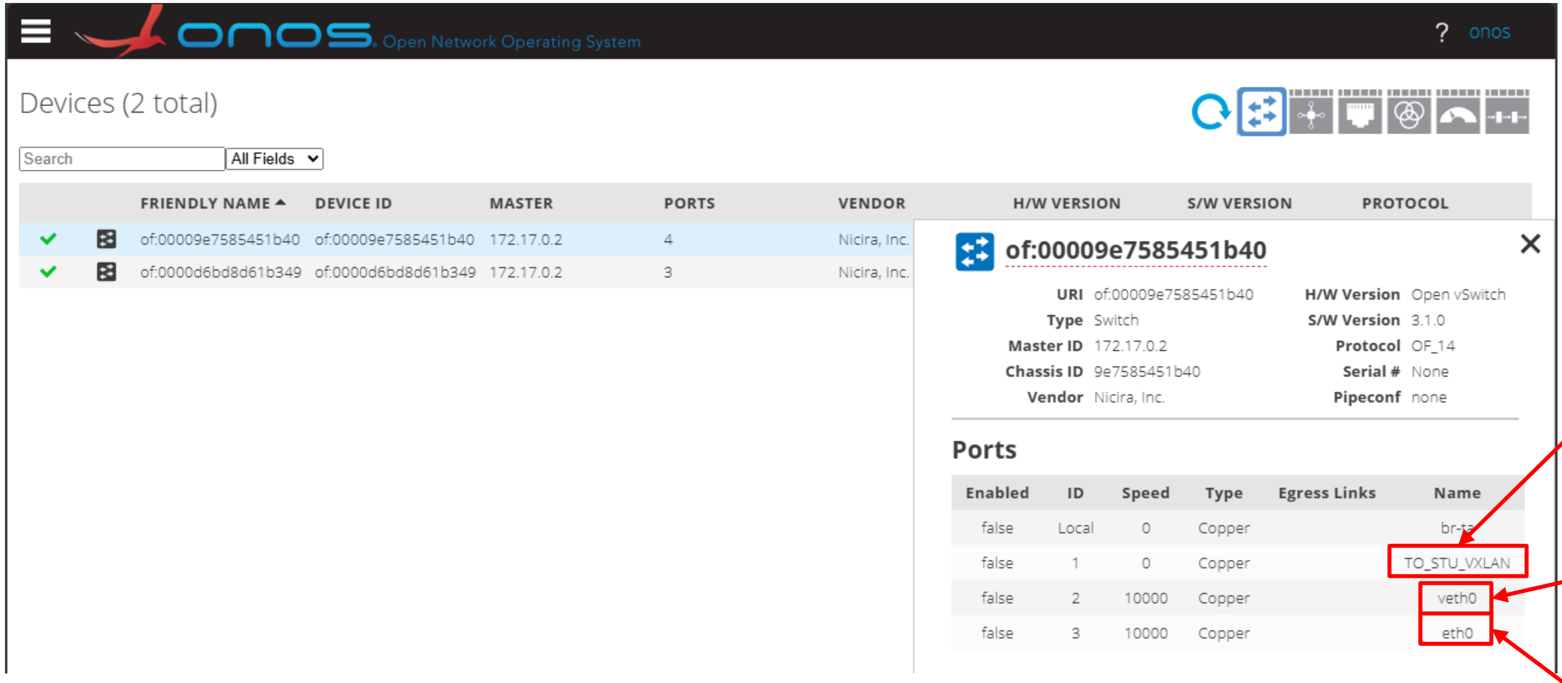
172.17.0.2
Devices 2

ONOS Summary

Version	2.7.1.b'fd151921db\n'
Devices	2
Links	1
Hosts	1
Topology SCCs	2
Intents	0
Flows	6

OVS and VXLAN (cont.)

- You can see ports on the onos GUI that TA have created for you.



onos Open Network Operating System ? onos

Devices (2 total)

Search | All Fields

		FRIENDLY NAME ▲	DEVICE ID	MASTER	PORTS	VENDOR	H/W VERSION	S/W VERSION	PROTOCOL
✓		of:00009e7585451b40	of:00009e7585451b40	172.17.0.2	4	Nicira, Inc.			
✓		of:0000d6bd8d61b349	of:0000d6bd8d61b349	172.17.0.2	3	Nicira, Inc.			

of:00009e7585451b40

URI	of:00009e7585451b40	H/W Version	Open vSwitch
Type	Switch	S/W Version	3.1.0
Master ID	172.17.0.2	Protocol	OF_14
Chassis ID	9e7585451b40	Serial #	None
Vendor	Nicira, Inc.	Pipeconf	none

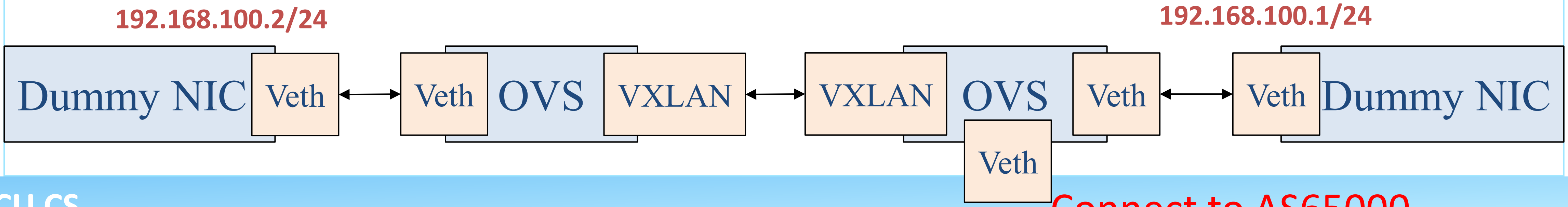
Ports

Enabled	ID	Speed	Type	Egress Links	Name
false	Local	0	Copper		br-ta
false	1	0	Copper		TO_STU_VXLAN
false	2	10000	Copper		veth0
false	3	10000	Copper		eth0

