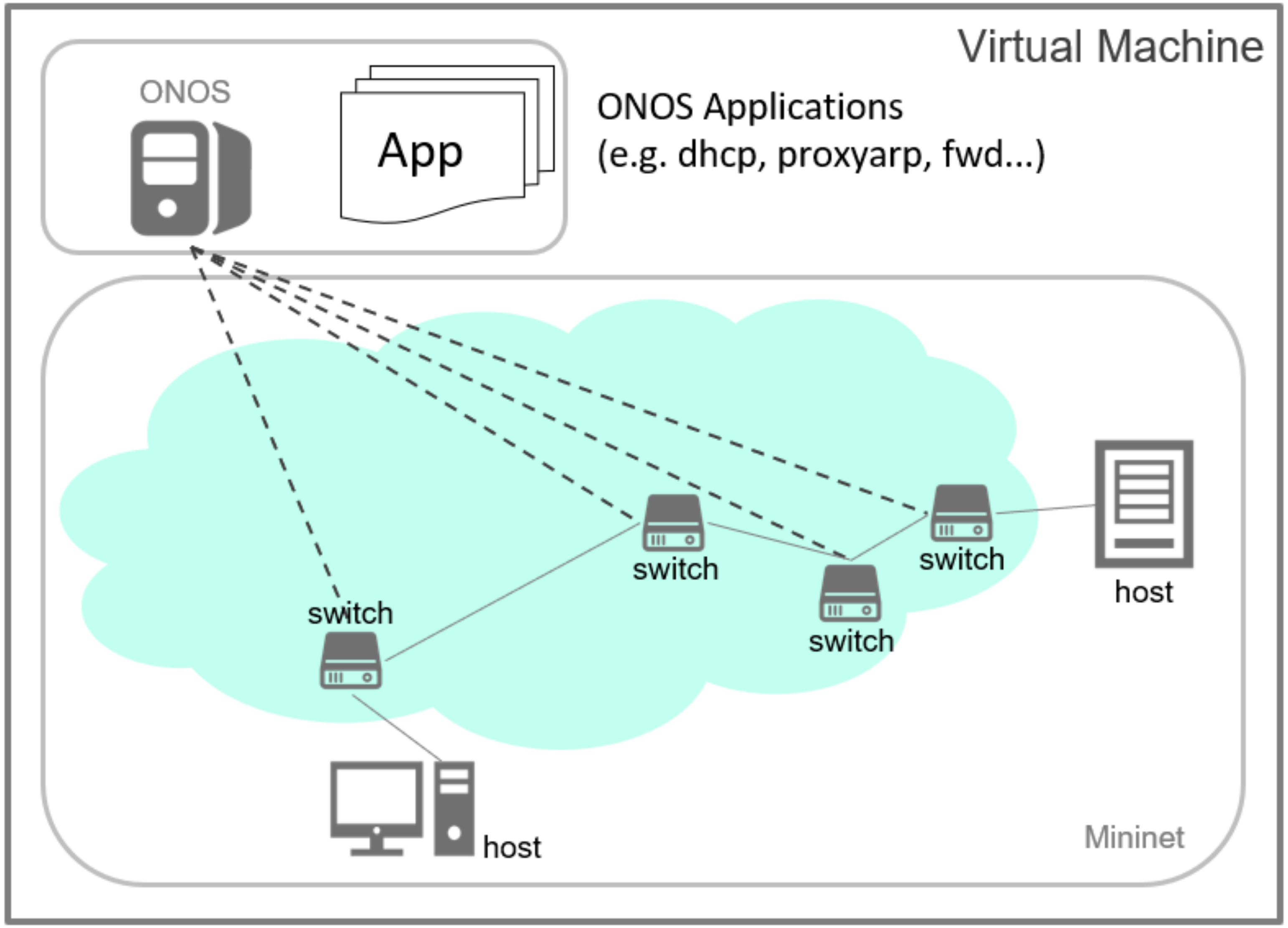
# Lab 3
# ONOS Application Development: SDN-enabled Learning Bridge

**Deadline: 2023/10/26 (THUR) 23:59**

# Outline

## Outline

# Overview


Virtual Machine
ONOS
App
ONOS Applications
(e.g. dhcp, proxyarp, fwd...)
switch
switch
switch
switch
host
host
Mininet

# Outline

# JDK Installation

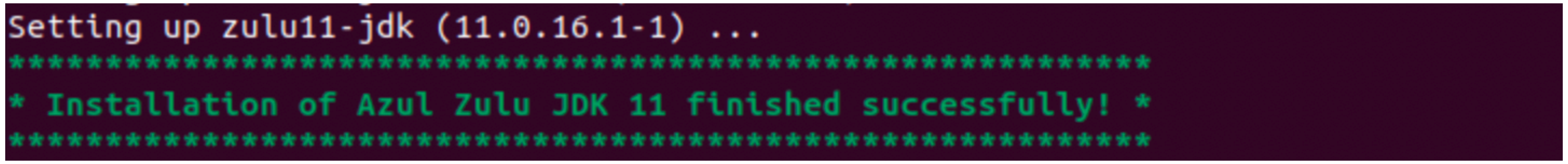
1. Download the “install_jdk” script from E3.
2. Add execution permission to the script:

```
$ chmod +x install_jdk
```

3. Execute the script:

```
$ ./install_jdk
```

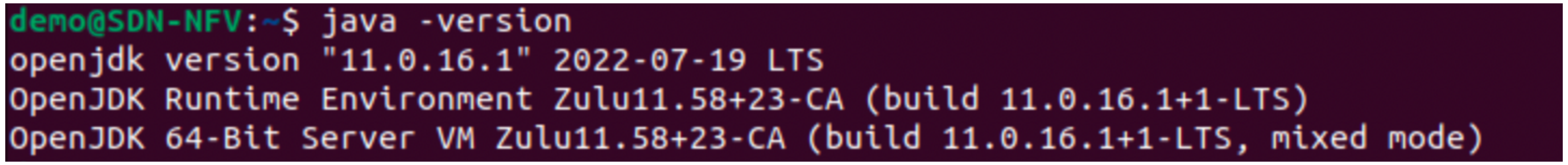
4. Once the installation finishes, you will see a success message:



```
Setting up zulu11-jdk (11.0.16.1-1) ...
************************************************************
* Installation of Azul Zulu JDK 11 finished successfully! *
************************************************************
```

5. Check the installed JDK version:

```
$ java -version
```



```
demo@SDN-NFV:~$ java -version
openjdk version "11.0.16.1" 2022-07-19 LTS
OpenJDK Runtime Environment Zulu11.58+23-CA (build 11.0.16.1+1-LTS)
OpenJDK 64-Bit Server VM Zulu11.58+23-CA (build 11.0.16.1+1-LTS, mixed mode)
```

## Apache Maven

- A software project management and comprehension tool.
- Based on the concept of a Project Object Model (POM).
- Manage a project's build, reporting and documentation from a **central piece** of information.
- It has been intalled in your VM by the “env_setup” script in Lab 1.
- Official website: https://maven.apache.org/

# Build ONOS Application Archetypes

- We will use ***onos-create-app*** command to generate an ONOS application template.
- ***onos-create-app*** command relies on the ONOS archetypes.
- We need to build ONOS archetypes first.
- Steps:
  - Specify ONOS version:

```
$ export ONOS_POM_VERSION=2.7.0
```

  - Build archetypes:

```
$ cd $ONOS_ROOT/tools/package/archetypes
$ mvn clean install -DskipTests
```

    - ***-DskipTests***: Skip running tests of the project.

# Outline

## Build ONOS Application Template

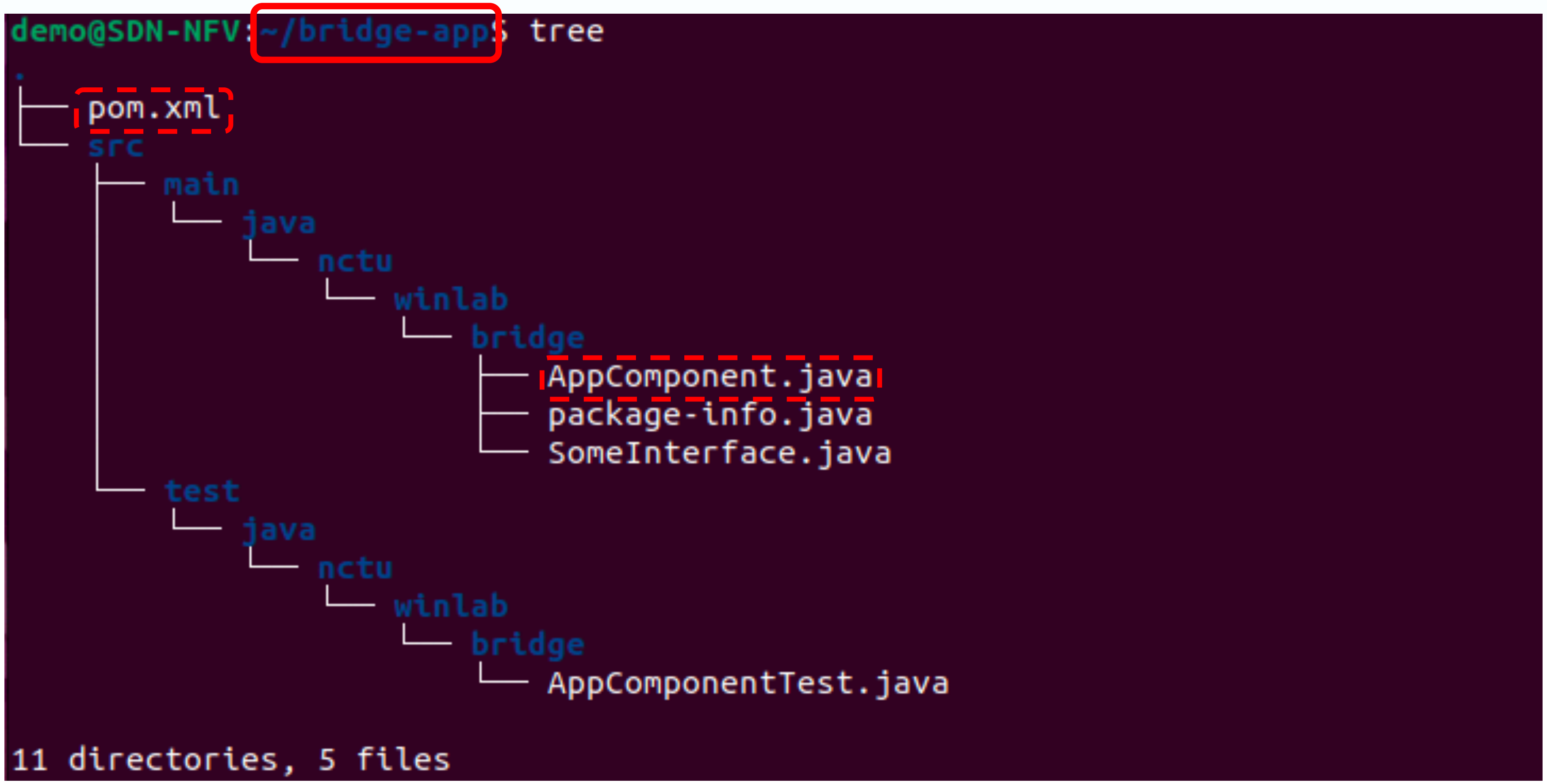
- Run ***onos-create-app***.

```
$ onos-create-app
...
[INFO] ...
Define value for property 'groupId': nctu.winlab
Define value for property 'artifactId': bridge-app
Define value for property 'version' 1.0-SNAPSHOT: : <enter>
Define value for property 'package' nctu.winlab: : nctu.winlab.bridge

Confirm properties configuration:
onosVersion: 2.7.0
groupId: nctu.winlab
artifactId: bridge-app
version: 1.0-SNAPSHOT
package: nctu.winlab.bridge
 Y: : <enter>
[INFO] ...
...
[INFO] BUILD SUCCESS
```

Archive ID for the created ONOS application.

# Folder Structure of Created ONOS Application

- ***onos-create-app*** command creates a folder named **bridge-app** (artifactId).
- Structure of **bridge-app** folder:

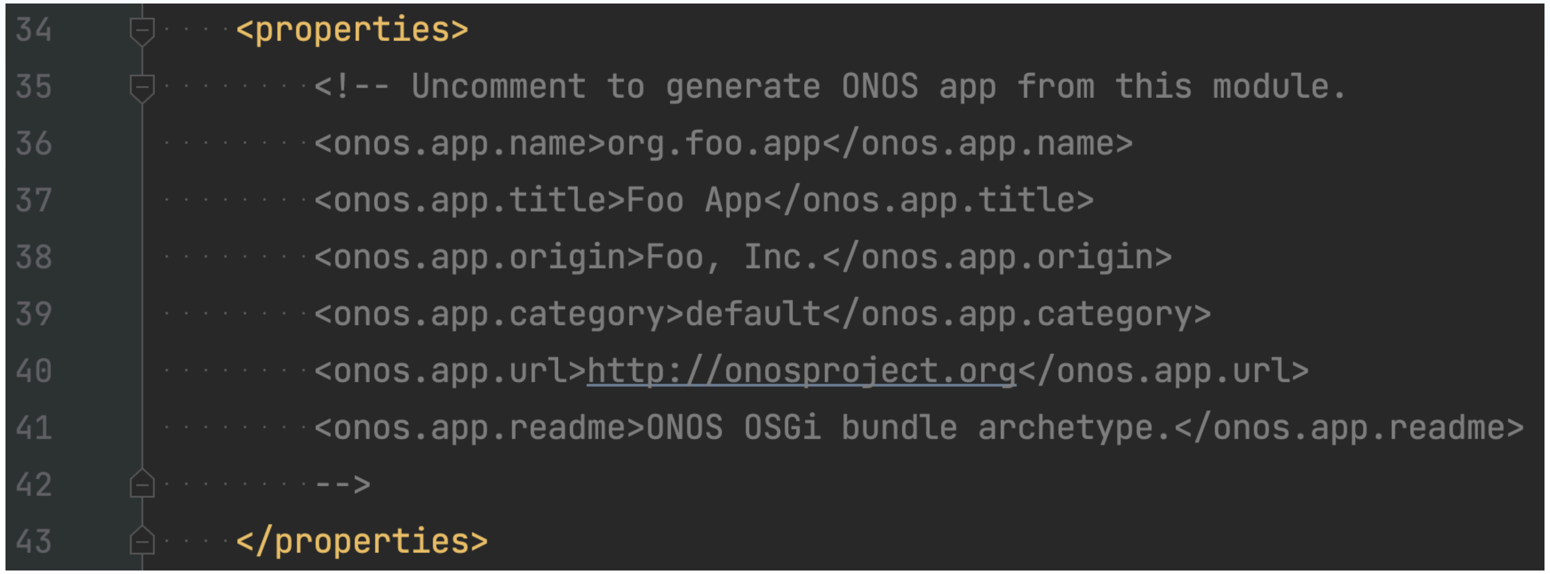


```
demo@SDN-NFV:~/bridge-app$ tree
.
├── pom.xml
└── src
    ├── main
    │   └── java
    │       └── nctu
    │           └── winlab
    │               └── bridge
    │                   ├── AppComponent.java
    │                   ├── package-info.java
    │                   └── SomeInterface.java
    └── test
        └── java
            └── nctu
                └── winlab
                    └── bridge
                        └── AppComponentTest.java

11 directories, 5 files
```

