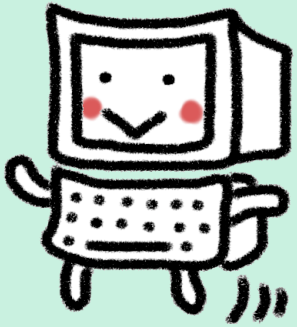


Project 4

Unicast DHCP Application

Deadline: 2020/11/04 (WED) 23:55

Outline

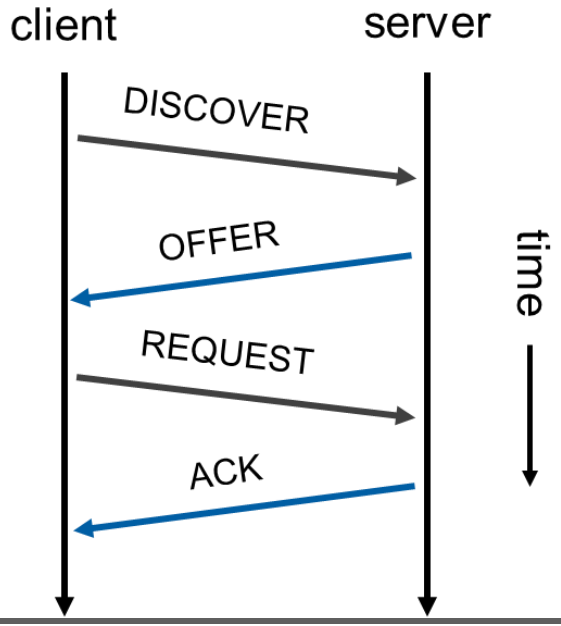
- Introduction to DHCP
  - What is DHCP?
  - DHCP Workflow
- Project 4
  - Overview
  - Workflow
  - Project Requirement
  - Supplements
- Upload Configuration for ONOS APPs
- How to Test Your DHCP APP
- Submission

Outline

- Introduction to DHCP
  - What is DHCP?
  - DHCP Workflow
- Project 4
  - Overview
  - Workflow
  - Project Requirement
  - Supplements
- Upload Configuration for ONOS APPs
- How to Test Your Unicast DHCP APP
- Submission

What is DHCP? (Dynamic Host Configuration Protocol)

- Provide necessary information for a host to access network
  - IP address, gateway, DNS (Domain Name Server), etc.
- Client and server use UDP port 68 and 67, respectively
- A DHCP transaction is completed by 4 messages:



Outline

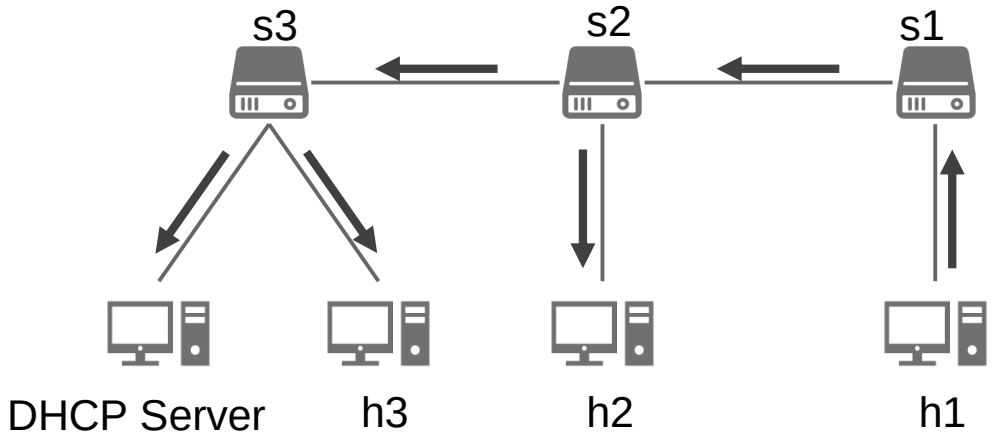
- Introduction to DHCP
  - What is DHCP?
  - DHCP Workflow
- Project 4
  - Overview
  - Workflow
  - Project Requirement
  - Supplements
- Upload Configuration for ONOS APPs
- How to Test Your Unicast DHCP APP
- Submission

Workflow (DHCP Discover)

- h1 attaches to network
  - Issue DHCPDISCOVER to locate available DHCP server (broadcast)
- A DHCP server receives DHCPDISCOVER
  - Reply DHCPOFFER (unicast or broadcast)
- h1 chooses a server to reply DHCPREQUEST (broadcast)
- The server replies with DHCPACK (unicast)
  - h1 now owns the assigned IP address

```
Src IP:  0.0.0.0
Dst IP:  255.255.255.255
Src MAC: <MAC of h1>
Dst MAC: ff:ff:ff:ff:ff:ff
```

DHCP DISCOVER

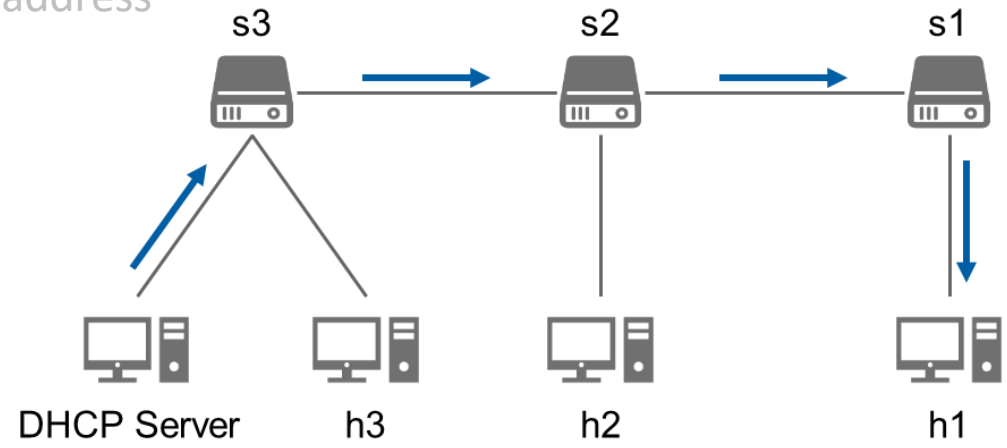


Workflow (DHCP Offer)