VLAN Port

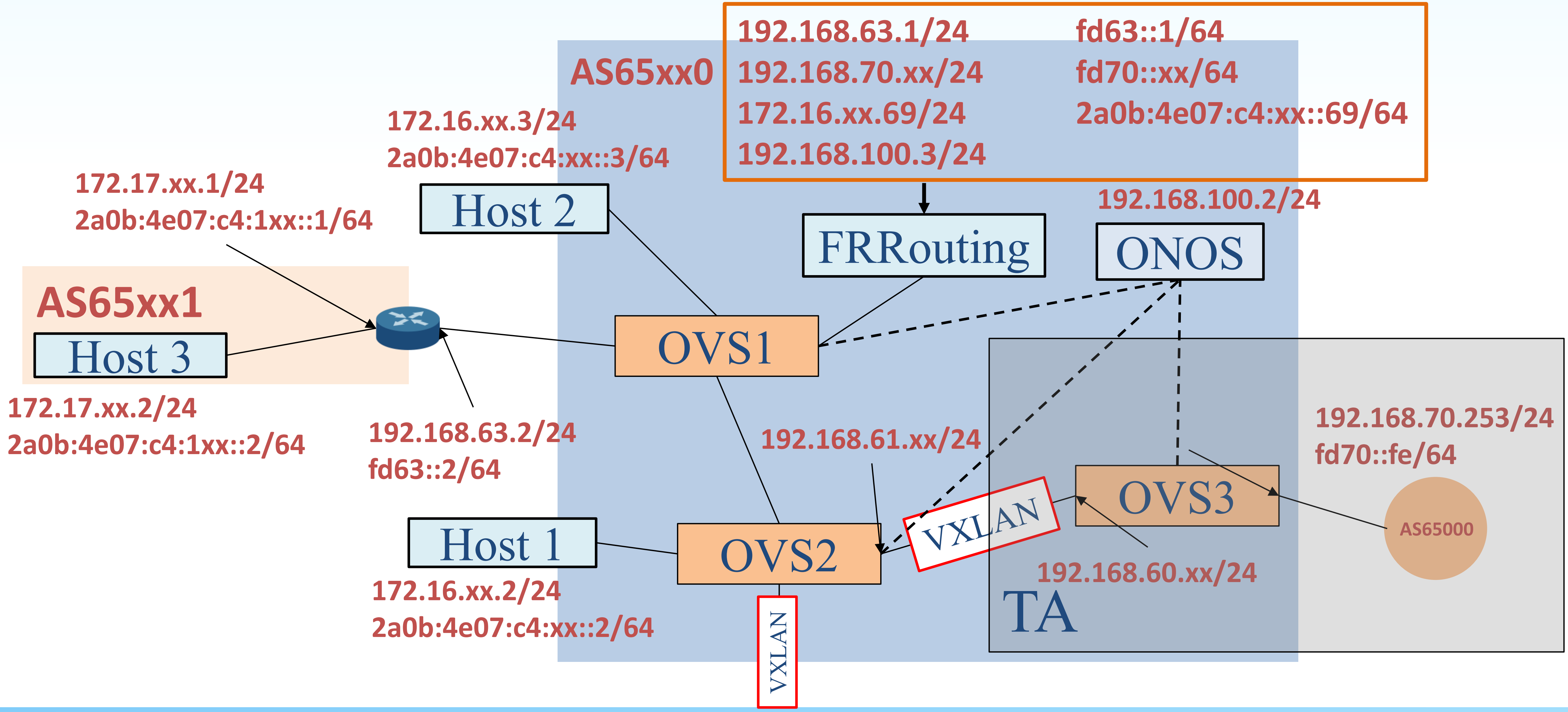
A Veth for dummy NIC

Connect to AS65000

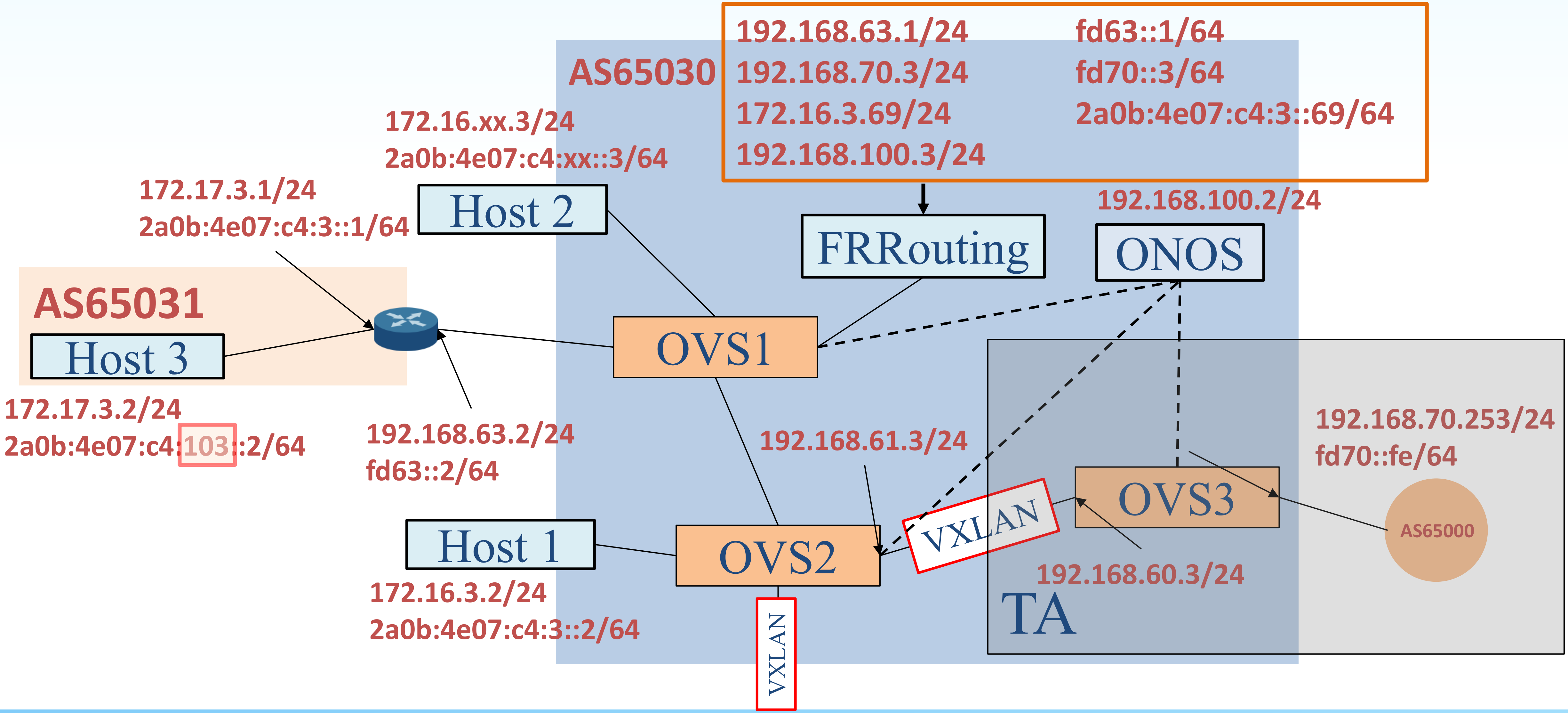


Connect to AS65000

Topology Template

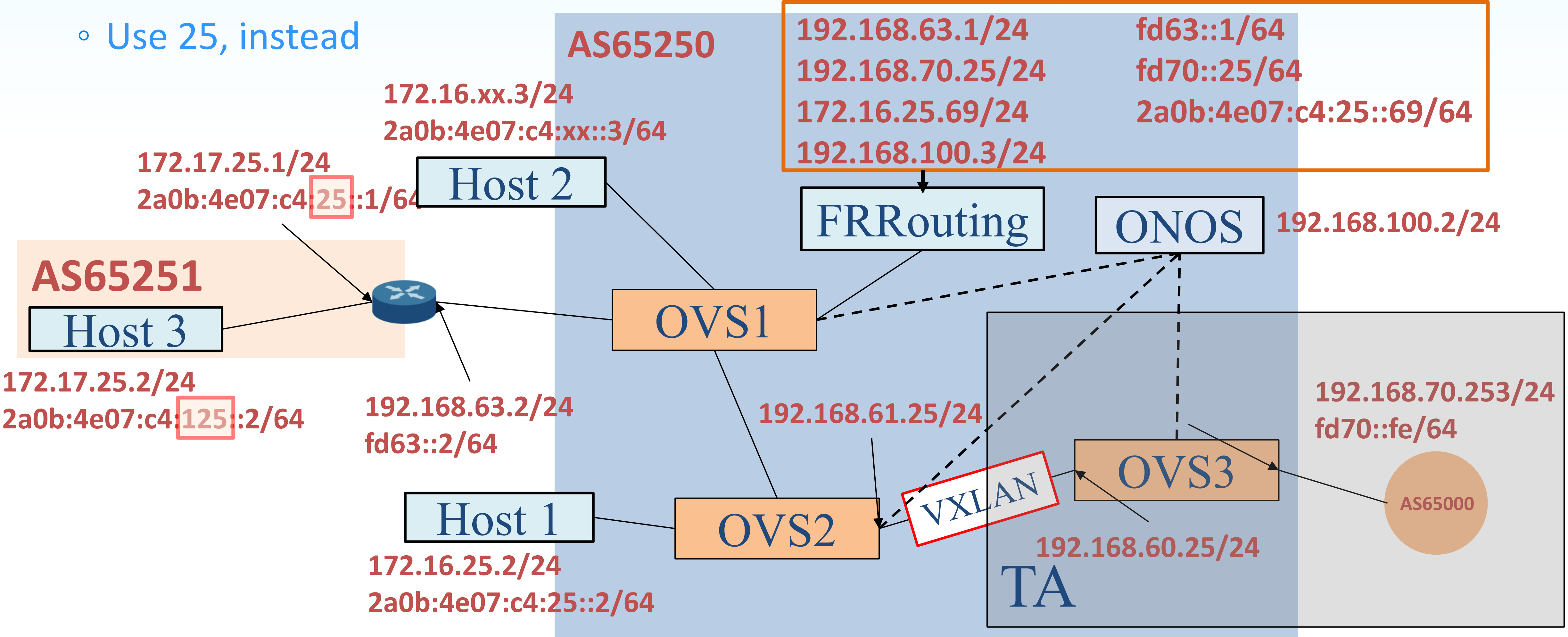


Sample Topology (x=3)



Sample Topology (x=25)

- To avoid misconfiguration, don’t use a-f as IP in your settings.
  - Use 25, instead



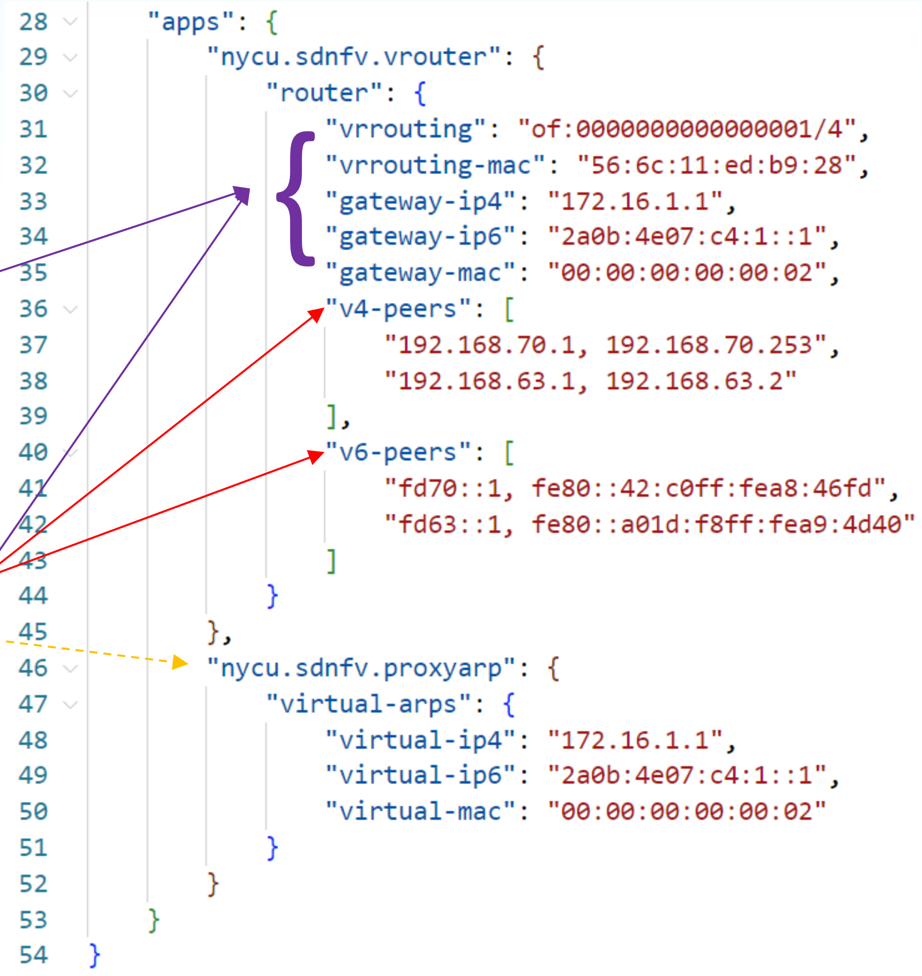
vRouter TODO List

- Create a VXLAN tunnel between your SDN network and TA’s server.
- ONOS can control OVS.
- Adding IPV6 capability.
- Intra/inter-domain packet handling.
- Virtual Gateway function.
- Edge routers can ping each other (FRRouting is not directly connect to external router).
- FRRouting can push FIB to ONOS controller.
  - (Connection between 192.168.100.1 and 192.168.100.3)
- Deal with inter-domain traffics, i.e. ASx’s hosts can ping ASy’s hosts.
- Deal with transit traffics, i.e. ASx’s hosts can ping from outside the world.