# Modify ONOS Application Properties

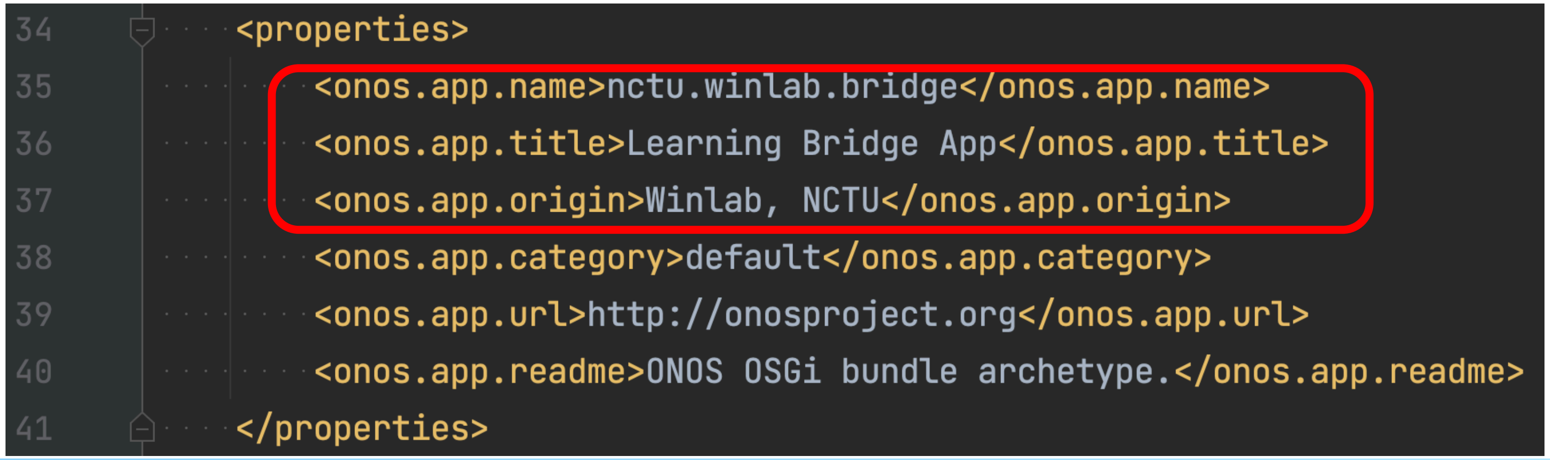
- Modify Project Object Model file pom.xml to describe your project.

pom.xml
**Before**



```
    <properties>
        <!-- Uncomment to generate ONOS app from this module.
        <onos.app.name>org.foo.app</onos.app.name>
        <onos.app.title>Foo App</onos.app.title>
        <onos.app.origin>Foo, Inc.</onos.app.origin>
        <onos.app.category>default</onos.app.category>
        <onos.app.url>http://onosproject.org</onos.app.url>
        <onos.app.readme>ONOS OSGi bundle archetype.</onos.app.readme>
        -->
    </properties>
```

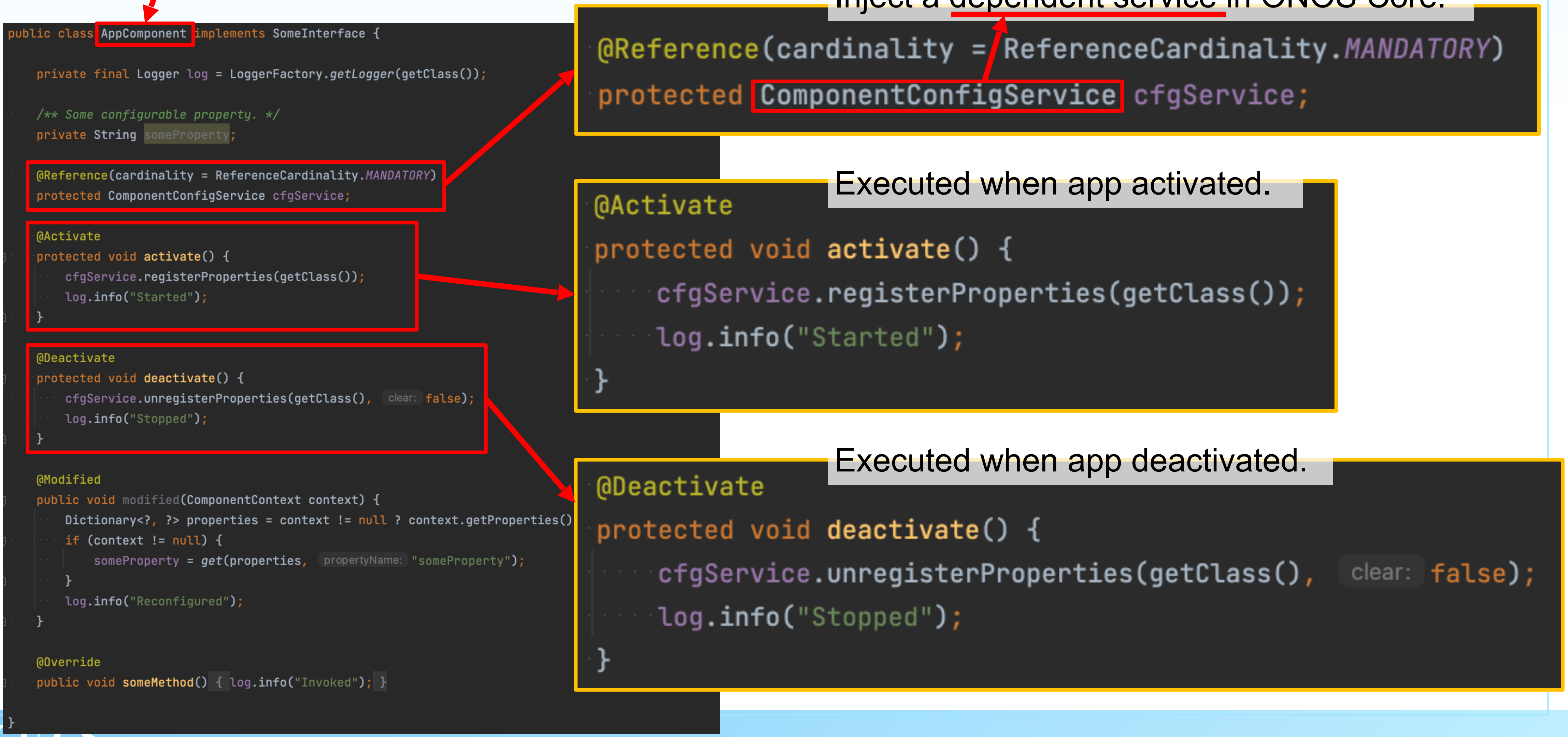
pom.xml
**After**



```
    <properties>
        <onos.app.name>nctu.winlab.bridge</onos.app.name>
        <onos.app.title>Learning Bridge App</onos.app.title>
        <onos.app.origin>Winlab, NCTU</onos.app.origin>
        <onos.app.category>default</onos.app.category>
        <onos.app.url>http://onosproject.org</onos.app.url>
        <onos.app.readme>ONOS OSGi bundle archetype.</onos.app.readme>
    </properties>
```

# Overview of AppComponent.java

- AppComponent.java code template.



```java
public class AppComponent implements SomeInterface {

    private final Logger log = LoggerFactory.getLogger(getClass());

    /** Some configurable property. */
    private String someProperty;

    @Reference(cardinality = ReferenceCardinality.MANDATORY)
    protected ComponentConfigService cfgService;

    @Activate
    protected void activate() {
        cfgService.registerProperties(getClass());
        log.info("Started");
    }

    @Deactivate
    protected void deactivate() {
        cfgService.unregisterProperties(getClass(),  clear: false);
        log.info("Stopped");
    }

    @Modified
    public void modified(ComponentContext context) {
        Dictionary<?, ?> properties = context != null ? context.getProperties()
        if (context != null) {
            someProperty = get(properties,  propertyName: "someProperty");
        }
        log.info("Reconfigured");
    }

    @Override
    public void someMethod() { log.info("Invoked"); }

}
```

Inject a dependent service in ONOS Core.

```java
@Reference(cardinality = ReferenceCardinality.MANDATORY)
protected ComponentConfigService cfgService;
```

Executed when app activated.

```java
@Activate
protected void activate() {
    cfgService.registerProperties(getClass());
    log.info("Started");
}
```

Executed when app deactivated.

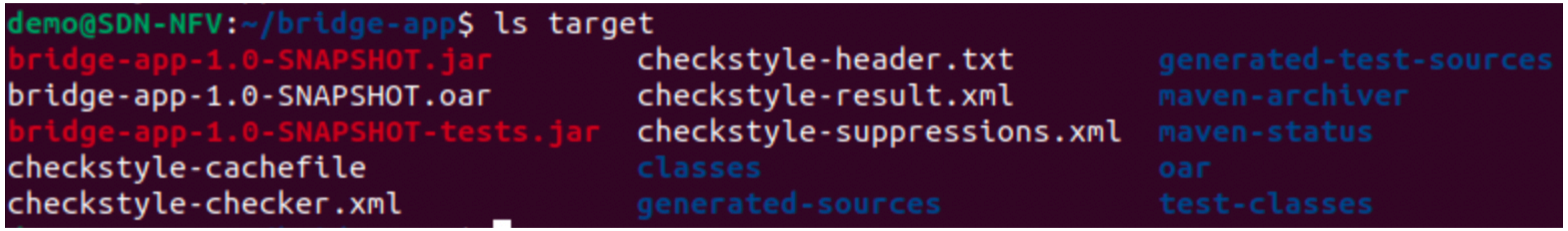
```java
@Deactivate
protected void deactivate() {
    cfgService.unregisterProperties(getClass(),  clear: false);
    log.info("Stopped");
}
```

## Outline

# Build, Install and Activate ONOS Application

- Build ONOS application:

```
# In the root of your application folder.
$ mvn clean install -DskipTests
```



```
demo@SDN-NFV:~/bridge-app$ ls target
bridge-app-1.0-SNAPSHOT.jar        checkstyle-header.txt        generated-test-sources
bridge-app-1.0-SNAPSHOT.oar        checkstyle-result.xml        maven-archiver
bridge-app-1.0-SNAPSHOT-tests.jar  checkstyle-suppressions.xml  maven-status
checkstyle-cachefile               classes                      oar
checkstyle-checker.xml             generated-sources            test-classes
```

} built results

- Run ONOS:

```
$ cd $ONOS_ROOT
$ bazel run onos-local -- clean debug
```

- Install and activate ONOS application:

```
# In the root of your application folder.
$ onos-app localhost install! target/<artifactId>-<version>.oar
                                      (brige-app-1.0-SNAPSHOT.oar)
```

  - ***install!***: Install and activate application immediately.

# Outline

- Overview
- **Build ONOS Application Project**
  - Environment Setup
  - Create an ONOS Application
  - Build, Install, and Activate ONOS Application
  - **Reinstall ONOS Application**
- Learning Bridge Function
  - Introduction
  - Workflow
- Project 3 Requirements
  - Create ONOS Application (10%)
  - Learning Bridge Function (60%)
  - Flow Rule Regulation (20%)
  - Submission Naming Convention (10%)
  - Restrictions

# Reinstall ONOS Application

If you modify your application, you need to rebuild and reinstall it on ONOS.

1. Rebuild application of new version:

```
# In the root of your application folder.
$ mvn clean install -DskipTests
```

2. Deactivate application of old version on ONOS:

```
$ onos-app localhost deactivate <onos.app.name>
```

- ***<onos.app.name>*** is set in your pom.xml. e.g. nctu.winlab.bridge

3. Uninstall application of old version:

```
$ onos-app localhost uninstall <onos.app.name>
```

4. Install and activate application of new version:

```
# In the root of your application folder.
$ onos-app localhost install! target/<artifactId>-<version>.oar
```

## References

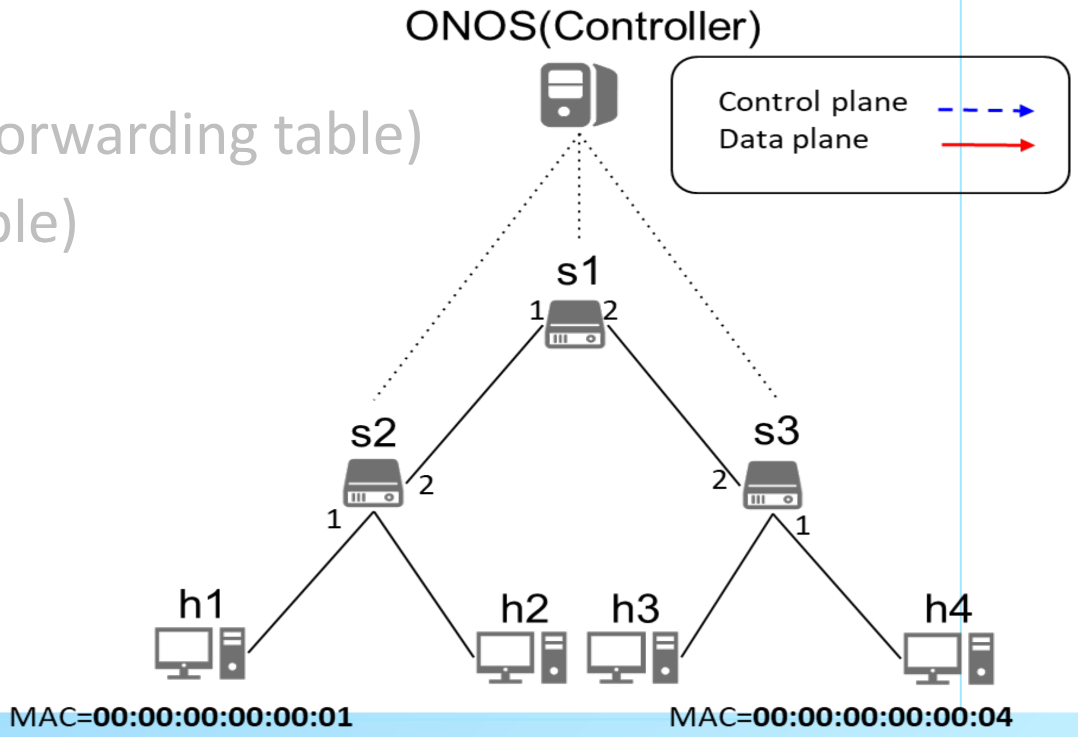
- Install Azul Zulu on Debian-based Linux
- ONOS Wiki – Template Application Tutorial
- ONOS Application Subsystem
- ONOS Java API (2.7.0)

# Outline

# Learning Bridge Functionality

- Switch functionality:
  - When receives a packet, matches Destination MAC
    - Matched: Forwards packet via specified port
    - Not matched: Packet-in
- ONOS App functionality:
  - When receives a Packet-in
    - Records Source MAC and incoming port (in forwarding table)
    - Looks up Destination MAC (in forwarding table)
      a. Not found:
         - Floods Packet-out.
      b. Found:
         - Sends Packet-out via designated port.
         - Installs flow rule on switch.