- h1 attaches to network
  - Issue DHCPDISCOVER to locate available DHCP server (broadcast)
- A DHCP server receives DHCPDISCOVER
  - Reply DHCPOFER (unicast or broadcast)
- h1 chooses a server to reply DHCPREQUEST (broadcast)
- The server replies with DHCPACK (unicast)
  - h1 now owns the assigned IP address

```
Src IP:  <IP of server>
Dst IP:  <IP of h1>
Src MAC: <MAC of server>
Dst MAC: <MAC of h1>
Your IP address: 10.0.0.2
Subnet Mask: 255.255.255.0
IP Address Lease Time: 3600
```

DHCP OFFER

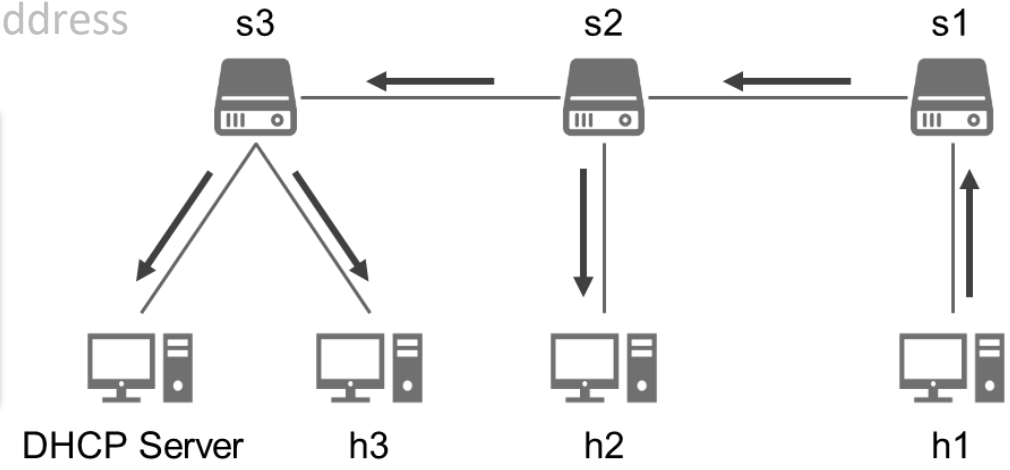


Workflow (DHCP Request)

- h1 attaches to network
  - Issue DHCPDISCOVER to locate available DHCP server (broadcast)
- A DHCP server receives DHCPDISCOVER
  - Reply DHCPOFER (unicast or broadcast)
- h1 chooses a server to reply DHCPREQUEST (broadcast)
- The server replies with DHCPACK (unicast)
  - h1 now owns the assigned IP address

```
Src IP:  0.0.0.0
Dst IP:  255.255.255.255
Src MAC: <MAC of h1>
Dst MAC: ff:ff:ff:ff:ff:ff
Requested IP address: 10.0.0.2
DHCP Server Identifier: <server IP>
```

DHCP REQUEST

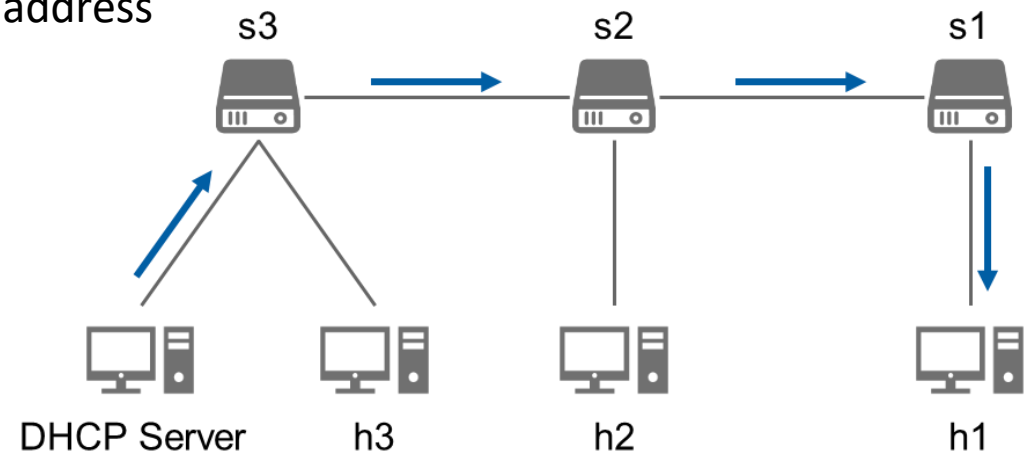


Workflow (DHCP Ack)

- h1 attaches to network
  - Issue DHCPDISCOVER to locate available DHCP server (broadcast)
- A DHCP server receives DHCPDISCOVER
  - Reply DHCPOFER (unicast or broadcast)
- h1 chooses a server to reply DHCPREQUEST (broadcast)
- The server replies with DHCPACK (unicast)
  - h1 now owns the assigned IP address

```
Src IP:  <IP of server>
Dst IP:  <IP of h1>
Src MAC: <MAC of server>
Dst MAC: <MAC of h1>
Your IP address: 10.0.0.2
Subnet Mask: 255.255.255.0
IP Address Lease Time: 3600
```

DHCP ACK



Outline

- Introduction to DHCP
  - What is DHCP?
  - DHCP Workflow
- Project 4
  - Overview
  - Workflow
  - Project Requirement
  - Supplements
- Upload Configuration for ONOS APPs
- How to Test Your Unicast DHCP APP
- Submission