APP Configs

- Recommended configs

- Create a VXLAN tunnel between your SDN network and TA’s server.
- ONOS can control OVS.
- Adding IPV6 capability.
- Intra/inter-domain packet handling.
- Virtual Gateway function.
- Edge routers can ping each other (FRRouting is not directly connect to external router).
- FRRouting can push FIB to ONOS controller.
- Deal with inter-domain traffics, i.e. ASx’s hosts can ping ASy’s hosts.
- Deal with transit traffics, i.e. ASx’s hosts can ping from outside the world.



```
    "apps": {
        "nycu.sdnfv.vrouter": {
            "router": {
                "vrrouting": "of:0000000000000001/4",
                "vrrouting-mac": "56:6c:11:ed:b9:28",
                "gateway-ip4": "172.16.1.1",
                "gateway-ip6": "2a0b:4e07:c4:1::1",
                "gateway-mac": "00:00:00:00:00:02",
                "v4-peers": [
                    "192.168.70.1, 192.168.70.253",
                    "192.168.63.1, 192.168.63.2"
                ],
                "v6-peers": [
                    "fd70::1, fe80::42:c0ff:fea8:46fd",
                    "fd63::1, fe80::a01d:f8ff:fea9:4d40"
                ]
            }
        },
        "nycu.sdnfv.proxyarp": {
            "virtual-arps": {
                "virtual-ip4": "172.16.1.1",
                "virtual-ip6": "2a0b:4e07:c4:1::1",
                "virtual-mac": "00:00:00:00:00:02"
            }
        }
    }
}
```

OUTLINE

- Router Comparison
- Service in Use
- Router’s activity
- Project Setup
- Final Project Topology
- Scoring Criteria and Verifications
- References

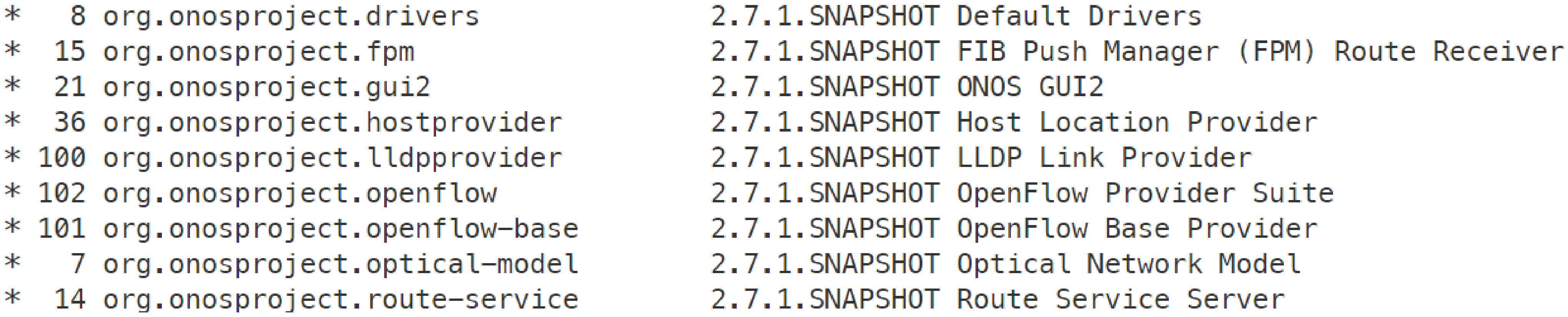
SDN AS Requirements

- For service customers (AS65xx0 SDN Network, **host 1**)
  - Able to ping **FRRouting's** IP (172.16.xx.69/24).
  - Able to ping **student y's FRRouting's** IP (172.16.yy.69/24).
  - Use https://tools.keycdn.com/ipv6-ping to see ICMP replies.

- For transit ISP (AS65xx1, **host 2**)
  - Able to ping **host 1** IP (172.16.xx.2/24).
  - Able to ping **student y's FRRouting's** IP (172.16.yy.69/24).
  - Able to ping **student y's host 2's** IP (172.17.yy.2/24).
  - Use https://tools.keycdn.com/ipv6-ping to see ICMP replies.

Deployment Requirements

- **DO NOT TRY TO ATTACK EITHER OTHER STUDENTS OR TA SERVERS**
- You are required to create a Makefile so that you are able to
  - Clear the entire project via **make clean** command
  - Deploy the project via **make deploy** command
- **Only openflow (and route service) related apps can be use**



```
*   8 org.onosproject.drivers               2.7.1.SNAPSHOT Default Drivers
*  15 org.onosproject.fpm                   2.7.1.SNAPSHOT FIB Push Manager (FPM) Route Receiver
*  21 org.onosproject.gui2                  2.7.1.SNAPSHOT ONOS GUI2
*  36 org.onosproject.hostprovider          2.7.1.SNAPSHOT Host Location Provider
* 100 org.onosproject.lldpprovider          2.7.1.SNAPSHOT LLDP Link Provider
* 102 org.onosproject.openflow              2.7.1.SNAPSHOT OpenFlow Provider Suite
* 101 org.onosproject.openflow-base         2.7.1.SNAPSHOT OpenFlow Base Provider
*   7 org.onosproject.optical-model         2.7.1.SNAPSHOT Optical Network Model
*  14 org.onosproject.route-service         2.7.1.SNAPSHOT Route Service Server
```