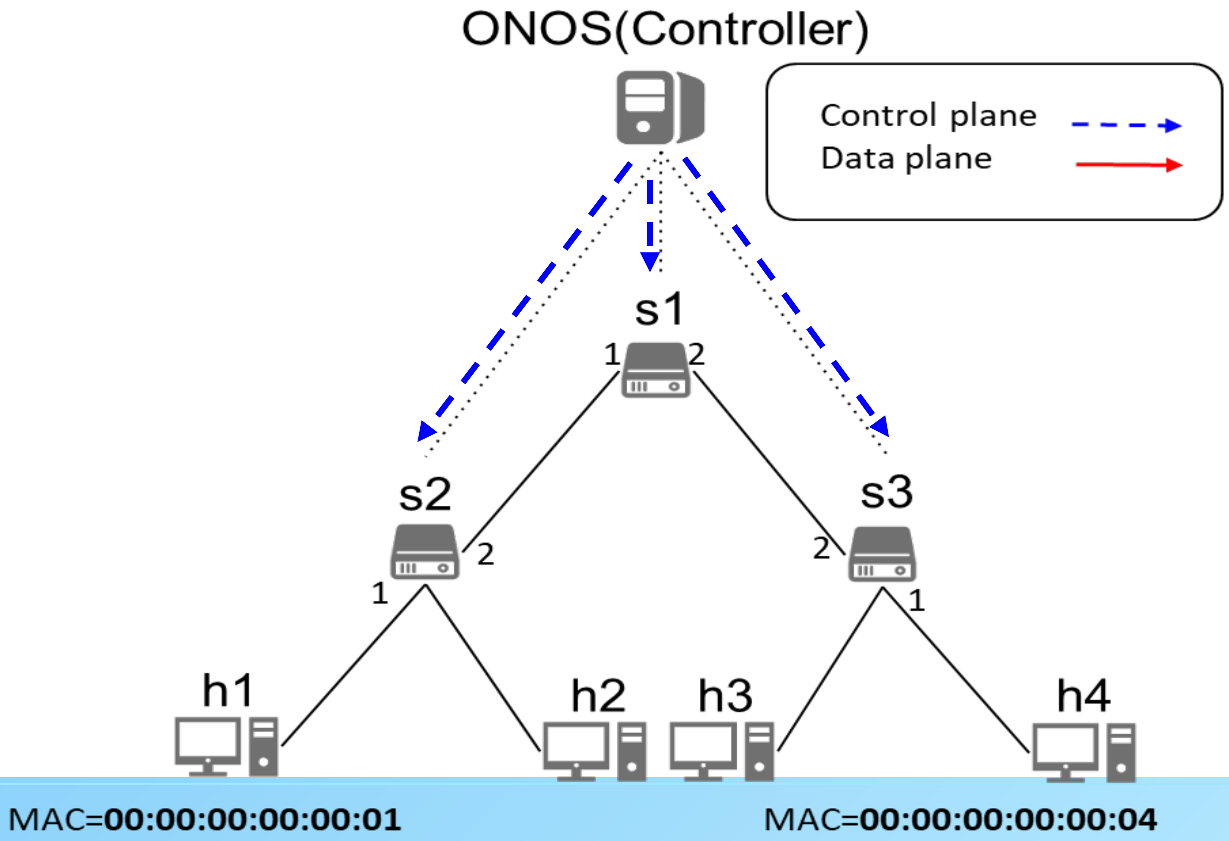
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| | | | | | |
| | | | | | |



## Outline

## Request for Packet-in

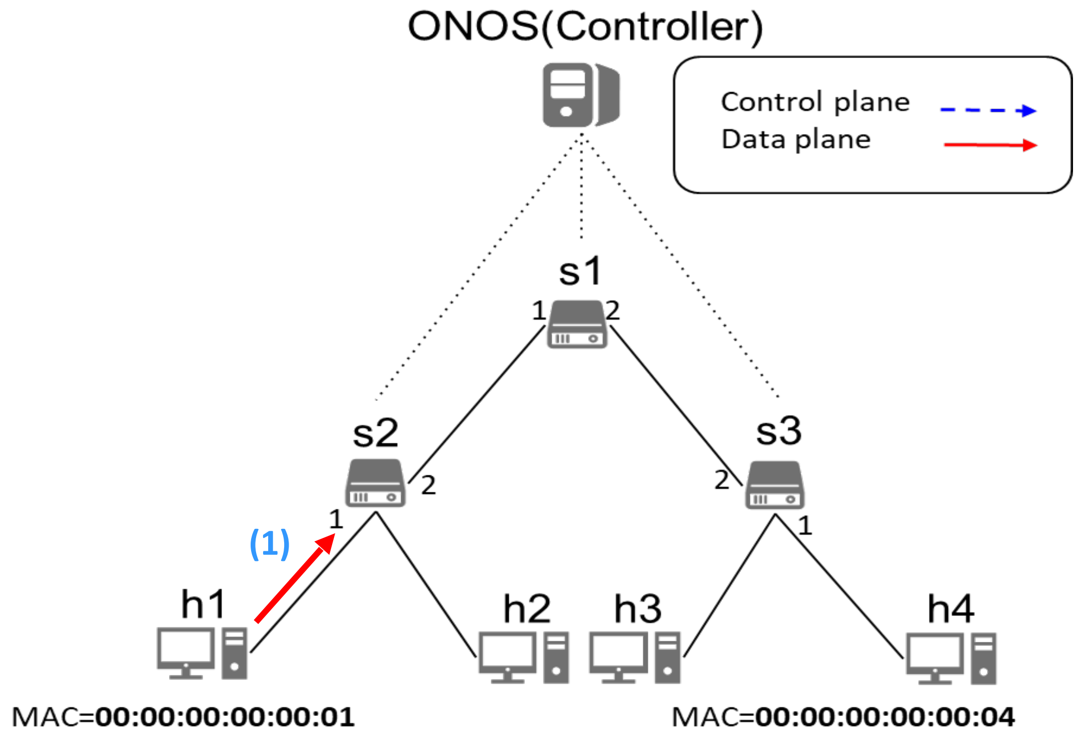
- When App is activated, it installs a rule on each switch.
  - To request Packet-in for IPv4 packets.
  - With very low priority.
- Don’t forget to cancel the request for Packet-in when your App is deactivated.



# Workflow (h1 -> h4)

1. h1 pings h4.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h4 receives packet from h1.

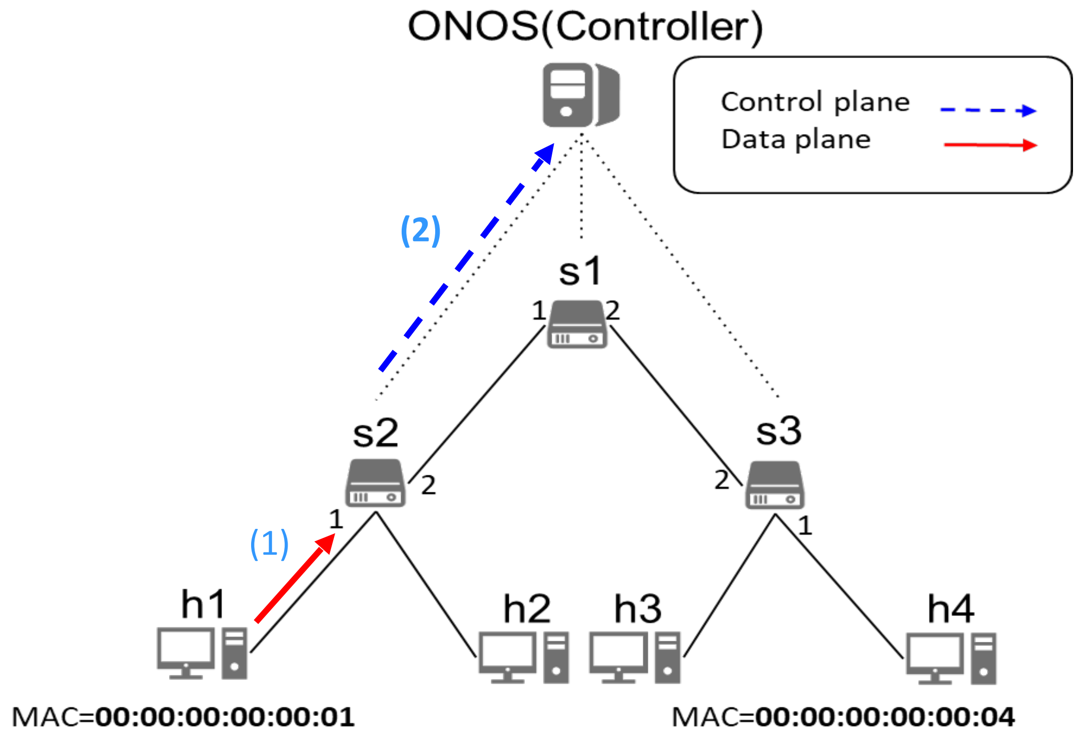
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| | | | | | |
| | | | | | |



# Workflow (h1 -> h4)

1. h1 pings h4.
2. Switch (s2) sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h4 receives packet from h1.

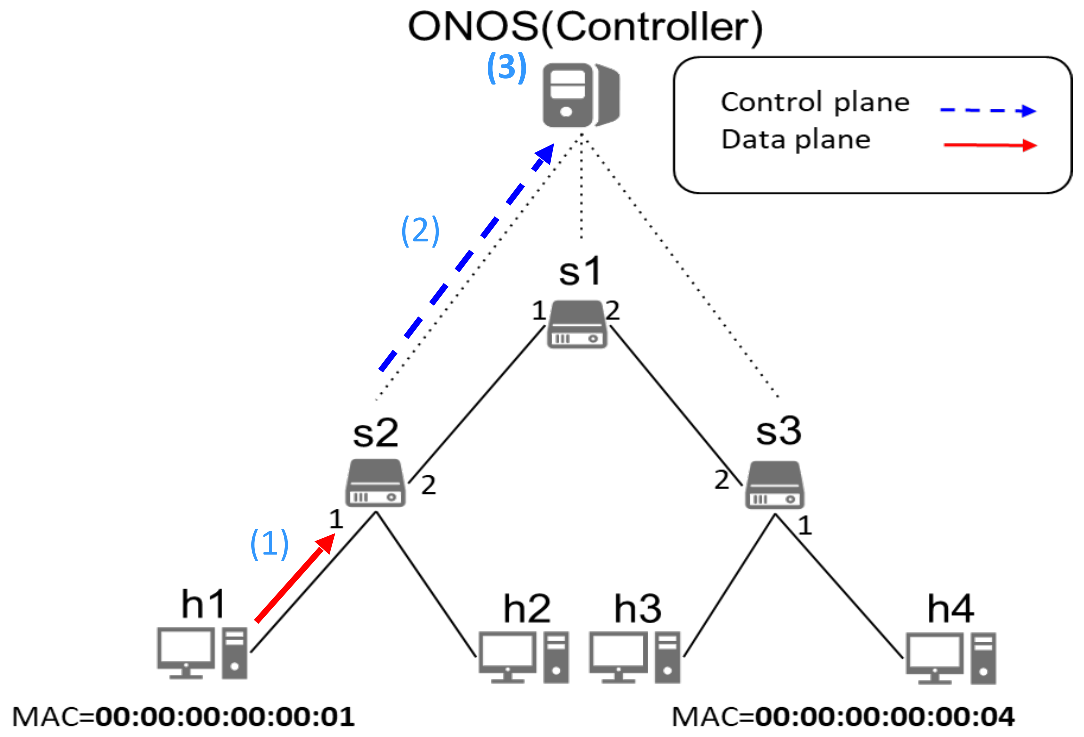
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| | | | | | |
| | | | | | |



# Workflow (h1 -> h4)

1. h1 pings h4.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h4 receives packet from h1.