Overview

- Originally, there are many broadcast packets in the network
- In this project, you need to implement an unicast DHCP application
  1. Configure a DHCP server location
  2. Install flow rules to Packet-in DHCP packets
  3. Compute path between a DHCP client and the DHCP application
  4. Install flow rules to forward DHCP packets via unicast

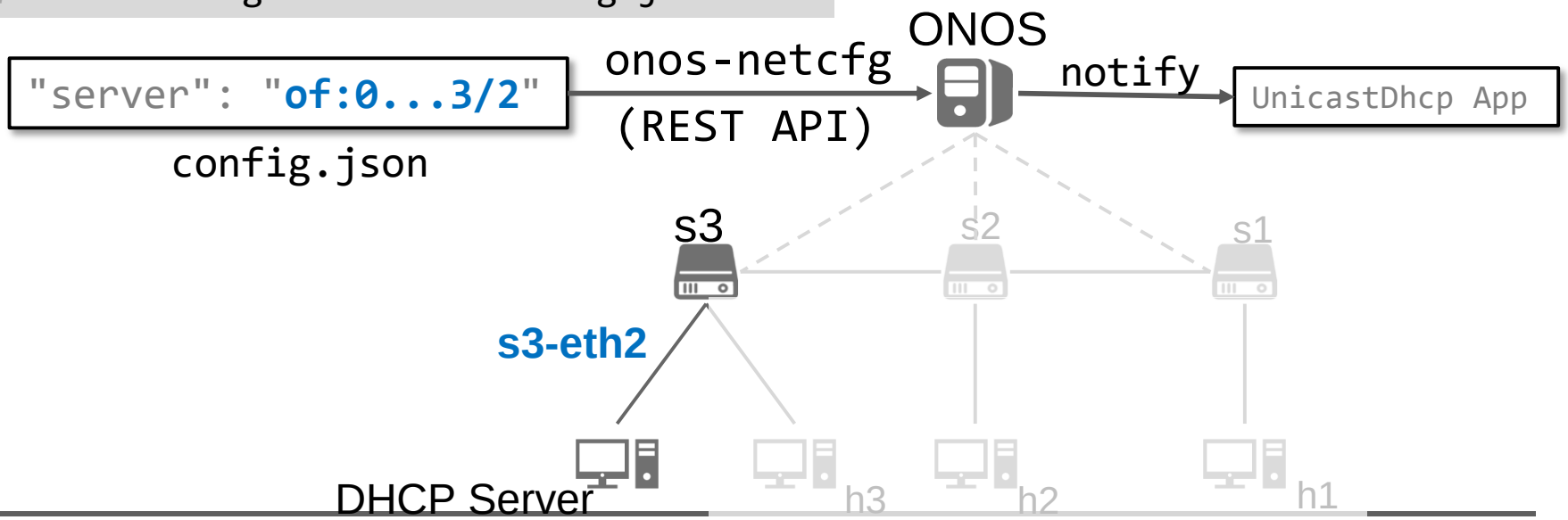
Outline

- Introduction to DHCP
  - What is DHCP?
  - DHCP Workflow
- **Project 4**
  - Overview
  - **Workflow**
  - Project Requirement
  - Supplements
- How to Provide Configuration for ONOS APPs
- How to Test Your Unicast DHCP APP
- Submission

Step 1 – Configure DHCP Server Location

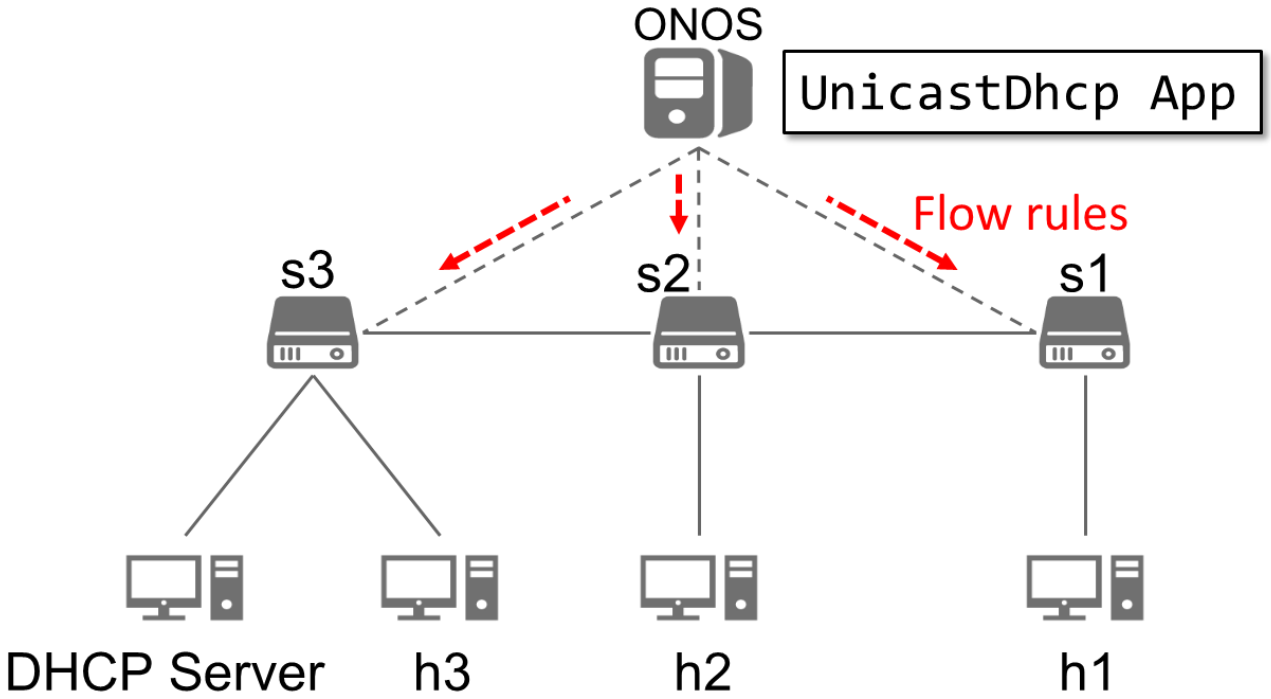
- Describing the *ConnectPoint* of DHCP server
  - *config.json*
- Upload the file to ONOS configuration service via REST API
- Should print configured location to ONOS log when notified

```
bash$ onos-netcfg localhost config.json
```