- **If not, you will be scored 0**

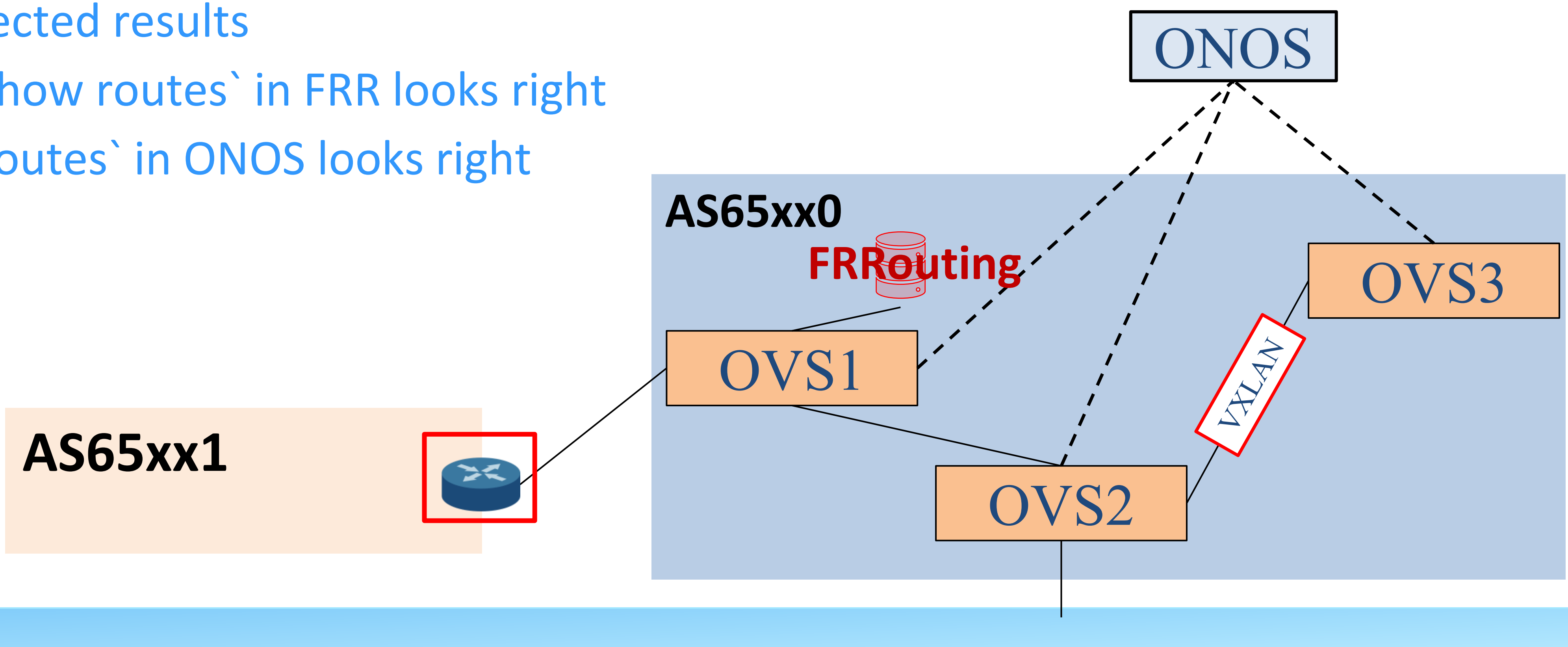
Scores

- Intra-domain traffic (from both AS)
  - IPv4 **(6 points) +** IPv6 **(4 points)**
- Inter-domain traffic (from both AS)
  - IPv4 **(12 points) +** IPv6 **(8 points)**
- Transit traffic
  - IPv4 **(12 points) +** IPv6 **(8 points)**
- Routes shown in IXP Manager
  - IPv4 **(2 points) +** IPv6 **(3 points)**

- Peer Traffic (**10 points** explanation + **10 points** verification)
- Anycast Traffic (**10 points** explanation + **10 points** verification)
- Able to communicate with the outside world **(5 points)**

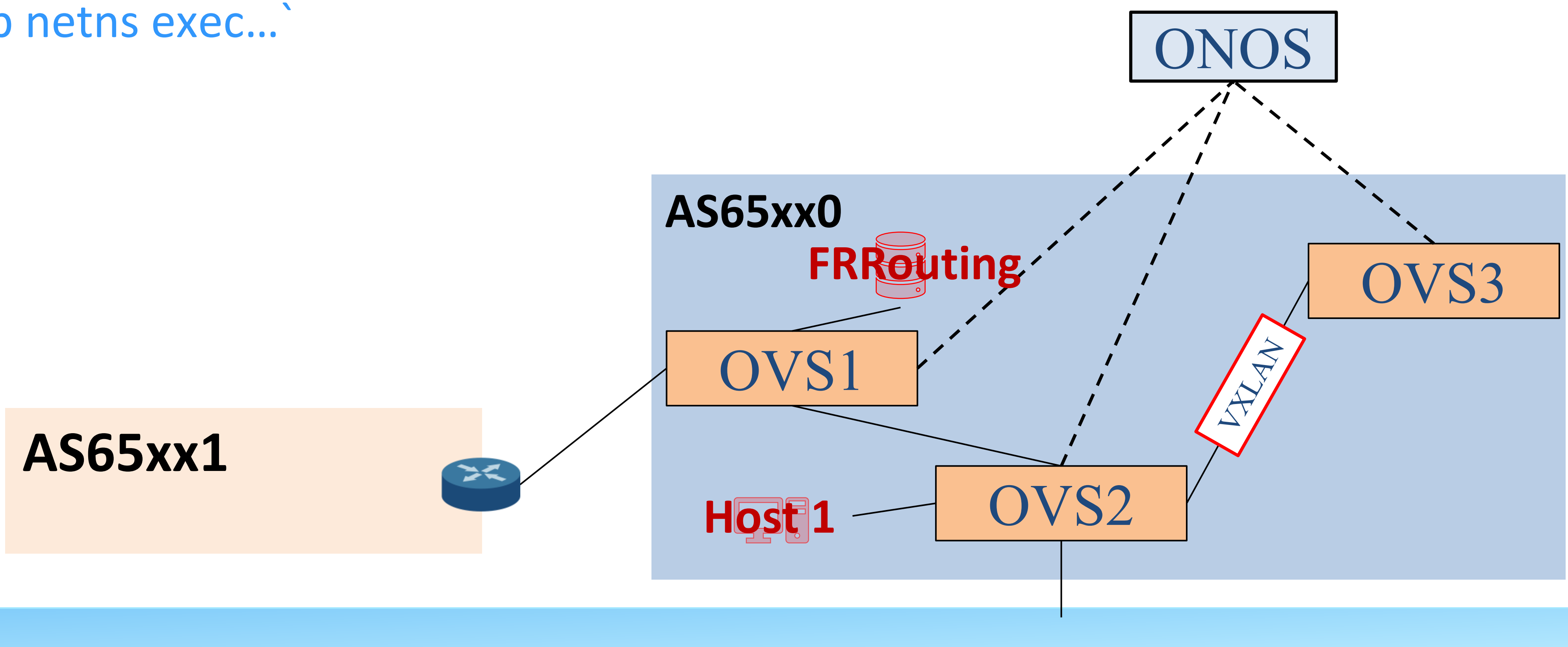
vRouter Verifications – Router Communication

- Assure Router Communication
  - Each FRRouting can ping its' neighbors
  - Things to think twice
    - What is the communication IP
  - Expected results
    - `show routes` in FRR looks right
    - `routes` in ONOS looks right



vRouter Verifications – Router Communication

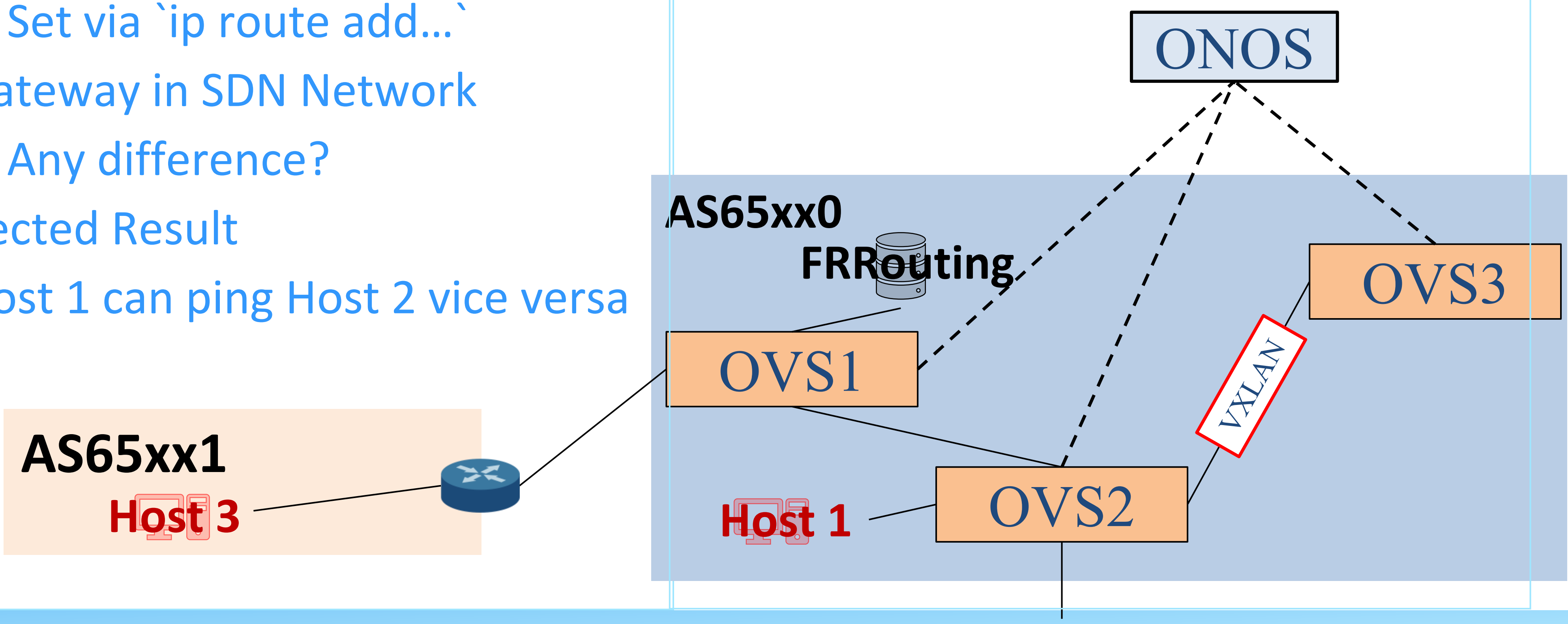
- Assure Intra Domain Traffic
  - BGP speaker in AS65xx0 can ping Web Server (host) vice versa
  - Host 1 can ping Host 2 vice versa
  - traefik/whoami does not have an interactive shell
    - \`ip netns exec…\`