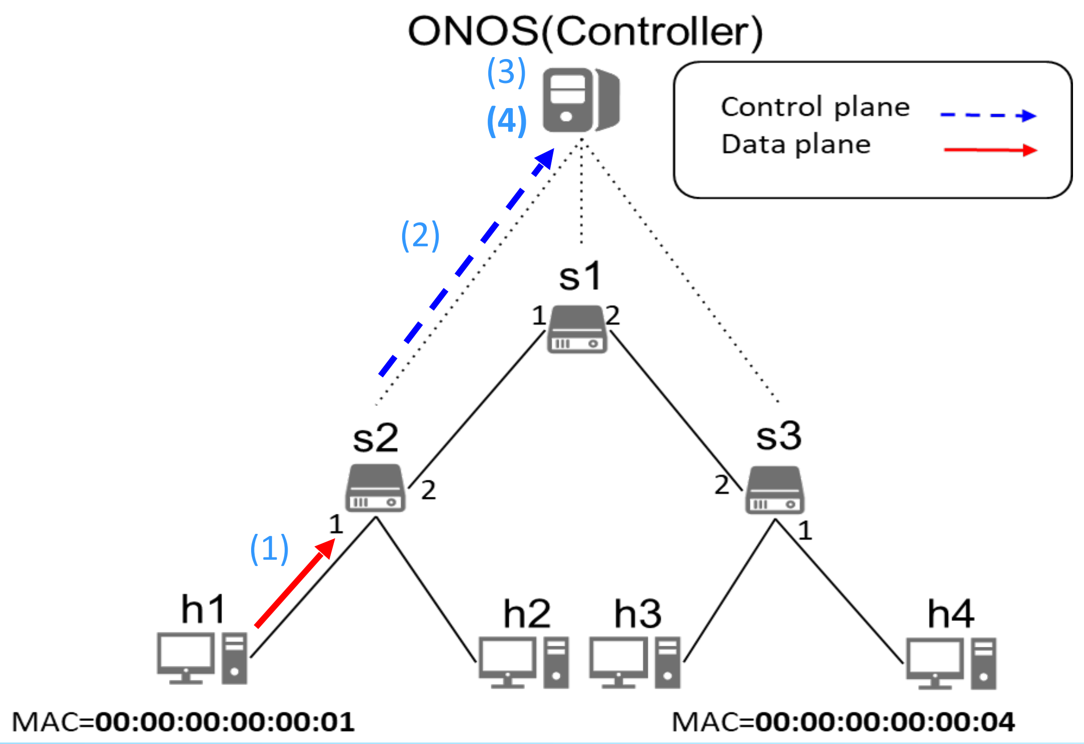
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| | | 00:......:01 | 1 | | |
| | | | | | |



# Workflow (h1 -> h4)

1. h1 pings h4.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h4 receives packet from h1.

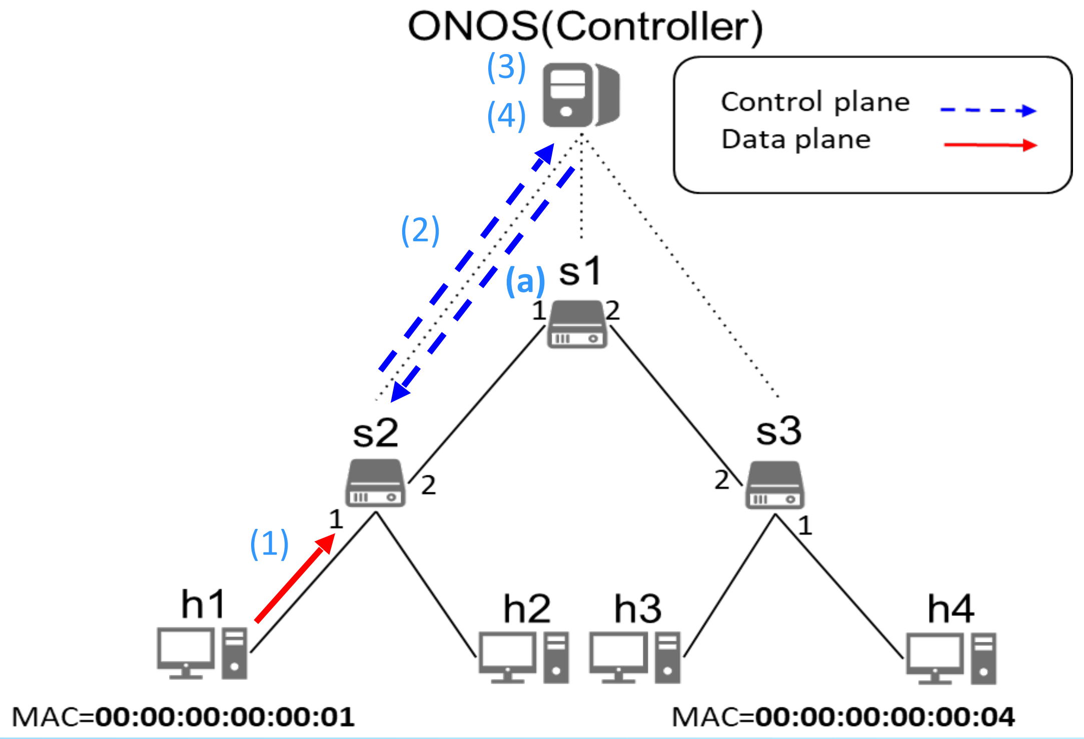
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| | | 00:......:01 | 1 | | |
| | | | | | |



# Workflow (h1 -> h4)

1. h1 pings h4.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h4 receives packet from h1.

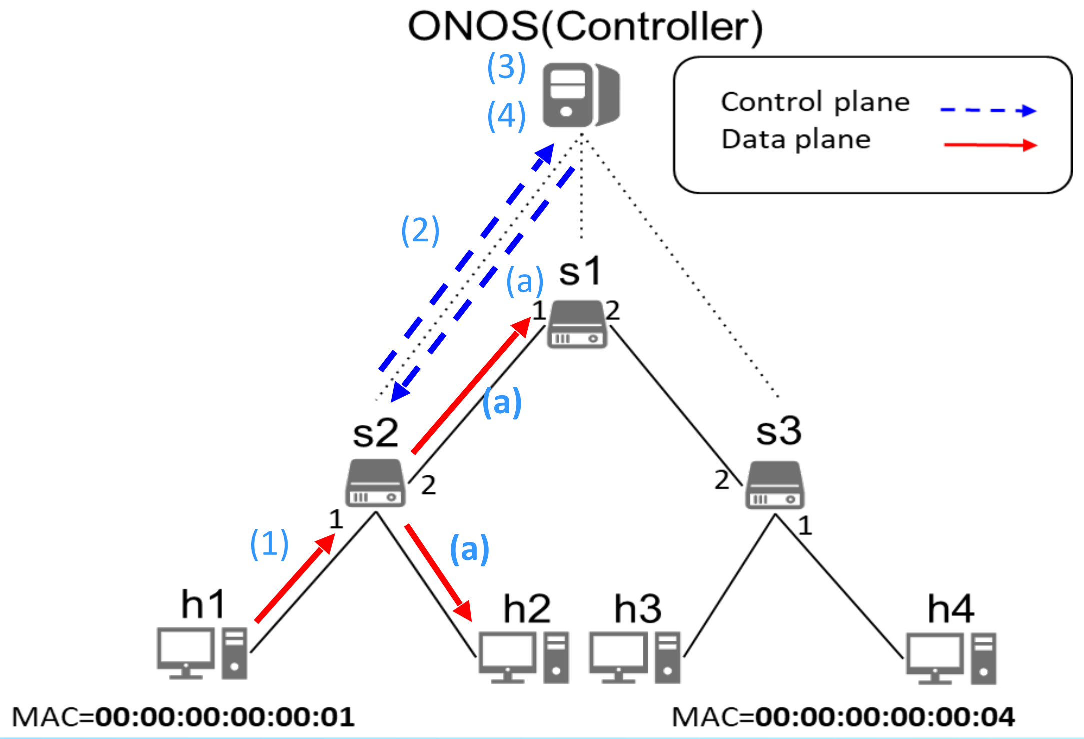
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| | | 00:......:01 | 1 | | |
| | | | | | |



# Workflow (h1 -> h4)

1. h1 pings h4.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h4 receives packet from h1.

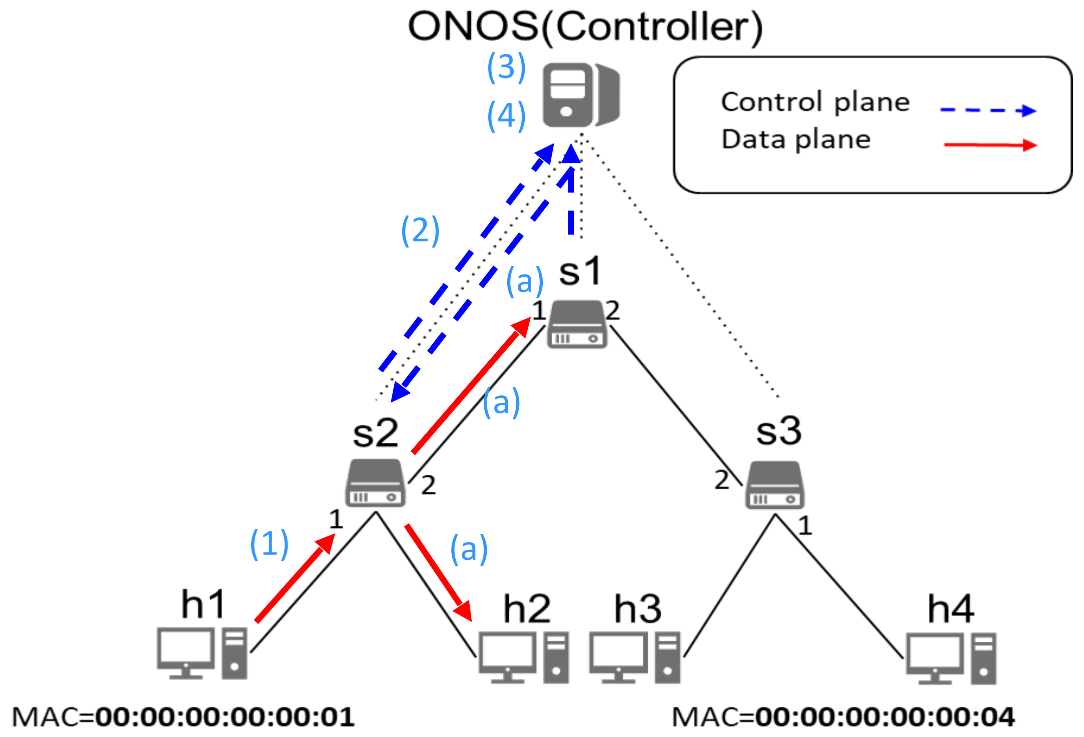
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| | | 00:......:01 | 1 | | |
| | | | | | |



# Workflow (h1 -> h4)

1. h1 pings h4.
2. Switch (s1) sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
    a. Destination MAC not found:
        - Floods Packet-out.
    b. Destination MAC found:
        - Sends Packet-out via designated port.
        - Installs flow rule on switch.
5. h4 receives packet from h1.

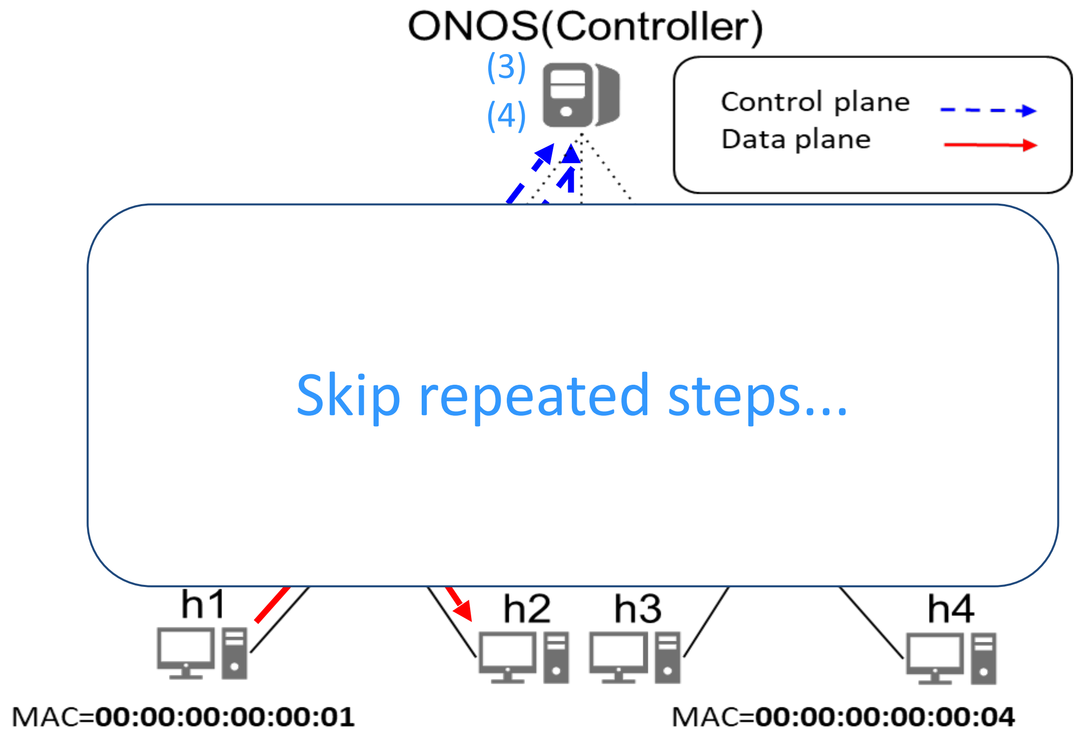
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| | | 00:......:01 | 1 | | |
| | | | | | |



# Workflow (h1 -> h4)

1. h1 pings h4.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h4 receives packet from h1.