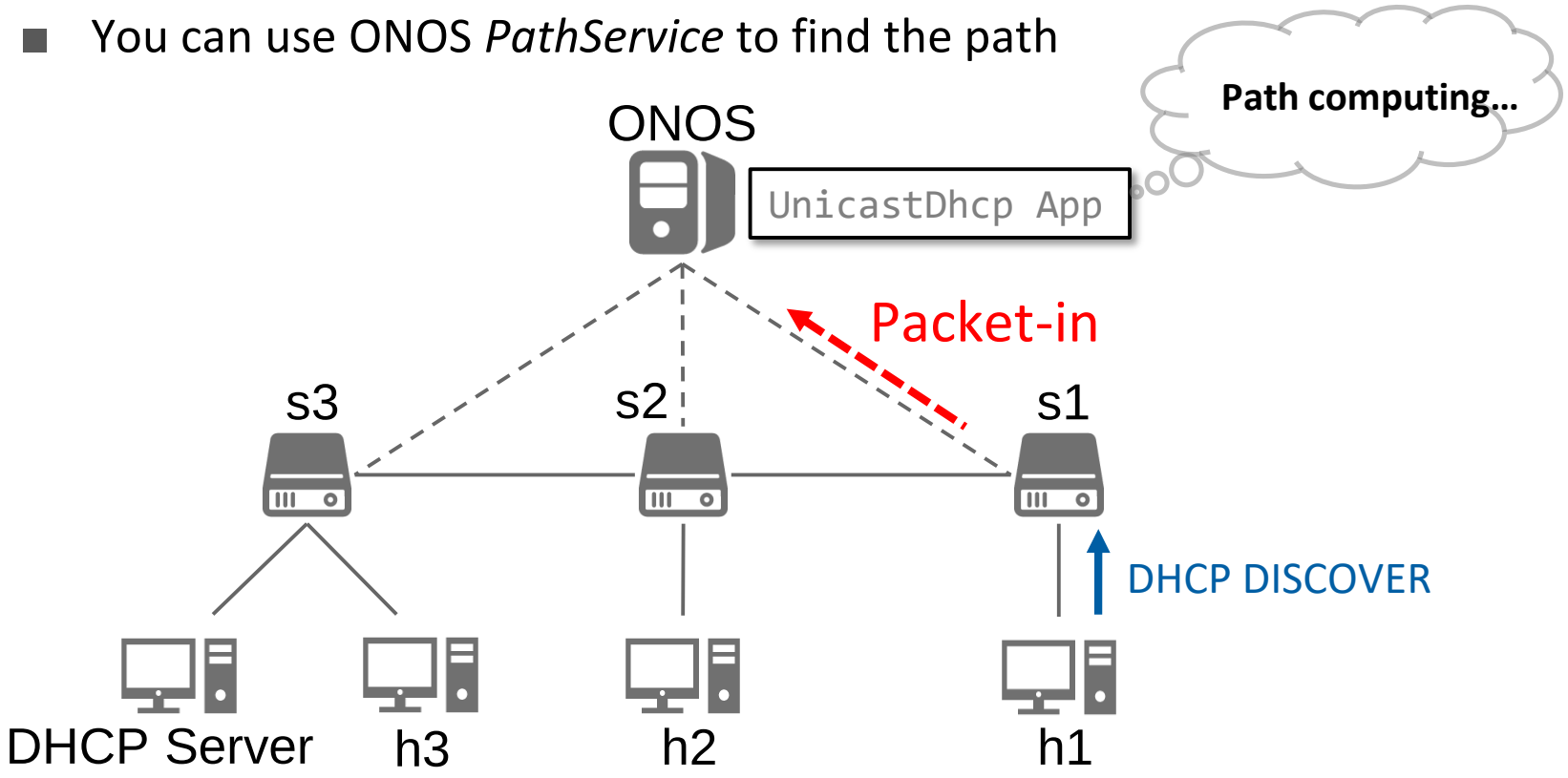
Step 2 – Packet-In DHCP Packets

- Request switches to Packet-in DHCP packets



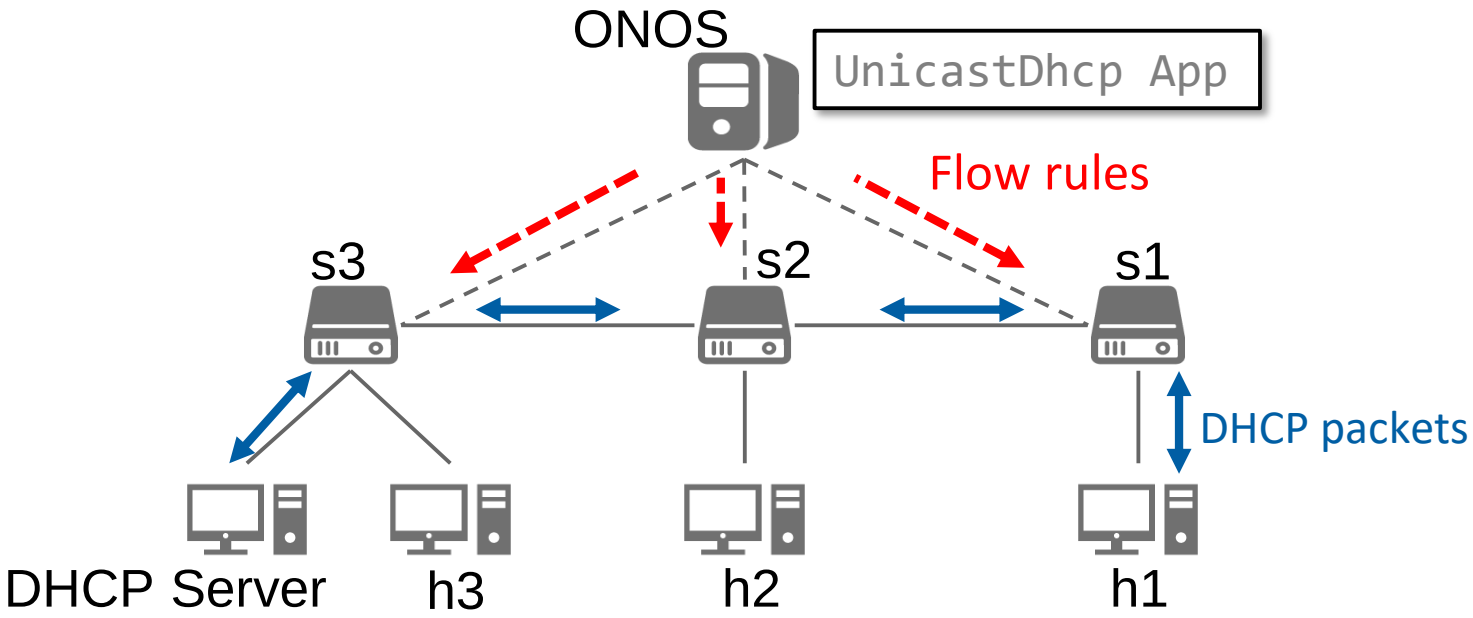
Step 3 – Compute Path

- Compute the path between requester and DHCP server
  - You can use ONOS *PathService* to find the path



Step 4 – Forward DHCP Packets via Unicast

- Install flow rules to forward DHCP packets
- Subsequent DHCP packets should all become unicast
  - DISCOVER, OFFER, REQUEST, ACK
- Interfaces not on the path should not receive any DHCP packet



Outline

- Introduction to DHCP
  - What is DHCP?
  - DHCP Workflow
- **Project 4**
  - Overview
  - Workflow
  - **Project Requirement**
  - Supplements
- How to Provide Configuration for ONOS APPs
- How to Test Your Unicast DHCP APP
- Submission

Naming Requirement

- You should follow the Maven project naming format below, or your project will not be scored
  - <groupId>: nctu.winlab
  - <artifactId>: unicastdhcp
  - <version>: <use default> (1.0-SNAPSHOT)
  - <Package>: nctu.winlab.unicastdhcp

Project 4 Requirements and Scoring Criteria

- ❑ **(10%)** Project naming convention
- ❑ **(30%)** Print DHCP location in ONOS log after uploading config file

```
| 190 - org.onosproject.onos-core-net - 2.2.0 | Application winlab.nctu.unicastdhcp has b

| 209 - winlab.nctu.unicastdhcp - 1.0.0.SNAPSHOT | DHCP server is at of:0000000000000003/2
```

- ❑ **(30%)** Host(s) can get IP address after using dhclient
- ❑ **(30%)** DHCP transaction packets should be forwarded by **unicast**

Suggestion

- In this project, it is not required to use ping to check connectivity.
  - For simplicity, you should deactivate **fwd** application
  - We will deactivate fwd when testing your App

```