vRouter Verifications – Inter Domain Traffic

- Assure Inter Domain Traffics
  - Get use of `routes` in OVS
  - Hint
    - Gateway in none-SDN network
      - Set via `ip route add…`
    - Gateway in SDN Network
      - Any difference?
  - Expected Result
    - Host 1 can ping Host 2 vice versa

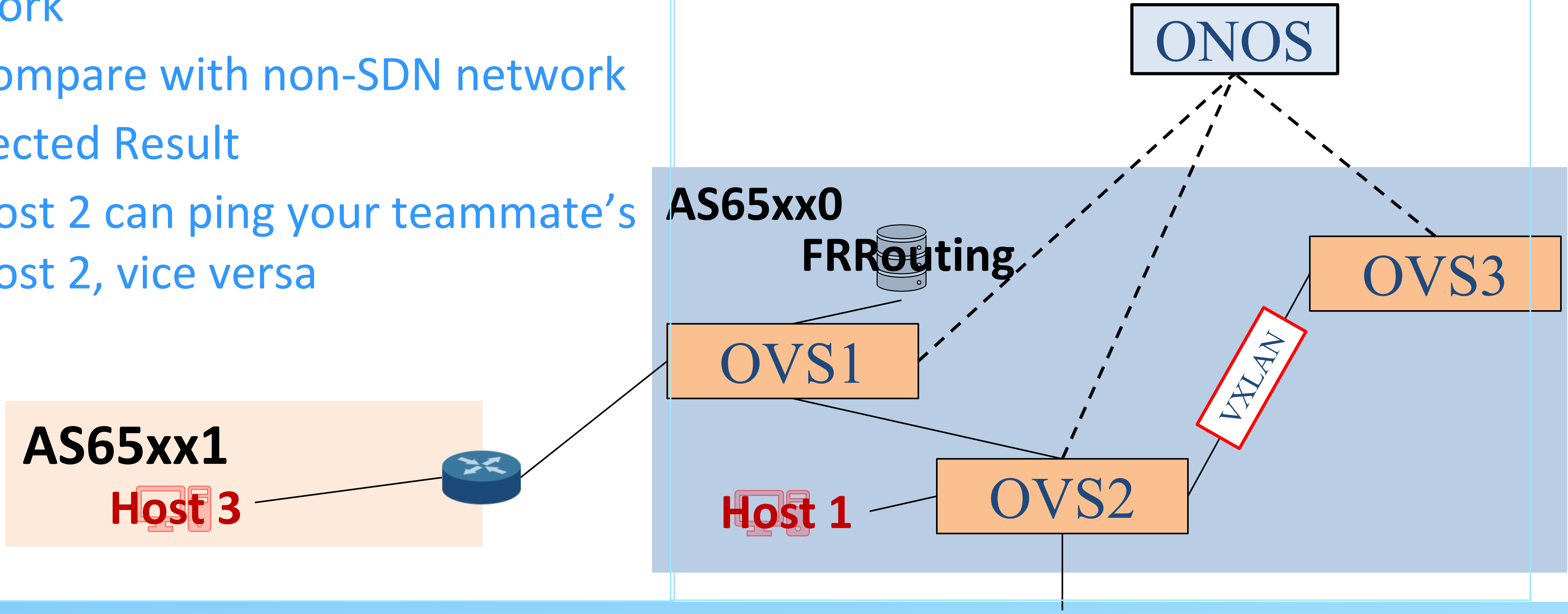
- **IMPORTANT**
  - **`mtr` (traceroute tool) shall not see the packets pass FRRouting.**



vRouter Verifications – Transit Traffic

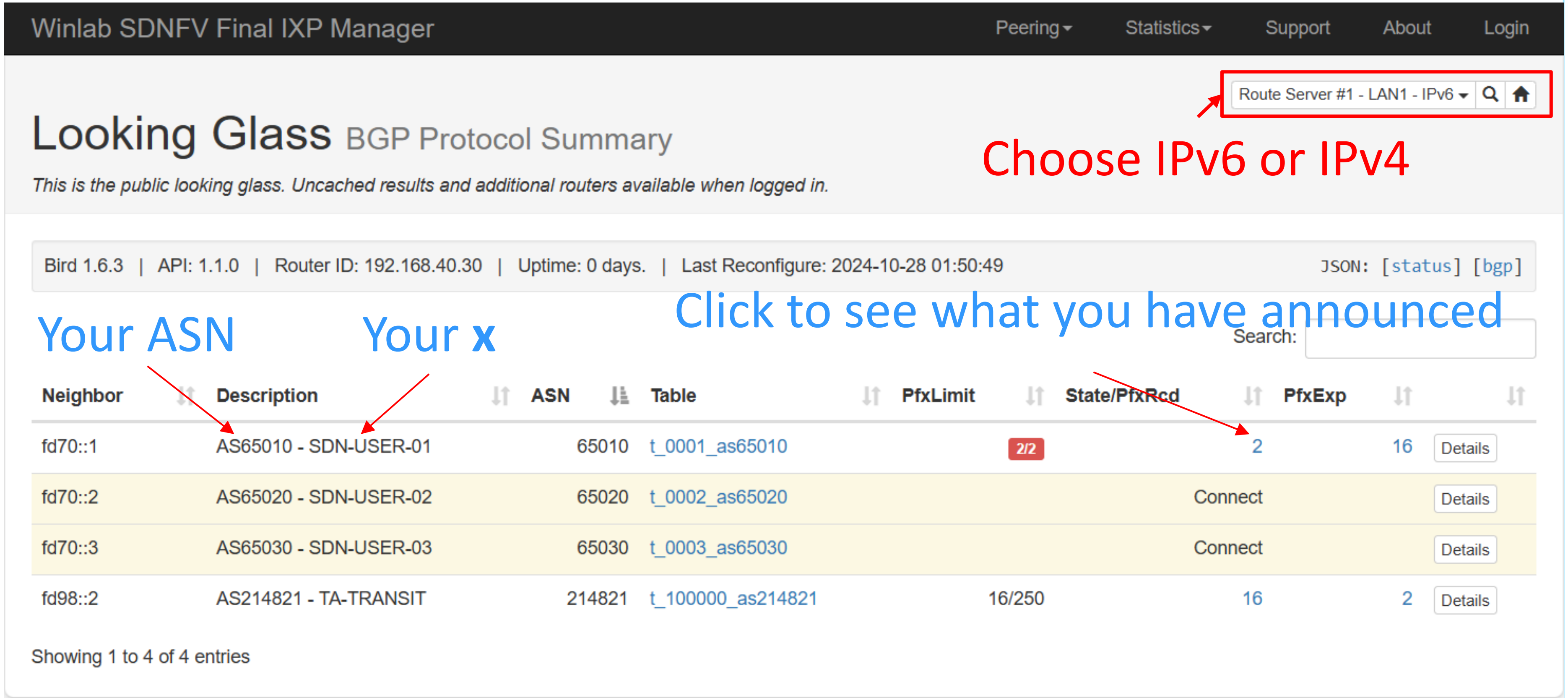
- Assure Transit Traffics
  - Get use of `routes` in OVS
  - Hint
    - Think again about how routers work
    - Compare with non-SDN network
  - Expected Result
    - Host 2 can ping your teammate’s Host 2, vice versa