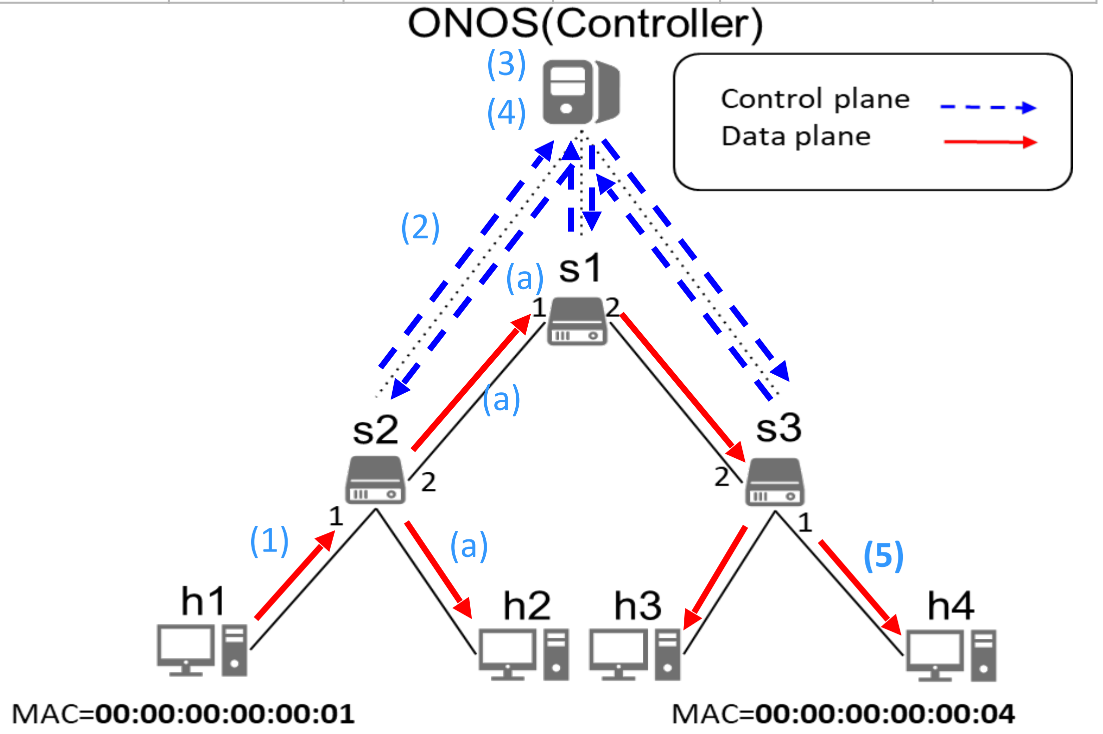
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| | | 00:......:01 | 1 | | |
| | | | | | |



# Workflow (h1 -> h4)

1. h1 pings h4.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h4 receives packet from h1.

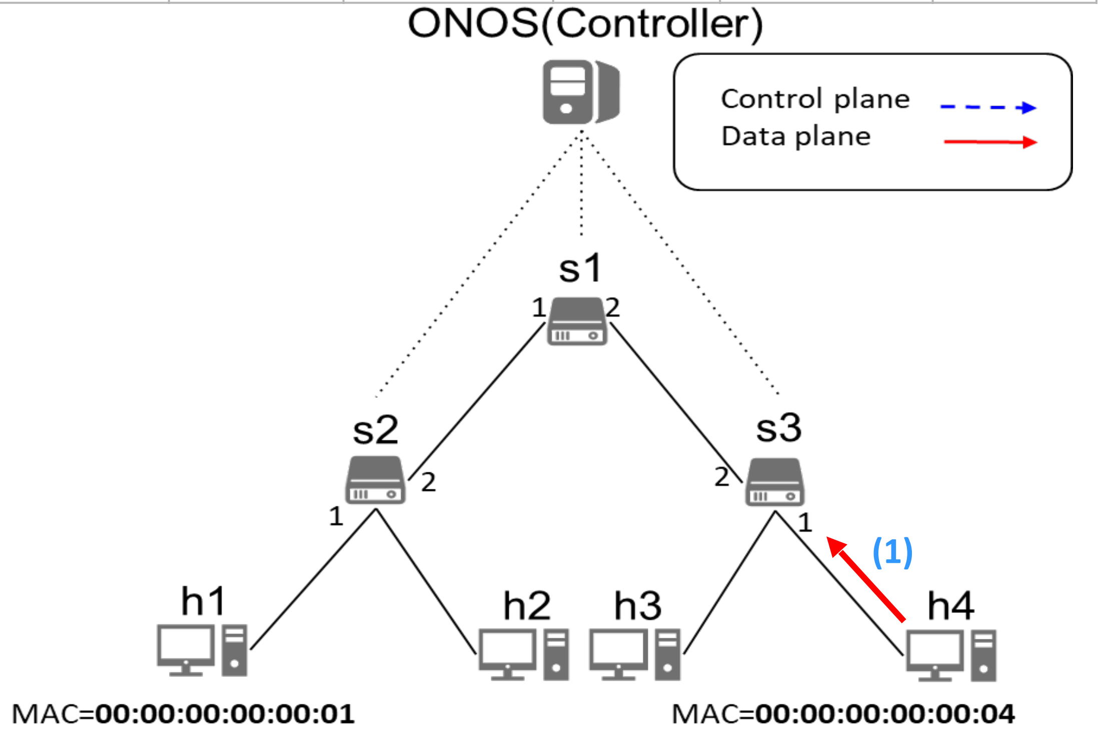
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| 00:......:01 | 1 | 00:......:01 | 1 | 00:......:01 | 2 |
| | | | | | |



# Workflow (h4 -> h1)

1. h4 replies to h1.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h1 receives packet from h4.

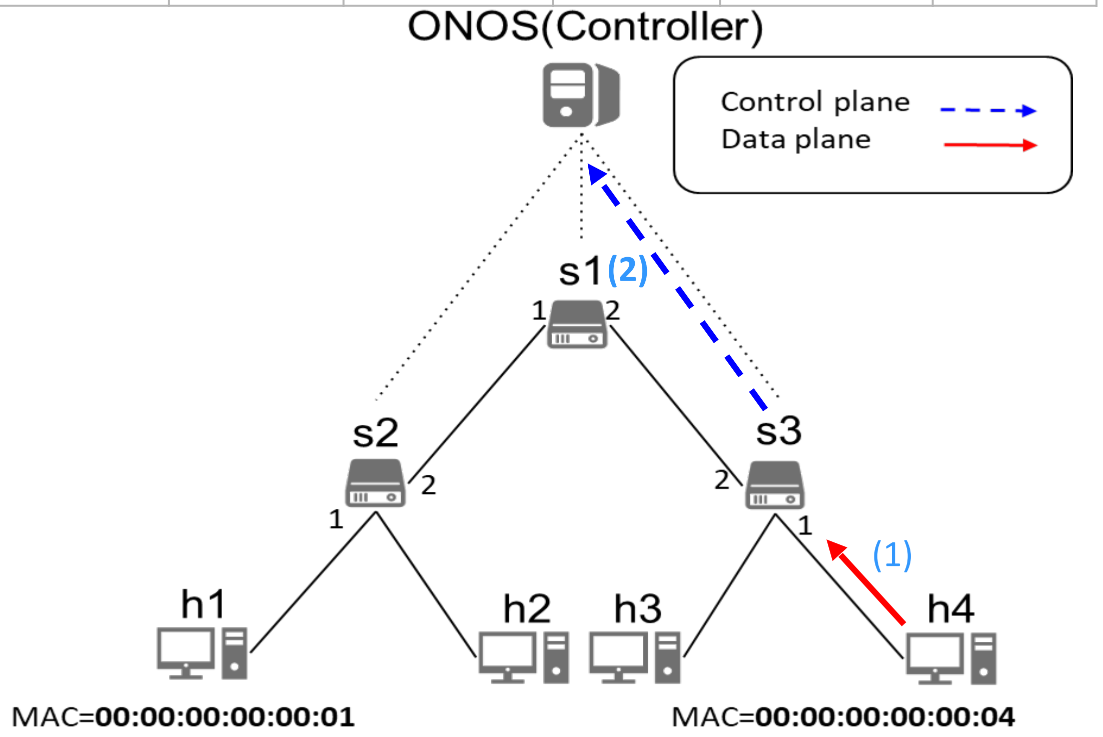
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| 00:......:01 | 1 | 00:......:01 | 1 | 00:......:01 | 2 |
| | | | | | |



## Workflow (h4 -> h1)

1. h4 replies to h1.
2. Switch (s3) sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
    a. Destination MAC not found:
        - Floods Packet-out.
    b. Destination MAC found:
        - Sends Packet-out via designated port.
        - Installs flow rule on switch.
5. h1 receives packet from h4.

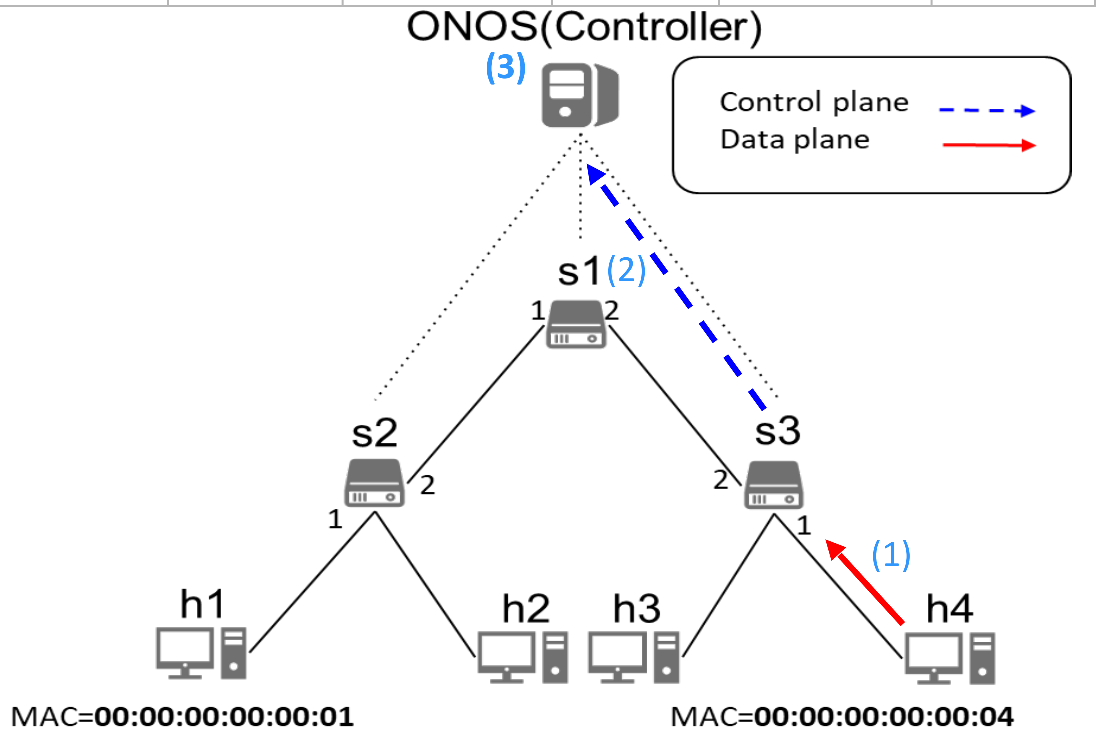
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| 00:......:01 | 1 | 00:......:01 | 1 | 00:......:01 | 2 |
| | | | | | |



# Workflow (h4 -> h1)

1. h4 replies to h1.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h1 receives packet from h4.

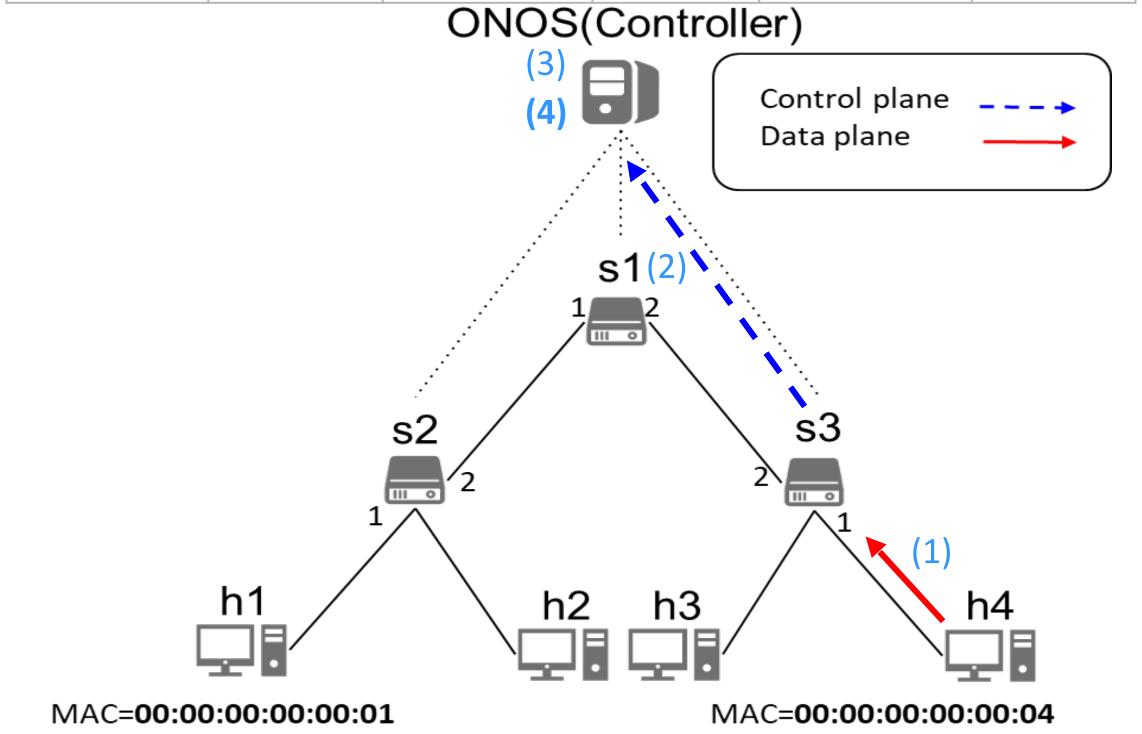
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| 00:......:01 | 1 | 00:......:01 | 1 | 00:......:01 | 2 |
| | | | | 00:......:04 | 1 |



# Workflow (h4 -> h1)

1. h4 replies to h1.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h1 receives packet from h4.