brian@root > apps -a -s
*   6 org.onosproject.drivers              2.2.0    Default Drivers
*   7 org.onosproject.optical-model        2.2.0    Optical Network Model
*  39 org.onosproject.gui2                 2.2.0    ONOS GUI2
*  52 org.onosproject.openflow-base        2.2.0    OpenFlow Base Provider
*  84 org.onosproject.hostprovider         2.2.0    Host Location Provider
*  85 org.onosproject.lldpprovider         2.2.0    LLDP Link Provider
*  86 org.onosproject.openflow             2.2.0    OpenFlow Provider Suite
* 192 winlab.nctu.unicastdhcp              1.0.SNAPSHOT ONOS OSGi bundle archetype
```

Outline

- Introduction to DHCP
  - What is DHCP?
  - DHCP Workflow
- **Project 4**
  - Overview
  - Workflow
  - Project Requirement
  - **Supplements**
- Upload Configuration for ONOS APPs
- How to Test Your DHCP APP
- Submission

Supplements

- "project4-supplement.zip" includes following files:
  1. Program and configuration files of a sample application **echoconfig**
     1) **AppComponent.java**:
        - Main program of **echoconfig** app
     2) **NameConfig.java**
        - validates and retrieves configuration data from config.json
     3) **config.json**:
        - configuration file for **echoconfig** app
  2. **Network Topology files for Unicast DHCP App**
     1) **topo.py**: mininet topology
     2) **dhcpd.conf**: DHCP configuration used by **topo.py**
  3. **Configuration file for Unicast DHCP App**
     - **unicastdhcp.json**: configuration file for unicast DHCP app

Outline

- Introduction to DHCP
  - What is DHCP?
  - DHCP Workflow
- Project 4
  - Overview
  - Workflow
  - Project Requirement
  - Supplements
- **Upload Configuration for ONOS APPs**
- How to Test Your DHCP APP
- Submission