- **IMPORTANT**
  - **`mtr` (traceroute tool) shall not see the packets pass FRRouting.**



SDN Requirements

- You can see what you have announced via IXP Manager
- http://140.113.60.186:8880/lg



Winlab SDNFV Final IXP Manager

Choose IPv6 or IPv4

Looking Glass BGP Protocol Summary

This is the public looking glass. Uncached results and additional routers available when logged in.

Bird 1.6.3 | API: 1.1.0 | Router ID: 192.168.40.30 | Uptime: 0 days. | Last Reconfigure: 2024-10-28 01:50:49

JSON: [status] [bgp]

Your ASN

Your **x**

Click to see what you have announced

Search:

Neighbor	Description	ASN	Table	PfxLimit	State/PfxRcd	PfxExp	
fd70::1	AS65010 - SDN-USER-01	65010	t_0001_as65010	2/2	2	16	Details
fd70::2	AS65020 - SDN-USER-02	65020	t_0002_as65020		Connect		Details
fd70::3	AS65030 - SDN-USER-03	65030	t_0003_as65030		Connect		Details
fd98::2	AS214821 - TA-TRANSIT	214821	t_100000_as214821	16/250	16	2	Details

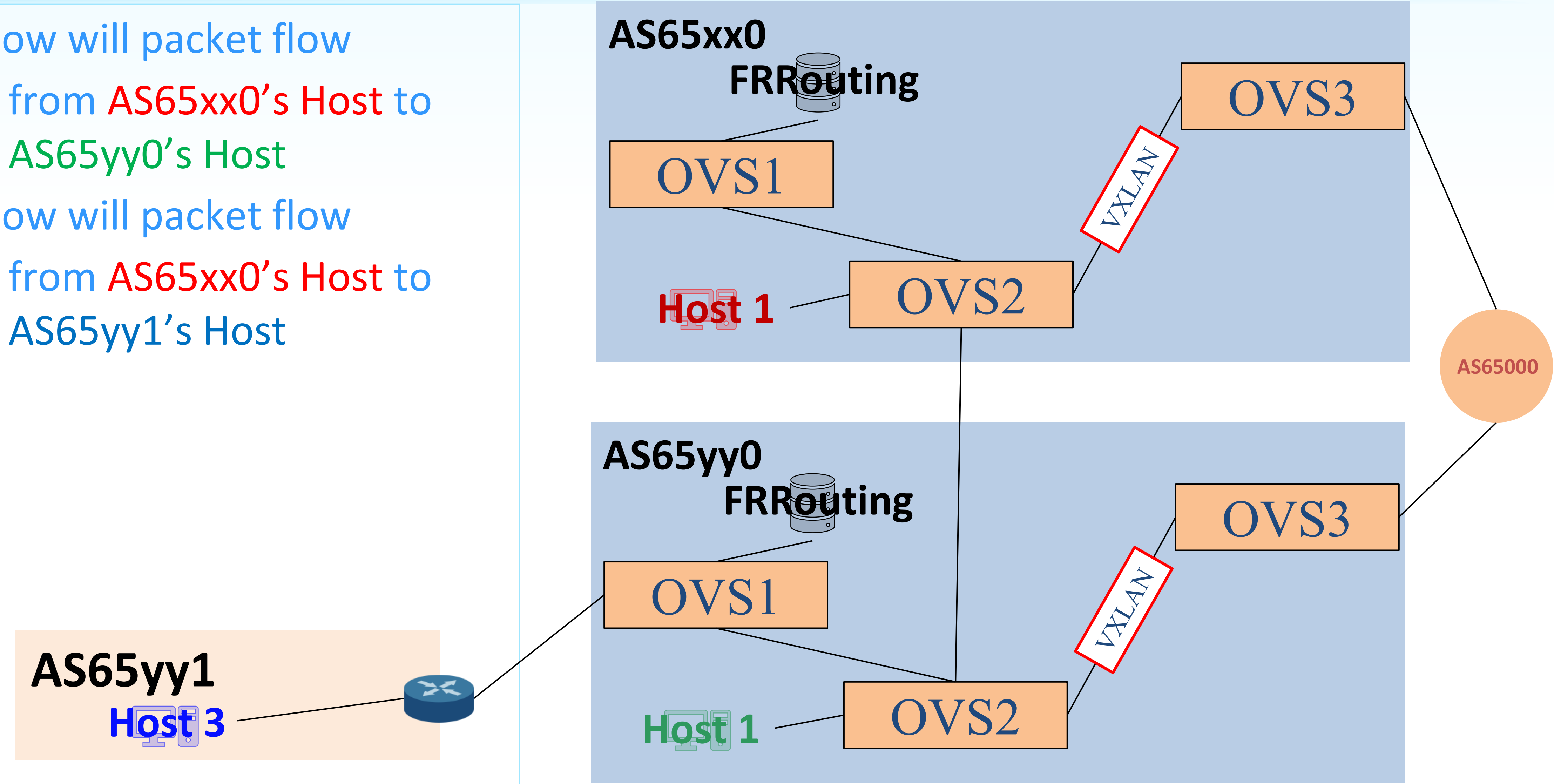
Showing 1 to 4 of 4 entries

vRouter Verifications – Peer Traffic

- AS65xx0 only advertises its own prefix to TA’s Router.
  - AS65yy0 will not receive AS65xx1’s prefix from AS65xx0.
  - TA’s Router will not receive AS65yy0, AS65zz0… from AS65xx0
    - Only two prefixes will be advertised to TA’s Router.
- Hint
  - AS65yy0 will receive AS65xx1’s prefix from AS65000
    - What does this mean?
  - Think twice
    - How will packet flow from AS65xx0’s Host to AS65yy1’s Host
    - How will packet flow from AS65xx1’s Host to AS65yy1’s Host

vRouter Verifications – Peer Traffic (cont.)