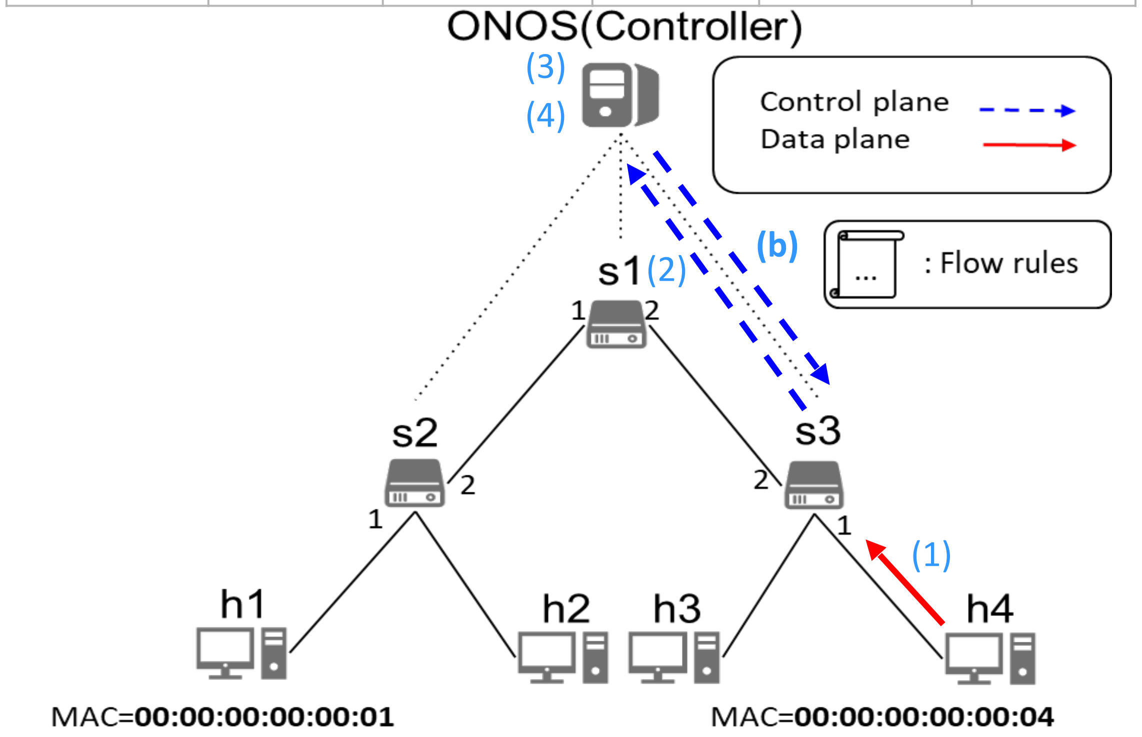
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| 00:……:01 | 1 | 00:……:01 | 1 | 00:……:01 | 2 |
| | | | | 00:……:04 | 1 |



# Workflow (h4 -> h1)

1. h4 replies to h1.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h1 receives packet from h4.

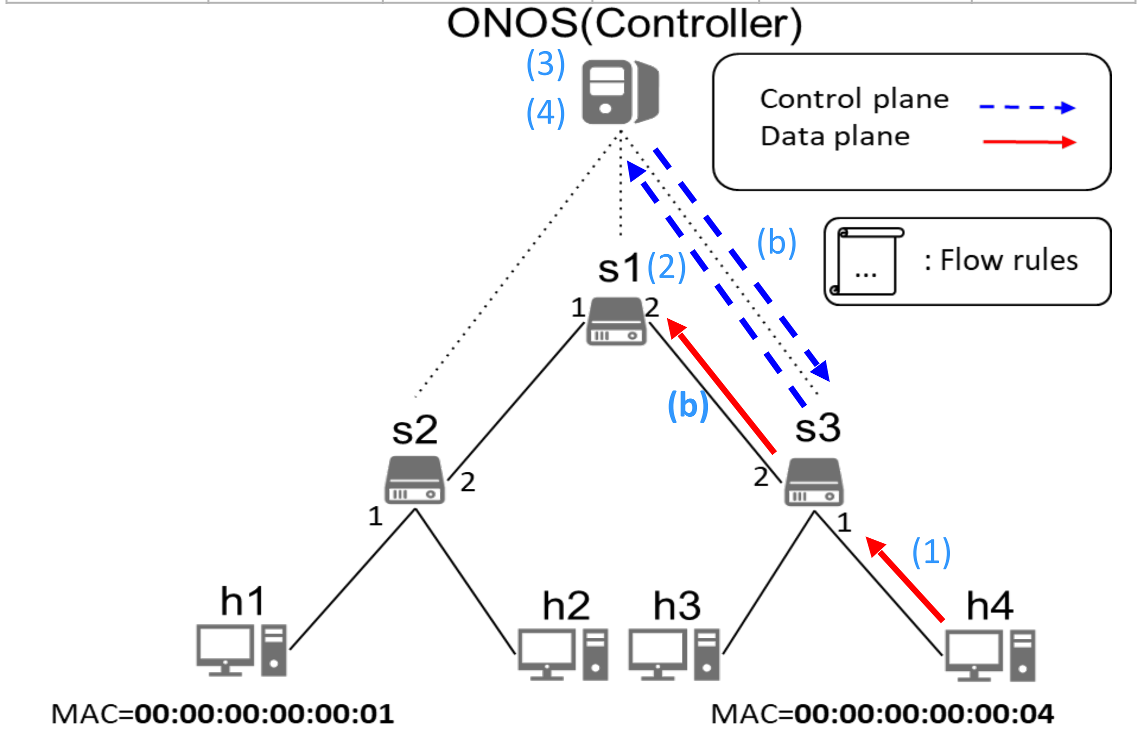
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| 00:......:01 | 1 | 00:......:01 | 1 | 00:......:01 | 2 |
| | | | | 00:......:04 | 1 |



# Workflow (h4 -> h1)

1. h4 replies to h1.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h1 receives packet from h4.

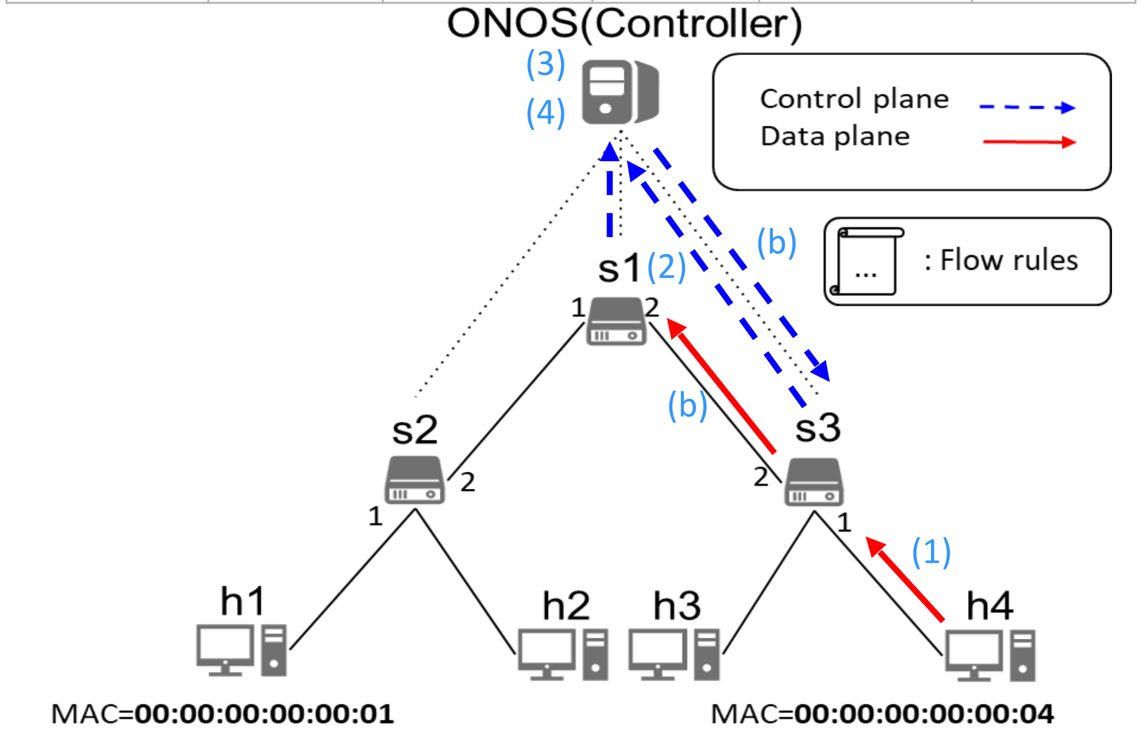
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| 00:......:01 | 1 | 00:......:01 | 1 | 00:......:01 | 2 |
| | | | | 00:......:04 | 1 |



# Workflow (h4 -> h1)

1. h4 replies to h1.
2. Switch (s1) sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h1 receives packet from h4.

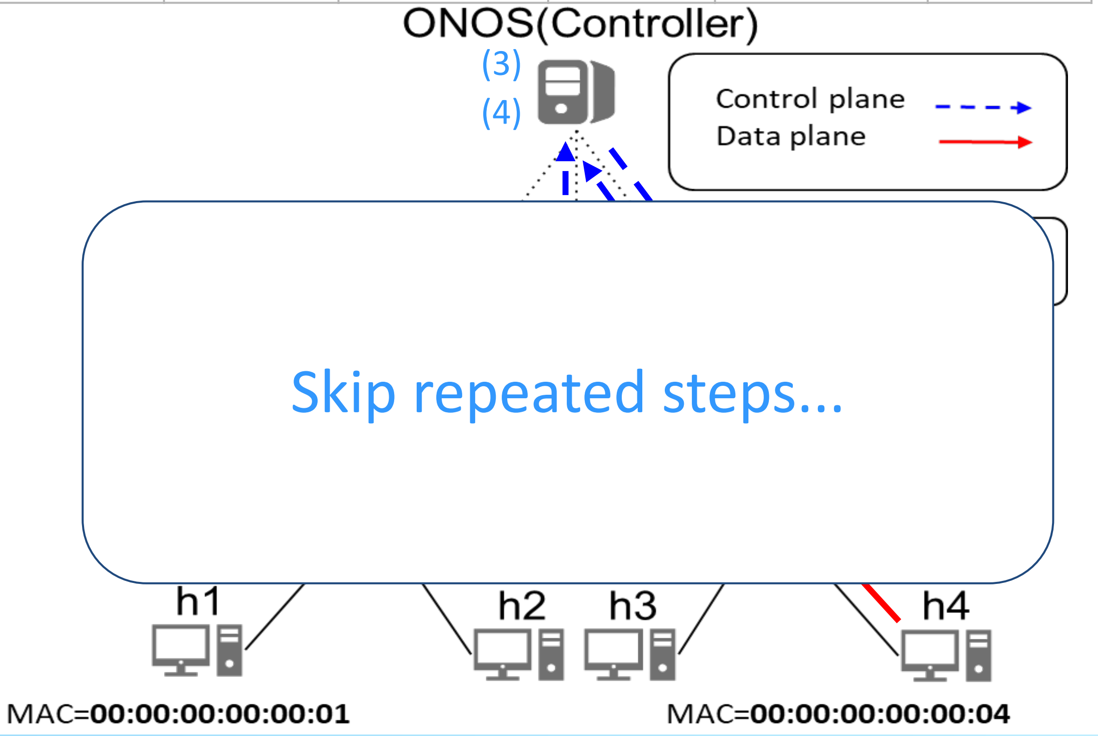
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| 00:......:01 | 1 | 00:......:01 | 1 | 00:......:01 | 2 |
| | | | | 00:......:04 | 1 |



# Workflow (h4 -> h1)

1. h4 replies to h1.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h1 receives packet from h4.

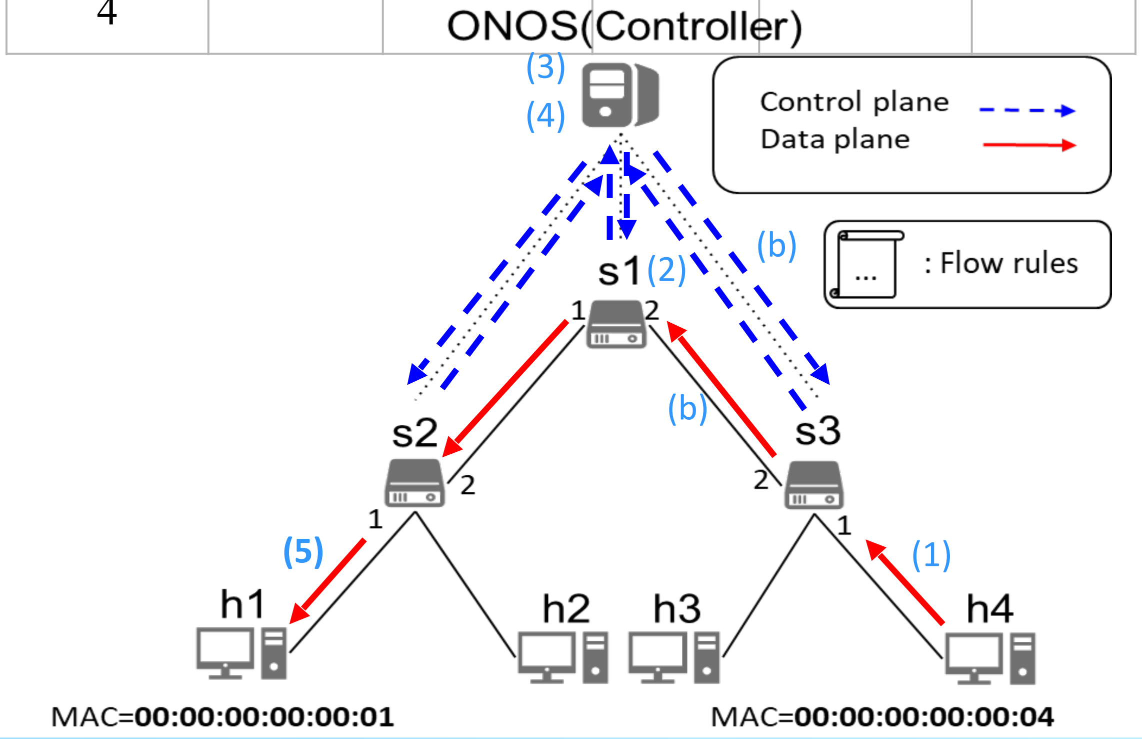
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| 00:......:01 | 1 | 00:......:01 | 1 | 00:......:01 | 2 |
| | | | | 00:......:04 | 1 |



# Workflow (h4 -> h1)

1. h4 replies to h1.
2. Switch sends Packet-in to Controller.
3. Controller updates MAC address table with source MAC and incoming port.
4. Controller looks up MAC address table for destination MAC:
   a. Destination MAC not found:
      - Floods Packet-out.
   b. Destination MAC found:
      - Sends Packet-out via designated port.
      - Installs flow rule on switch.
5. h1 receives packet from h4.