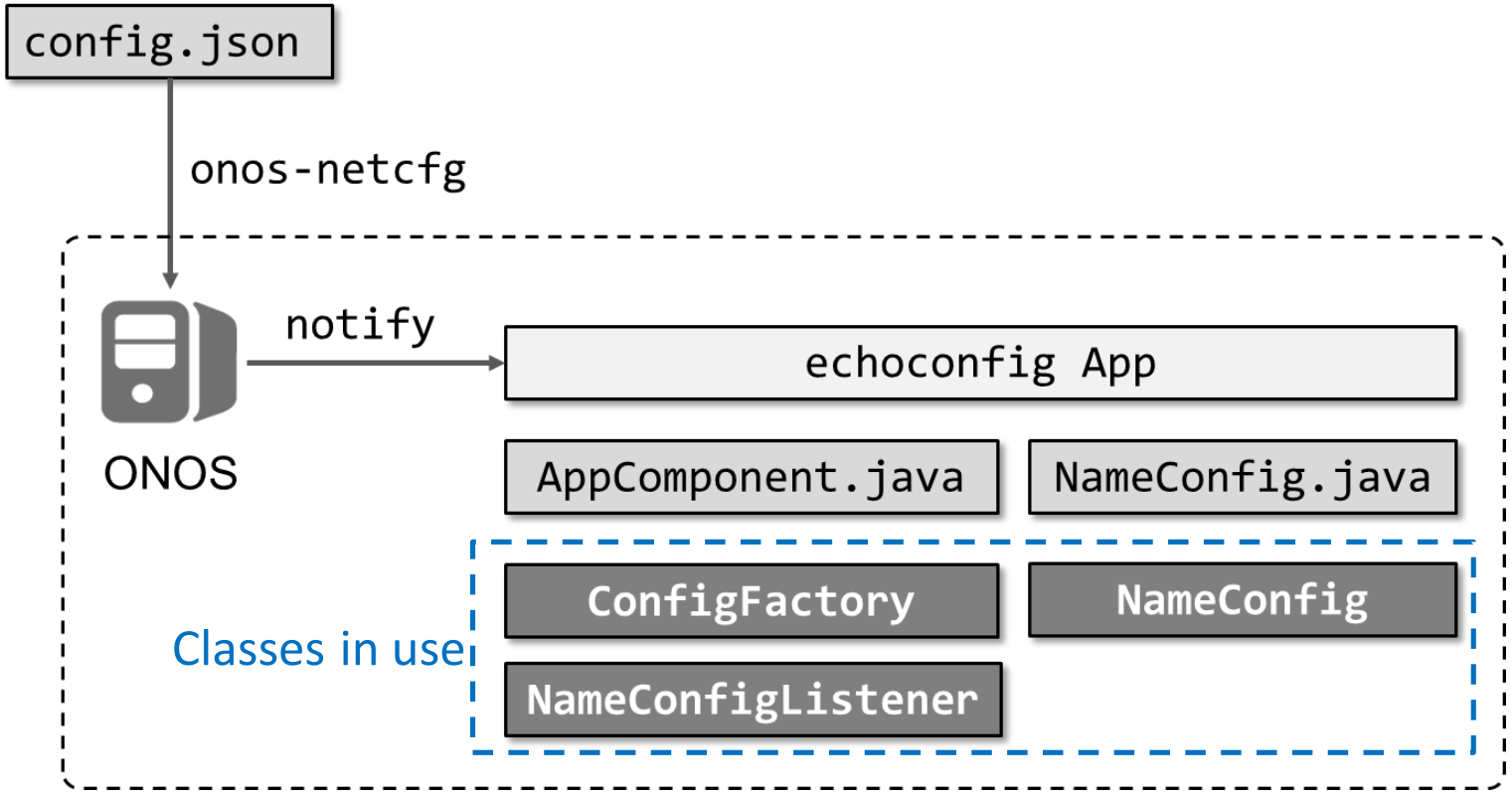
Supplements

- "project4-supplement.zip" includes following files:
  1. Program and configuration files of a sample application **echoconfig**
     1) **AppComponent.java**:
        - Main program of **echoconfig**
     2) **NameConfig.java**:
        - validates and retrieves configuration data from config.json
     3) **config.json**:
        - configuration file for **echoconfig**
  2. **Network Topology files for Unicast DHCP App**
     1) **topo.py**: mininet topology
     2) **dhcpd.conf**: DHCP configuration used by **topo.py**
  3. **Network Configuration file for Unicast DHCP App**
     - **unicastdhcp.json**: configuration file for unicast DHCP app

An Example Application – echoconfig

- ***Echoconfig*** App :
  echoes (prints out) a name specified in the configuration file
- Components of ***echoconfig***
  1. ***AppComponent.java***: main program of ***echoconfig*** that
     - Listen to ***configuration file uploaded*** event
     - Instantiate a ***NameConfig*** object
     - Prints ***value of name*** specified in configuration file
  2. ***NameConfig.java***
     - Provide functions to validate and retrieve data from ***config.json***
  3. ***config.json***: configuration file for ***echoconfig***

echoconfig APP and Configuration Uploading



NameConfig.java – NameConfig Class

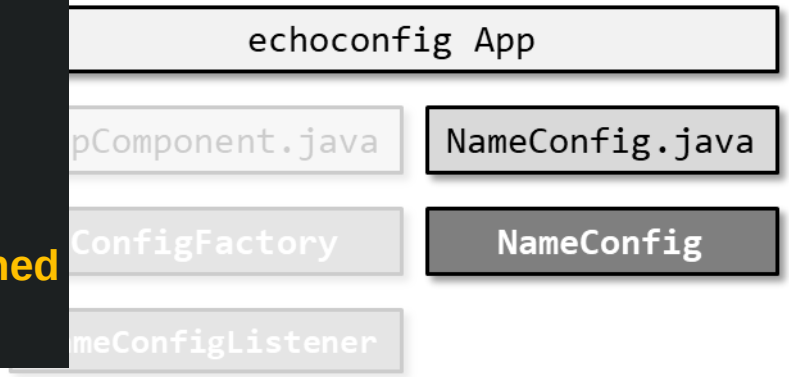
- Provide functions to:
  - Validate contents of ***config.json***
    - e.g. Check presence of required fields
  - Retrieve *name* value from ***config.json***

```
"apps": {
  "winlab.nctu.echoconfig": {
    "whoami": {
      "name": "Magikarp"
    }
  }
}
```

config.json

```
21 public class NameConfig extends Config<ApplicationId> {
22
23   public static final String NAME = "name";
24
25   @Override
26   public boolean isValid() {
27     return hasOnlyFields(NAME);
28   }
29
30   public String name() {
31     return get(NAME, null);
32   }
33 }
```