- How will packet flow
  - from AS65xx0’s Host to AS65yy0’s Host
- How will packet flow
  - from AS65xx0’s Host to AS65yy1’s Host



vRouter Verifications – Anycast Web Container

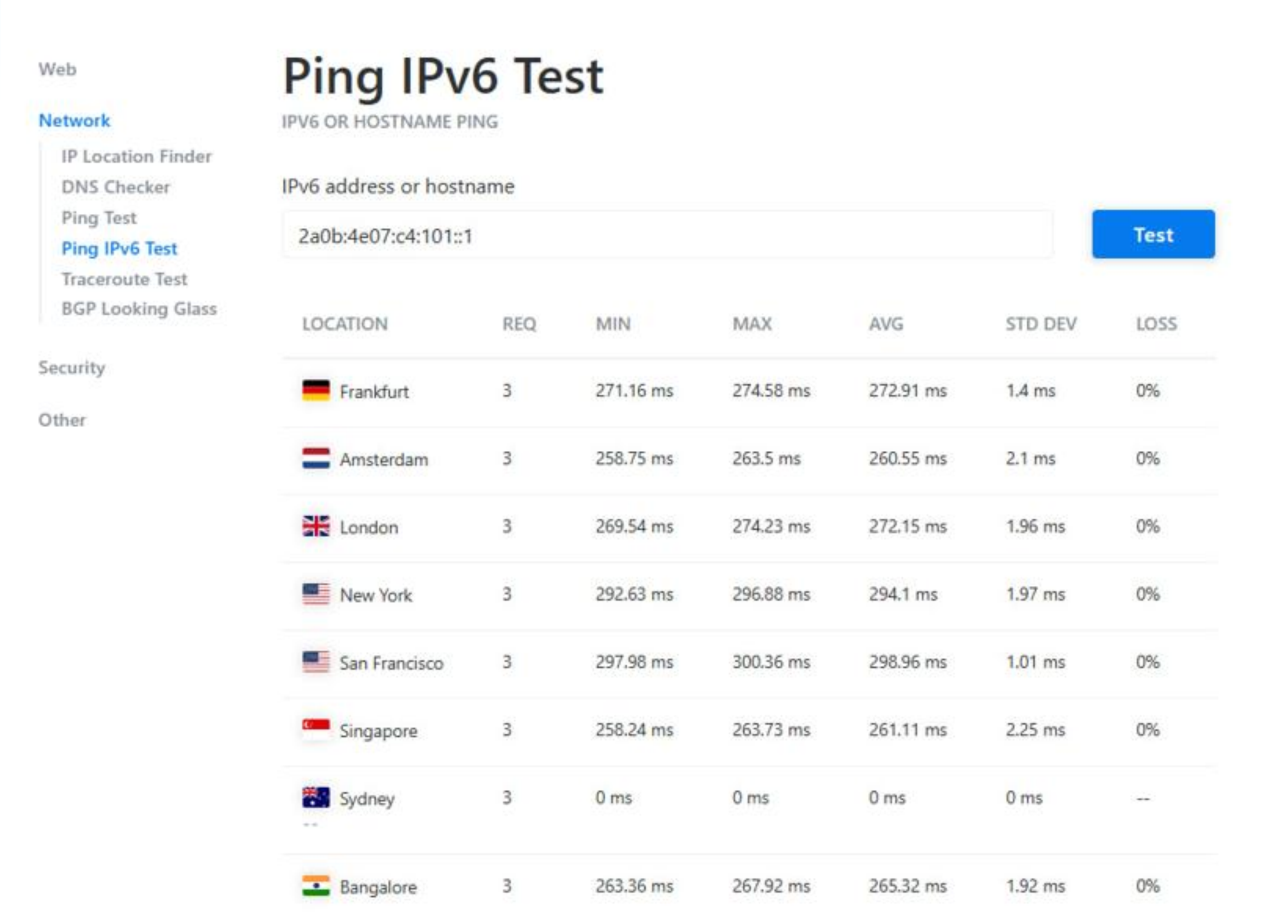
- Requirement
  - Add two web container in AS65xx0 with the same IP, one on OVS1 and one on OVS2.
  - Use traefik/whoami as image
  - Why?
    - OVS1 and OVS2 might be in different physical zone (America vs Taiwan)
    - OVS1 and OVS2 have very high latency
    - OVS1 and VOS2’s link is very expensive
    - An anycast web container
  - Expected Result
    - Requests near OVS1 will receive OVS1’s web container’s result
    - Requests near OVS2 will receive OVS2’s web container’s result

vRouter Verifications – Anycast Web Container

- An example
  - Since AS65xx1’s Host (Host 3) is closer to OVS1; therefore, curl shall response OVS1’s web container result.
  - Since AS65yy0’1 Host (Your teammates Host 1) is closer to OVS2; therefore, curl shall response OVS2’s web container result.
- Hint
  - Anycast is a very tricky part; you have to look closely at what happens in the topology.
  - This might crash your whole work. Please be careful and perhaps use version control methods while doing your project.

SDN Requirements

- You can see that you have connected to the outside world
- https://tools.keycdn.com/ipv6-ping



Ping IPv6 Test

IPV6 OR HOSTNAME PING

IPv6 address or hostname: 2a0b:4e07:c4:101::1

LOCATION	REQ	MIN	MAX	AVG	STD DEV	LOSS
Frankfurt	3	271.16 ms	274.58 ms	272.91 ms	1.4 ms	0%
Amsterdam	3	258.75 ms	263.5 ms	260.55 ms	2.1 ms	0%
London	3	269.54 ms	274.23 ms	272.15 ms	1.96 ms	0%
New York	3	292.63 ms	296.88 ms	294.1 ms	1.97 ms	0%
San Francisco	3	297.98 ms	300.36 ms	298.96 ms	1.01 ms	0%
Singapore	3	258.24 ms	263.73 ms	261.11 ms	2.25 ms	0%
Sydney --	3	0 ms	0 ms	0 ms	0 ms	--
Bangalore	3	263.36 ms	267.92 ms	265.32 ms	1.92 ms	0%

OUTLINE

- Router Comparison
- Service in Use
- Router’s activity
- Project Setup
- Final Project Topology
- Scoring Criteria and Verifications
- References

TA Contacts

- If you have any problem
  - Mail to me
  - Register demo time for help

- Final Project
  - Demo Register: [Bookings - Outlook]
  - Deadline: 12/??

Reference

- ovs-vsctl(8) - Linux manual page (https://man7.org/linux/man-pages/man8/ovs-vsctl.8.html)
- ip(8) - Linux man page (https://linux.die.net/man/8/ip)