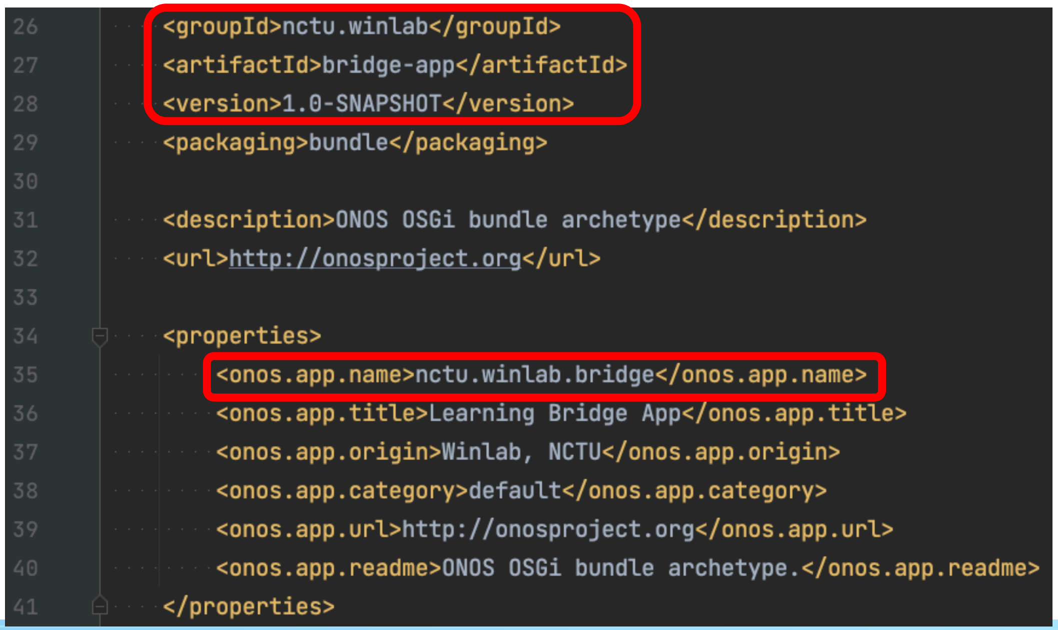
| s1 | | s2 | | s3 | |
|---|---|---|---|---|---|
| MAC | Port | MAC | Port | MAC | Port |
| 00:......:01 | 1 | 00:......:01 | 1 | 00:......:01 | 2 |
| 00:......:04 | 2 | 00:......:04 | 2 | 00:......:04 | 1 |



# Outline

# Create ONOS application

- You must set values in the pom.xml file as the following: **(10%)**
  - <groupId>: nctu.winlab
  - <artifactId>: bridge-<last 3 digits of your ID>
  - <version>: (default)
  - <onos.app.name>: nctu.winlab.bridge
- You earn credits only if all settings are correct.



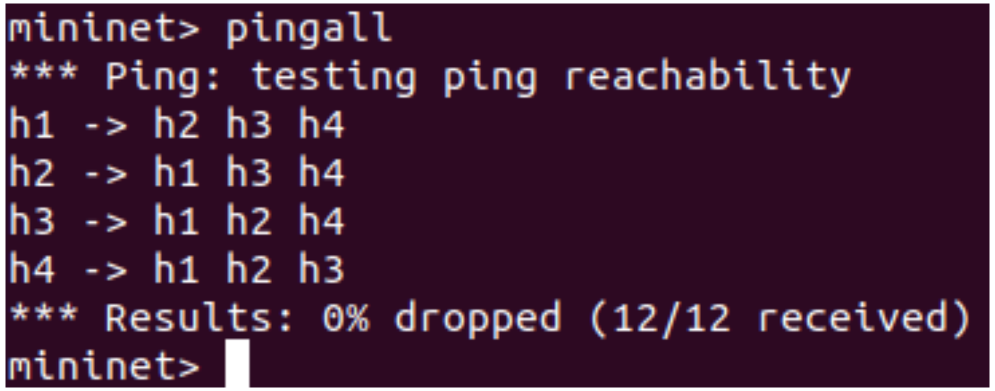
```
    <groupId>nctu.winlab</groupId>
    <artifactId>bridge-app</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>bundle</packaging>

    <description>ONOS OSGi bundle archetype</description>
    <url>http://onosproject.org</url>

    <properties>
        <onos.app.name>nctu.winlab.bridge</onos.app.name>
        <onos.app.title>Learning Bridge App</onos.app.title>
        <onos.app.origin>Winlab, NCTU</onos.app.origin>
        <onos.app.category>default</onos.app.category>
        <onos.app.url>http://onosproject.org</onos.app.url>
        <onos.app.readme>ONOS OSGi bundle archetype.</onos.app.readme>
    </properties>
```

# Outline

# Learning Bridge Function (1/4)

- Ping should work for all host pairs.

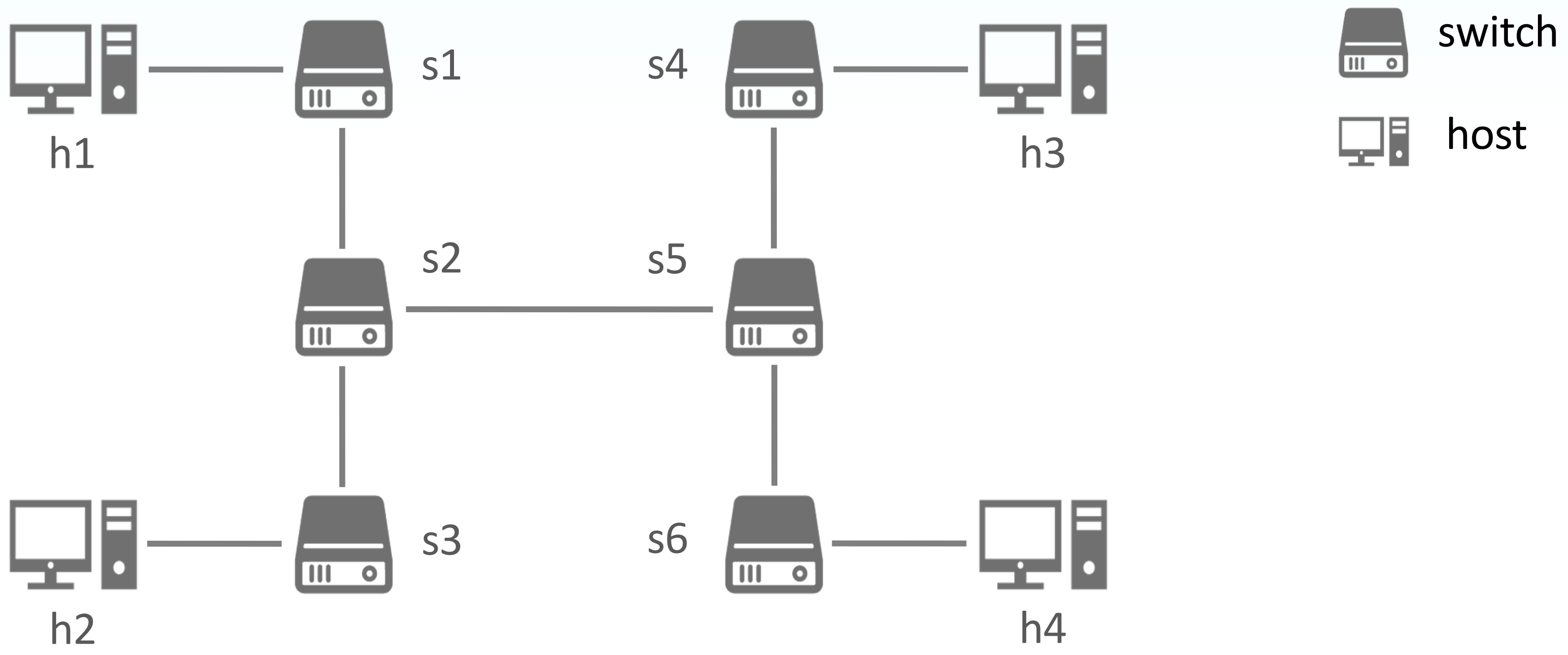
```
mininet> pingall
```



```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet>
```

## Learning Bridge Function (2/4)

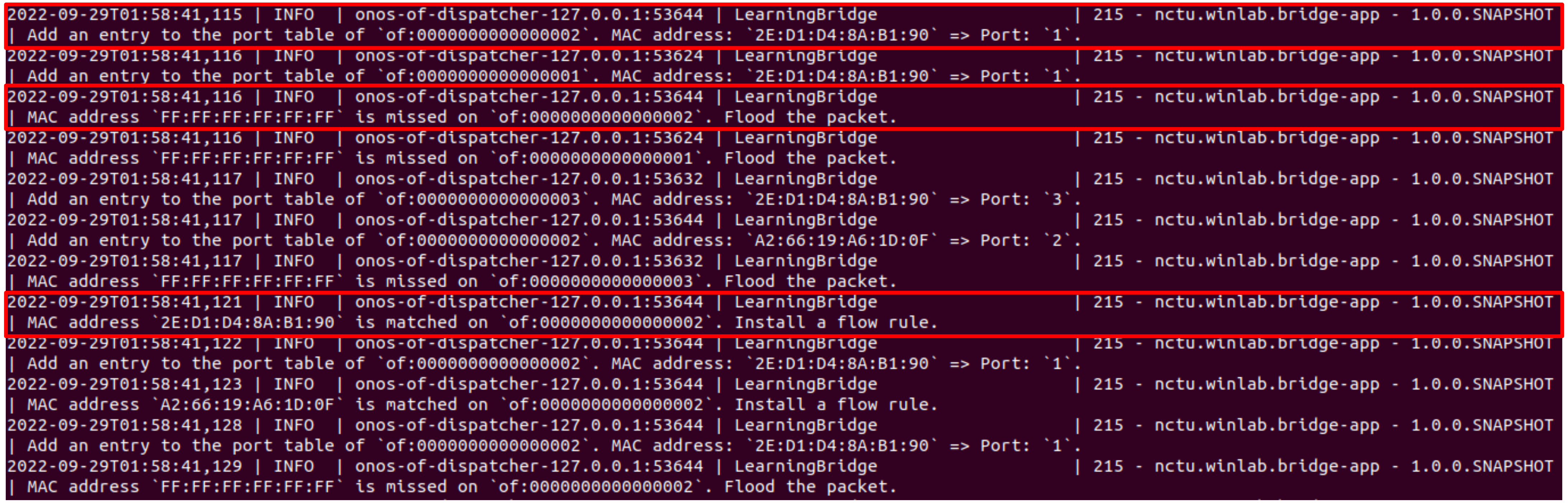
- Learning Bridge Function with the following topology. **(20%)**



- Learning Bridge Function with additional topology. **(20%)**
  - The additional topology will be announced when demo starts.

## Learning Bridge Function (3/4)

- Use ***log.info()*** to record actions done by your application.
  1. New entry is added into the forwarding table. **(6%)**
  2. Destination MAC address is missed. Flood the packet. **(7%)**
  3. Destination MAC address is matched. Install a flow rule. **(7%)**
- You earn credits only if each log pattern is exactly the same as the given one.



```
1.
2022-09-29T01:58:41,115 | INFO  | onos-of-dispatcher-127.0.0.1:53644 | LearningBridge                   | 215 - nctu.winlab.bridge-app - 1.0.0.SNAPSHOT
| Add an entry to the port table of `of:0000000000000002`. MAC address: `2E:D1:D4:8A:B1:90` => Port: `1`.
2022-09-29T01:58:41,116 | INFO  | onos-of-dispatcher-127.0.0.1:53624 | LearningBridge                   | 215 - nctu.winlab.bridge-app - 1.0.0.SNAPSHOT
| Add an entry to the port table of `of:0000000000000001`. MAC address: `2E:D1:D4:8A:B1:90` => Port: `1`.
2.
2022-09-29T01:58:41,116 | INFO  | onos-of-dispatcher-127.0.0.1:53644 | LearningBridge                   | 215 - nctu.winlab.bridge-app - 1.0.0.SNAPSHOT
| MAC address `FF:FF:FF:FF:FF:FF` is missed on `of:0000000000000002`. Flood the packet.
2022-09-29T01:58:41,116 | INFO  | onos-of-dispatcher-127.0.0.1:53624 | LearningBridge                   | 215 - nctu.winlab.bridge-app - 1.0.0.SNAPSHOT
| MAC address `FF:FF:FF:FF:FF:FF` is missed on `of:0000000000000001`. Flood the packet.
2022-09-29T01:58:41,117 | INFO  | onos-of-dispatcher-127.0.0.1:53632 | LearningBridge                   | 215 - nctu.winlab.bridge-app - 1.0.0.SNAPSHOT
| Add an entry to the port table of `of:0000000000000003`. MAC address: `2E:D1:D4:8A:B1:90` => Port: `3`.
2022-09-29T01:58:41,117 | INFO  | onos-of-dispatcher-127.0.0.1:53644 | LearningBridge                   | 215 - nctu.winlab.bridge-app - 1.0.0.SNAPSHOT
| Add an entry to the port table of `of:0000000000000002`. MAC address: `A2:66:19:A6:1D:0F` => Port: `2`.
2022-09-29T01:58:41,117 | INFO  | onos-of-dispatcher-127.0.0.1:53632 | LearningBridge                   | 215 - nctu.winlab.bridge-app - 1.0.0.SNAPSHOT
| MAC address `FF:FF:FF:FF:FF:FF` is missed on `of:0000000000000003`. Flood the packet.
3.
2022-09-29T01:58:41,121 | INFO  | onos-of-dispatcher-127.0.0.1:53644 | LearningBridge                   | 215 - nctu.winlab.bridge-app - 1.0.0.SNAPSHOT
| MAC address `2E:D1:D4:8A:B1:90` is matched on `of:0000000000000002`. Install a flow rule.
2022-09-29T01:58:41,122 | INFO  | onos-of-dispatcher-127.0.0.1:53644 | LearningBridge                   | 215 - nctu.winlab.bridge-app - 1.0.0.SNAPSHOT
| Add an entry to the port table of `of:0000000000000002`. MAC address: `2E:D1:D4:8A:B1:90` => Port: `1`.
2022-09-29T01:58:41,123 | INFO  | onos-of-dispatcher-127.0.0.1:53644 | LearningBridge                   | 215 - nctu.winlab.bridge-app - 1.0.0.SNAPSHOT
| MAC address `A2:66:19:A6:1D:0F` is matched on `of:0000000000000002`. Install a flow rule.
2022-09-29T01:58:41,128 | INFO  | onos-of-dispatcher-127.0.0.1:53644 | LearningBridge                   | 215 - nctu.winlab.bridge-app - 1.0.0.SNAPSHOT
| Add an entry to the port table of `of:0000000000000002`. MAC address: `2E:D1:D4:8A:B1:90` => Port: `1`.
2022-09-29T01:58:41,129 | INFO  | onos-of-dispatcher-127.0.0.1:53644 | LearningBridge                   | 215 - nctu.winlab.bridge-app - 1.0.0.SNAPSHOT
| MAC address `FF:FF:FF:FF:FF:FF` is missed on `of:0000000000000002`. Flood the packet.
```