NameConfig* Class defined in *NameConfig.java



AppComponent.java – ConfigFactory

- Instantiate a factory to create a **NameConfig** object
  - The arguments serve as key for ONOS to fetch the correct factory

```
42  private final ConfigFactory factory =
43      new ConfigFactory<ApplicationId, NameConfig>(
44          APP_SUBJECT_FACTORY, NameConfig.class, "whoami") {
45        @Override
46        public NameConfig createConfig() {
47          return new NameConfig();
48        }
49      };
```

*Config*Factory Class

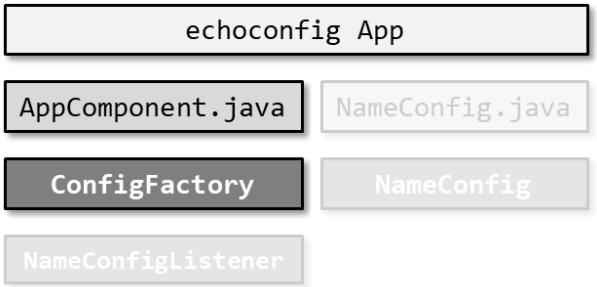
```
"apps": {
  "winlab.nctu.echoconfig": {
    "whoami": {
      "name": "Magikarp"
    }
  }
}
```

config.json

- Register ConfigFactory object with ONOS

```
61    appId = coreService.registerApplication("winlab.nctu.echoconfig");
62    cfgService.addListener(cfgListener);
63    cfgService.registerConfigFactory(factory);
64    log.info("Started");
```

Register Factory with cfgService



- ONOS will use the factory later when receives upload event

AppComponent.java – NameConfigListener

- Listen to network configuration event (e.g. A config file is uploaded)
- ONOS will call `event()` when it receives event

Instantiate a Listener

```
41  private final NameConfigListener cfgListener = new NameConfigListener();
```

```
74  private class NameConfigListener implements NetworkConfigListener {
75    @Override
76    public void event(NetworkConfigEvent event) {
77      if ((event.type() == CONFIG_ADDED || event.type() == CONFIG_UPDATED)
78          && event.configClass().equals(NameConfig.class)) {
79        NameConfig config = cfgService.getConfig(appId, NameConfig.class);
80        if (config != null) {
81          log.info("It is {}!", config.name());
82        }
83      }
```

*NameConfig*Listener Class

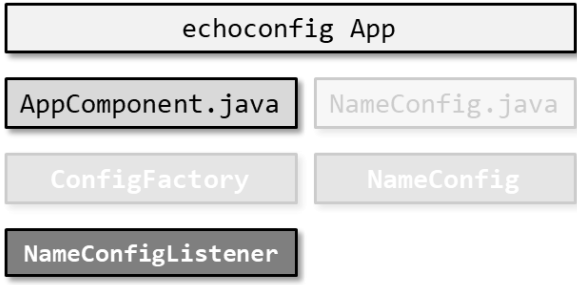
```
"apps": {
  "winlab.nctu.echoconfig": {
    "whoami": {
      "name": "Magikarp"
    }
  }
}
```

config.json

- Register the listener object with ONOS

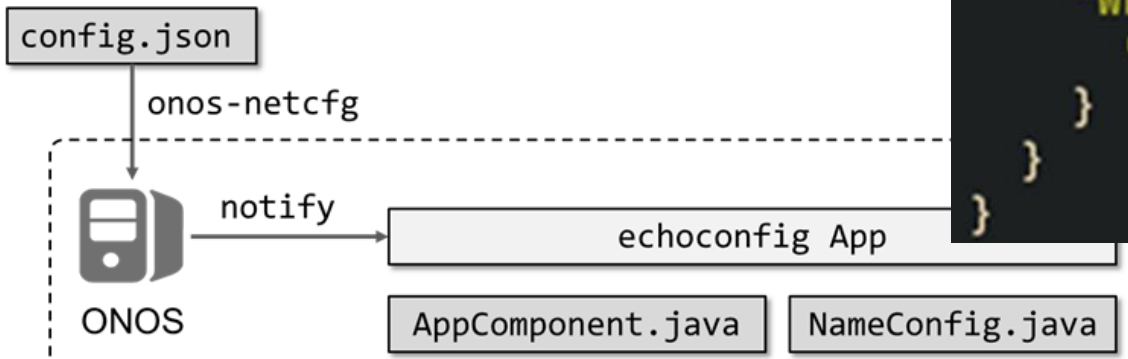
```
61    appId = coreService.registerApplication("winlab.nctu.echoconfig");
62    cfgService.addListener(cfgListener);
63    cfgService.registerConfigFactory(factory);
64    log.info("Started");
```

Register Listener with cfgService



echoconfig Demostration

- Upload config.json




```
"apps": {
  "winlab.nctu.echoconfig": {
    "whoami": {
      "name": "Magikarp"
    }
  }
}
```

config.json

- ONOS log will show following message

```
| 11 - org.apache.karaf.features.core - 4.2.6 | Starting bundles:
| 11 - org.apache.karaf.features.core - 4.2.6 |   winlab.nctu.echoconfig/1.0.0.SNAPSHOT
| 209 - winlab.nctu.echoconfig - 1.0.0.SNAPSHOT | Started
| 11 - org.apache.karaf.features.core - 4.2.6 | Done.
        | 190 - org.onosproject.onos-core-net - 2.2.0 | Application winlab.nctu.echoconfig has be

        | 209 - winlab.nctu.echoconfig - 1.0.0.SNAPSHOT | It is Magikarp!
```

ONOS Log

Outline

- Introduction to DHCP
  - What is DHCP?
  - DHCP Workflow
- Project 4
  - Overview
  - Workflow
  - Project Requirement
  - Supplements
- Upload Configuration for ONOS APPs
- **How to Test Your Unicast DHCP APP**
- Submission

Supplements

- "project4-supplement.zip" includes following files:
  1. Program and configuration files of a sample application **echoconfig**
     1) **AppComponent.java**:
        - Main program of **echoconfig** app
     2) **NameConfig.java**
        - validates and retrieves configuration data from config.json
     3) **config.json**:
        - configuration file for **echoconfig** app
  2. **Network Topology files for Unicast DHCP App**
     1) **topo.py**: mininet topology
     2) **dhcpd.conf**: DHCP configuration used by **topo.py**
  3. **Configuration file for Unicast DHCP App**
     - **unicastdhcp.json**: configuration file for unicast DHCP app

DHCP Utility Setup

- Install DHCP utility (isc-dhcp-server) before starting this project

```
bash$ sudo apt update && sudo apt install isc-dhcp-server
```

- To use dhcpd inside mininet host properly, you should modify AppArmor settings (**only need to be done for the first time**)
  - For server

```
bash$ sudo ln -s /etc/apparmor.d/usr.sbin.dhcpd \
                 /etc/apparmor.d/disable/
bash$ sudo apparmor_parser -R /etc/apparmor.d/usr.sbin.dhcpd
```

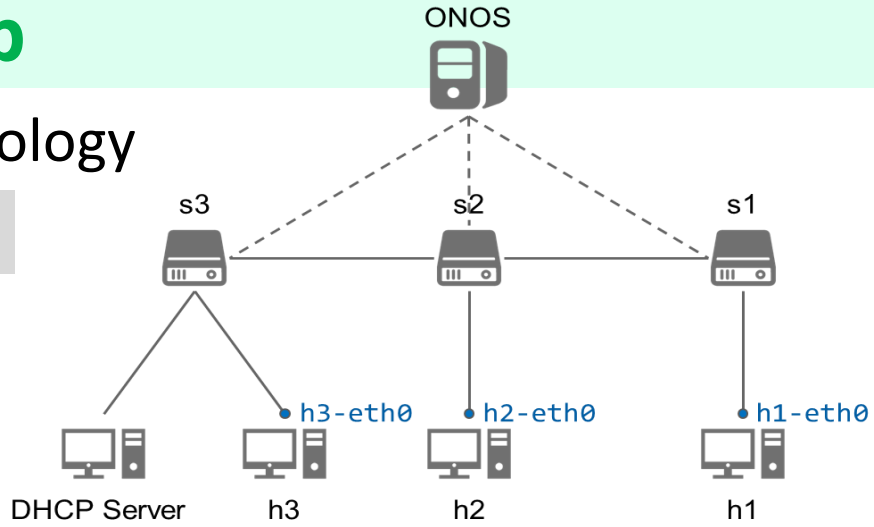
  - For client

```
bash$ sudo /etc/init.d/apparmor stop
bash$ sudo sed -i '30i /var/lib/dhcp{,3}/dhcpclient* lrw,' \
                       /etc/apparmor.d/sbin.dhclient
bash$ sudo /etc/init.d/apparmor start
```

How to Test Your App

- Use **topo.py** to build the topology

```
bash$ sudo python topo.py
```



Three hosts without IP addresses in the given topology

- Run DHCP on h1

```
mininet> h1 dhclient -v h1-eth0
```

✓ Note: Release current lease before re-issue a DHCP request on an interface (to observe all messages of a DHCP transaction)

```
mininet> h1 dhclient -r h1-eth0
```

Demonstration

1. h1-eth0 does not have an IPv4 address yet

```
mininet> h1 ifconfig h1-eth0
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::e8e9:78ff:fefb:fd01  prefixlen 64  scopeid 0x20<link>
        ether ea:e9:78:fb:fd:01  txqueuelen 1000  (Ethernet)
```

2. Observe DHCP procedure on h1-eth0

```
mininet> h1 dhclient -v h1-eth0
Internet Systems Consortium DHCP Client 4.3.5
Copyright 2004-2016 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/h1-eth0/ea:e9:78:fb:fd:01
Sending on   LPF/h1-eth0/ea:e9:78:fb:fd:01
Sending on   Socket/fallback
DHCPDISCOVER on h1-eth0 to 255.255.255.255 port 67 interval 3 (xid=0xd74d5b7c)
DHCPDISCOVER on h1-eth0 to 255.255.255.255 port 67 interval 3 (xid=0xd74d5b7c)
DHCPREQUEST of 10.1.11.100 on h1-eth0 to 255.255.255.255 port 67 (xid=0x7c5b4dd7)
DHCPOFFER of 10.1.11.100 from 10.1.11.3
DHCPACK of 10.1.11.100 from 10.1.11.3
bound to 10.1.11.100 -- renewal in 232 seconds.
```

DHCP Messages

3. h1-eth0 now has an IPv4 address

```
mininet> h1 ifconfig h1-eth0
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.1.11.100  netmask 255.255.255.0  broadcast 10.1.11.255
        inet6 fe80::e8e9:78ff:fefb:fd01  prefixlen 64  scopeid 0x20<link>
        ether ea:e9:78:fb:fd:01  txqueuelen 1000  (Ethernet)
```

Outline

- Introduction to DHCP
  - What is DHCP?
  - DHCP Workflow
- Project 4
  - Overview
  - Workflow
  - Project Requirement
  - Supplements
- Upload Configuration for ONOS APPs
- How to Test Your Unicast DHCP APP
- **Submission**

Submission

- Files
  - All files of your application
- Submission
  - Upload ".zip" file to e3
    - Name: **project4_<studentID>.zip**
  - Incorrect naming convention or format will not be scored

References

- ONOS Java API (2.2.0)
- The Network Configuration Service