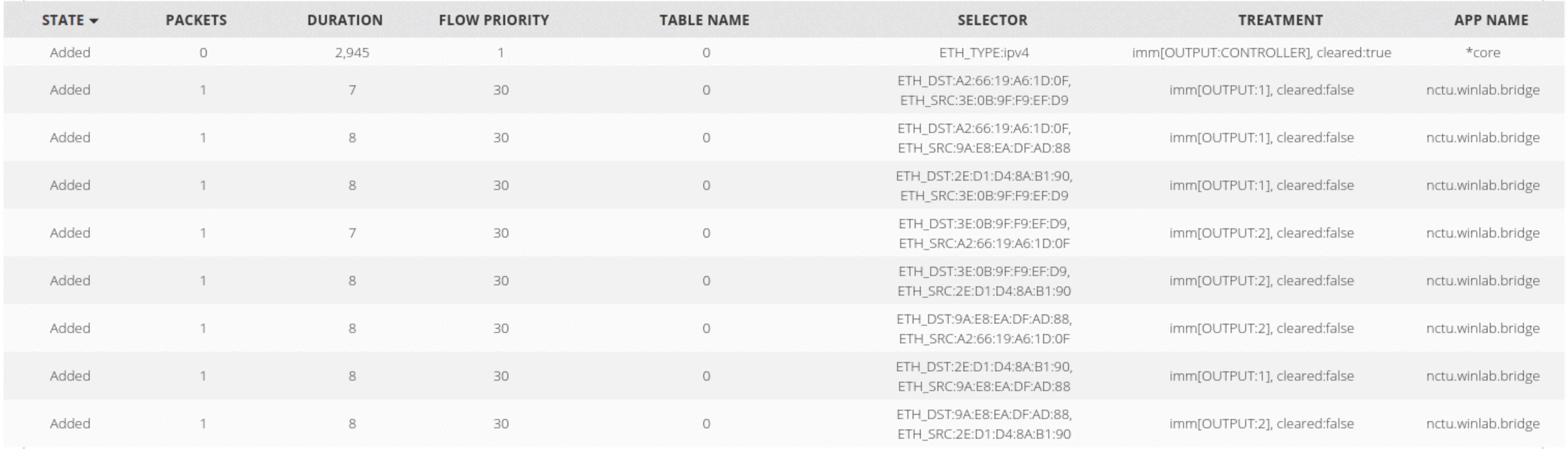
## Learning Bridge Function (4/4)

1. New entry is added into the MAC address table.
   - Pattern: “**Add an entry to the port table of \`{device ID}\`. MAC address: \`{MAC}\` => Port: \`{port}\`.**”
   - Example: “Add an entry to the port table of \`of:0000000000000002\`. MAC address: \`2E:D1:D4:8A:B1:90\` => Port: \`1\`.”
2. Destination MAC address is missed. Flood the packet.
   - Pattern: ”**MAC address \`{MAC}\` is missed on \`{device ID}\`. Flood the packet.**”
   - Example: “MAC address \`FF:FF:FF:FF:FF:FF\` is missed on \`of:0000000000000002\`. Flood the packet.”
3. Destination MAC address is matched. Install a flow rule.
   - Pattern: ”**MAC address \`{MAC}\` is matched on \`{device ID}\`. Install a flow rule.**”
   - Example: “MAC address \`2E:D1:D4:8A:B1:90\` is matched on \`of:0000000000000002\`. Install a flow rule.”

## Outline

# Flow Rule Regulation

- Rule requirements:
  - Match field (selector): ETH_SRC, ETH_DST **(5%)**
  - Action field (treatment): OUTPUT **(5%)**
  - Flow priority: 30 **(5%)**
  - Flow timeout: 30 **(5%)**



| STATE | PACKETS | DURATION | FLOW PRIORITY | TABLE NAME | SELECTOR | TREATMENT | APP NAME |
|---|---|---|---|---|---|---|---|
| Added | 0 | 2,945 | 1 | 0 | ETH_TYPE:ipv4 | imm[OUTPUT:CONTROLLER], cleared:true | *core |
| Added | 1 | 7 | 30 | 0 | ETH_DST:A2:66:19:A6:1D:0F, ETH_SRC:3E:0B:9F:F9:EF:D9 | imm[OUTPUT:1], cleared:false | nctu.winlab.bridge |
| Added | 1 | 8 | 30 | 0 | ETH_DST:A2:66:19:A6:1D:0F, ETH_SRC:9A:E8:EA:DF:AD:88 | imm[OUTPUT:1], cleared:false | nctu.winlab.bridge |
| Added | 1 | 8 | 30 | 0 | ETH_DST:2E:D1:D4:8A:B1:90, ETH_SRC:3E:0B:9F:F9:EF:D9 | imm[OUTPUT:1], cleared:false | nctu.winlab.bridge |
| Added | 1 | 7 | 30 | 0 | ETH_DST:3E:0B:9F:F9:EF:D9, ETH_SRC:A2:66:19:A6:1D:0F | imm[OUTPUT:2], cleared:false | nctu.winlab.bridge |
| Added | 1 | 8 | 30 | 0 | ETH_DST:3E:0B:9F:F9:EF:D9, ETH_SRC:2E:D1:D4:8A:B1:90 | imm[OUTPUT:2], cleared:false | nctu.winlab.bridge |
| Added | 1 | 8 | 30 | 0 | ETH_DST:9A:E8:EA:DF:AD:88, ETH_SRC:A2:66:19:A6:1D:0F | imm[OUTPUT:2], cleared:false | nctu.winlab.bridge |
| Added | 1 | 8 | 30 | 0 | ETH_DST:2E:D1:D4:8A:B1:90, ETH_SRC:9A:E8:EA:DF:AD:88 | imm[OUTPUT:1], cleared:false | nctu.winlab.bridge |
| Added | 1 | 8 | 30 | 0 | ETH_DST:9A:E8:EA:DF:AD:88, ETH_SRC:2E:D1:D4:8A:B1:90 | imm[OUTPUT:2], cleared:false | nctu.winlab.bridge |

# Outline

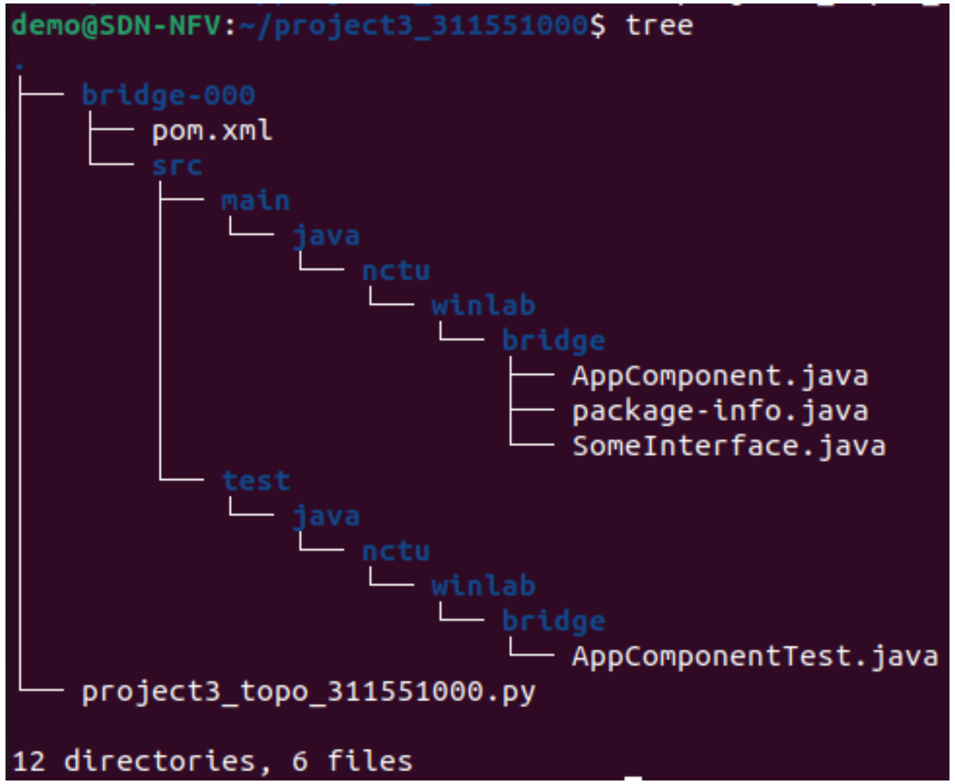
# Submission Naming Convention (1/2)

- Naming conventions in your python script
  - Name of python script: project3_topo_<studentID>.py
  - Name of topology class: Project3_Topo_<studentID>
  - Name of dictionary's key: topo_<studentID>
- Command to execute your script

```
 $ sudo mn --custom=project3_topo_<studentID>.py \
   --topo=topo_<studentID> \
   --controller=remote,127.0.0.1:6653 \
   --switch=ovs,protocols=OpenFlow14
```

# Submission Naming Convention (2/2)

- Move your bridge-app and the python script into directory **project3_<student ID>**.



```
demo@SDN-NFV:~/project3_311551000$ tree
.
├── bridge-000
│   ├── pom.xml
│   └── src
│       ├── main
│       │   └── java
│       │       └── nctu
│       │           └── winlab
│       │               └── bridge
│       │                   ├── AppComponent.java
│       │                   ├── package-info.java
│       │                   └── SomeInterface.java
│       └── test
│           └── java
│               └── nctu
│                   └── winlab
│                       └── bridge
│                           └── AppComponentTest.java
└── project3_topo_311551000.py

12 directories, 6 files
```

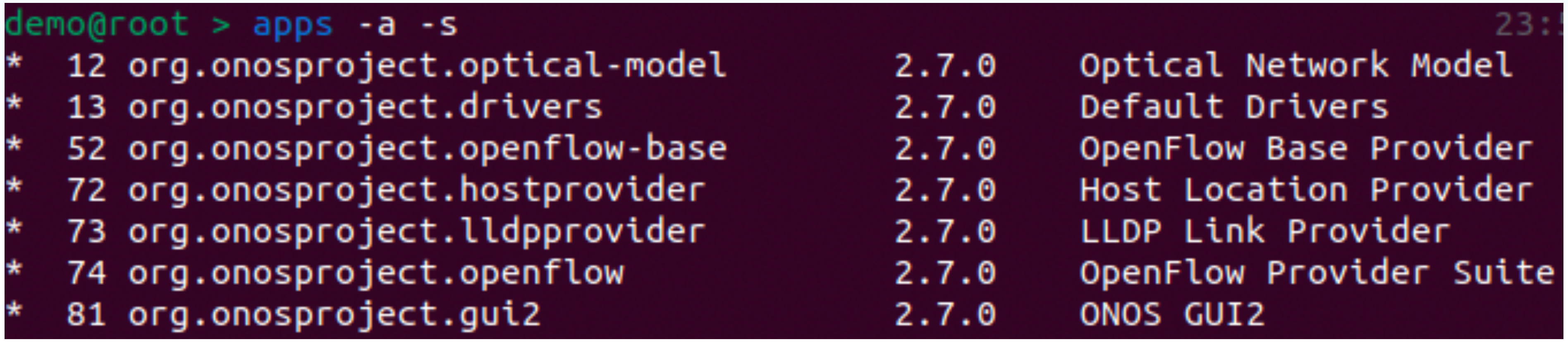
- Compress the directory into a **zip** file named as **project3_<student ID>.zip**.
- Upload your zip file to [E3](E3).
- You earn credits only if your submission follows above rules. **(10%)**

# Outline

- Overview
- Build ONOS Application Project
  - Environment Setup
  - Create an ONOS Application
  - Build, Install, and Activate ONOS Application
  - Reinstall ONOS Application
- Learning Bridge Function
  - Introduction
  - Workflow
- Project 3 Requirements
  - Create ONOS Application (10%)
  - Learning Bridge Function (60%)
  - Flow Rule Regulation (20%)
  - Submission Naming Convention (10%)
  - Restrictions

# Restrictions

- We will test your application with only the following applications activated:



```
demo@root > apps -a -s                                                  23:
*  12 org.onosproject.optical-model        2.7.0    Optical Network Model
*  13 org.onosproject.drivers              2.7.0    Default Drivers
*  52 org.onosproject.openflow-base        2.7.0    OpenFlow Base Provider
*  72 org.onosproject.hostprovider         2.7.0    Host Location Provider
*  73 org.onosproject.lldpprovider         2.7.0    LLDP Link Provider
*  74 org.onosproject.openflow             2.7.0    OpenFlow Provider Suite
*  81 org.onosproject.gui2                 2.7.0    ONOS GUI2
```

- You must only use classes under **org.onosproject.net.flowobjective** or **org.onosproject.net.flow** package to install flow rules on network devices.
  - Otherwise, subject to deduct 40% total credits.

## Hints

- You can trace ReactiveForwarding.java to figure out how to install flow rules.
- When receives Packet-in, your application need to send Packet-out to switch, in addition to installing flow rule.
- How to debug:
  - Use Logger to print runtime information.
  - Use Wireshark to capture your packets.

## Lab 3 Demo

- Date: TA will open a demo time-reserved table one week before demo. The demo dates will be in the week after Lab 3 deadline.
- Demo questions will show when demo starts.
- The demo score will be **40%** of total score.
  - e.g. If your earn 100% credits for submission and 80% credits for demo, then your total score of Lab3 will be **100 x 60% + 80 x 40% = 92**.

## About help!

- For lab problem, ask at e3 forum
  - Ask at the e3 forum
  - TAs will help to clarify Lab contents instead of giving answers!
  - Please describe your questions with sufficient context,
    - e.g. Environment setup, Input/Output, Screenshots, …
- For personal problem mail to **sdnta@win.cs.nctu.edu.tw**
  - You have special problem and you can’t meet the deadline
  - You got weird score with project
- No Fixed TA hour