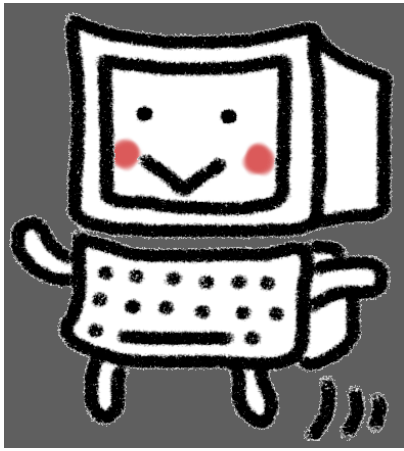


Project 6

Unicast DHCP Application

Date: 2019/05/09 (Thu.)

Deadline: 2019/05/26 (Sun.)

Outline

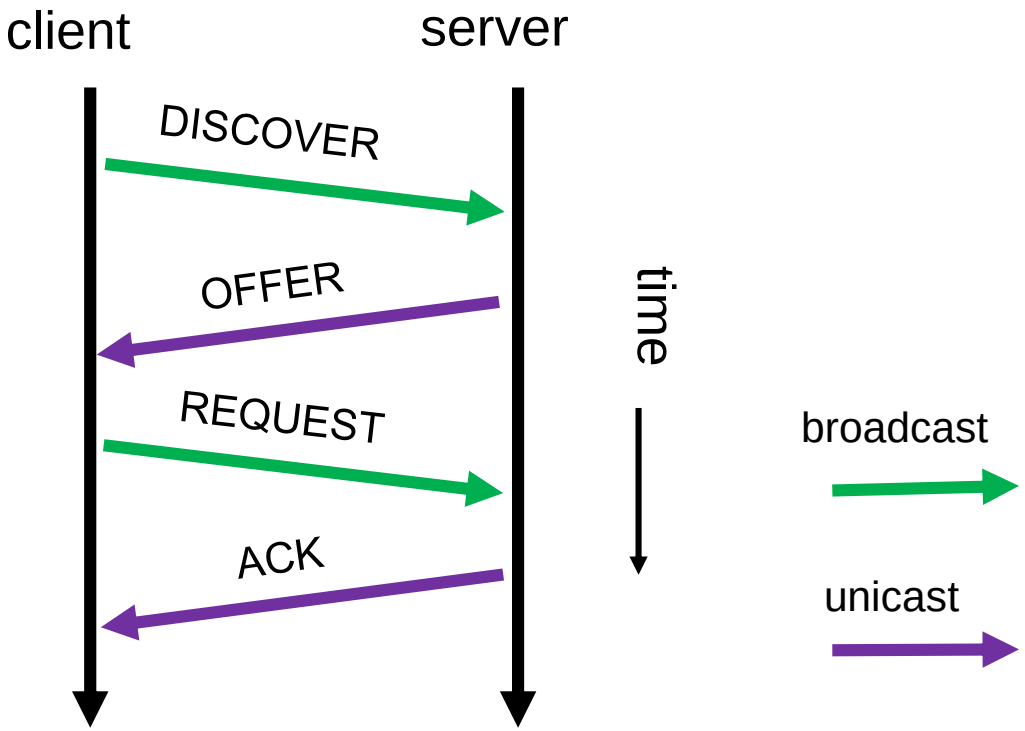
- About DHCP
- Project 6 Requirements
- Hints

Outline

- About DHCP
- Project 6 Requirements
- Hints

About DHCP (1/8)

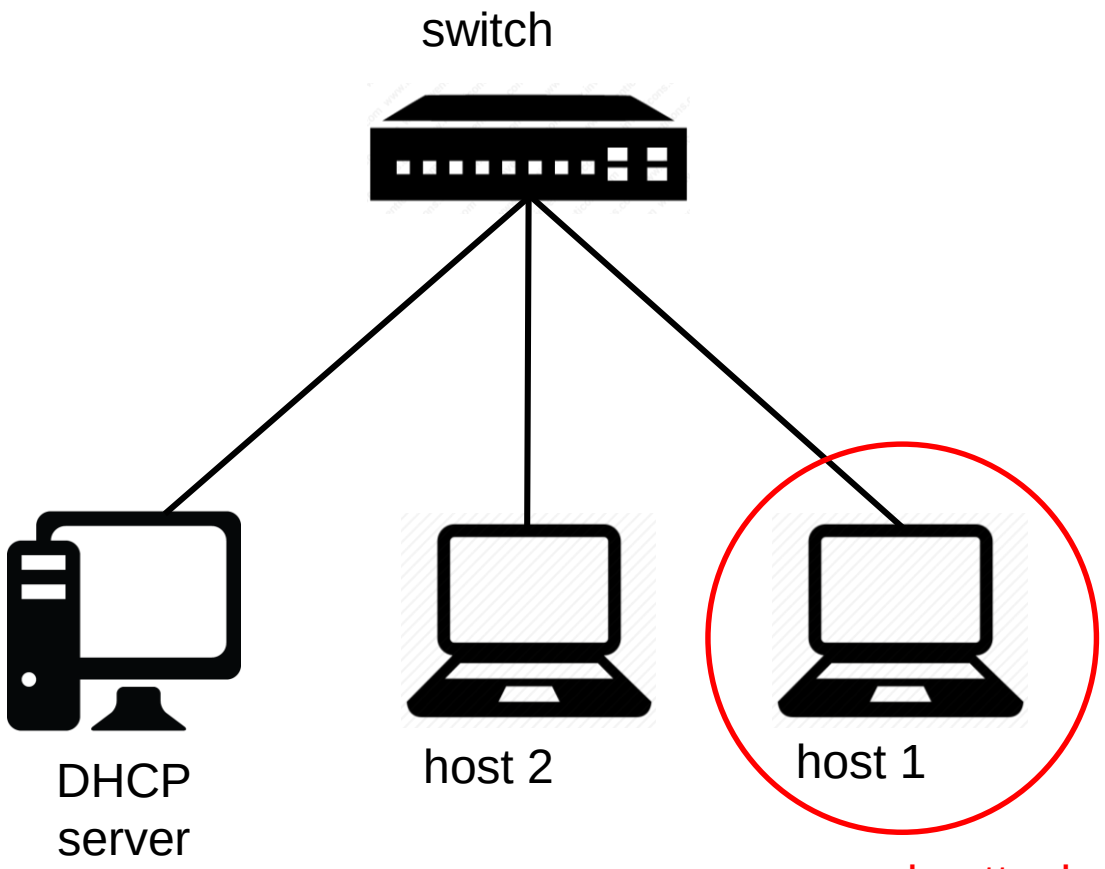
- Dynamic Host Configuration Protocol
  - Provide necessary information for a host to access network
    - IP address, gateway, DNS (Domain Name Server), etc.
  - A DHCP transaction is completed by 4-way handshake:



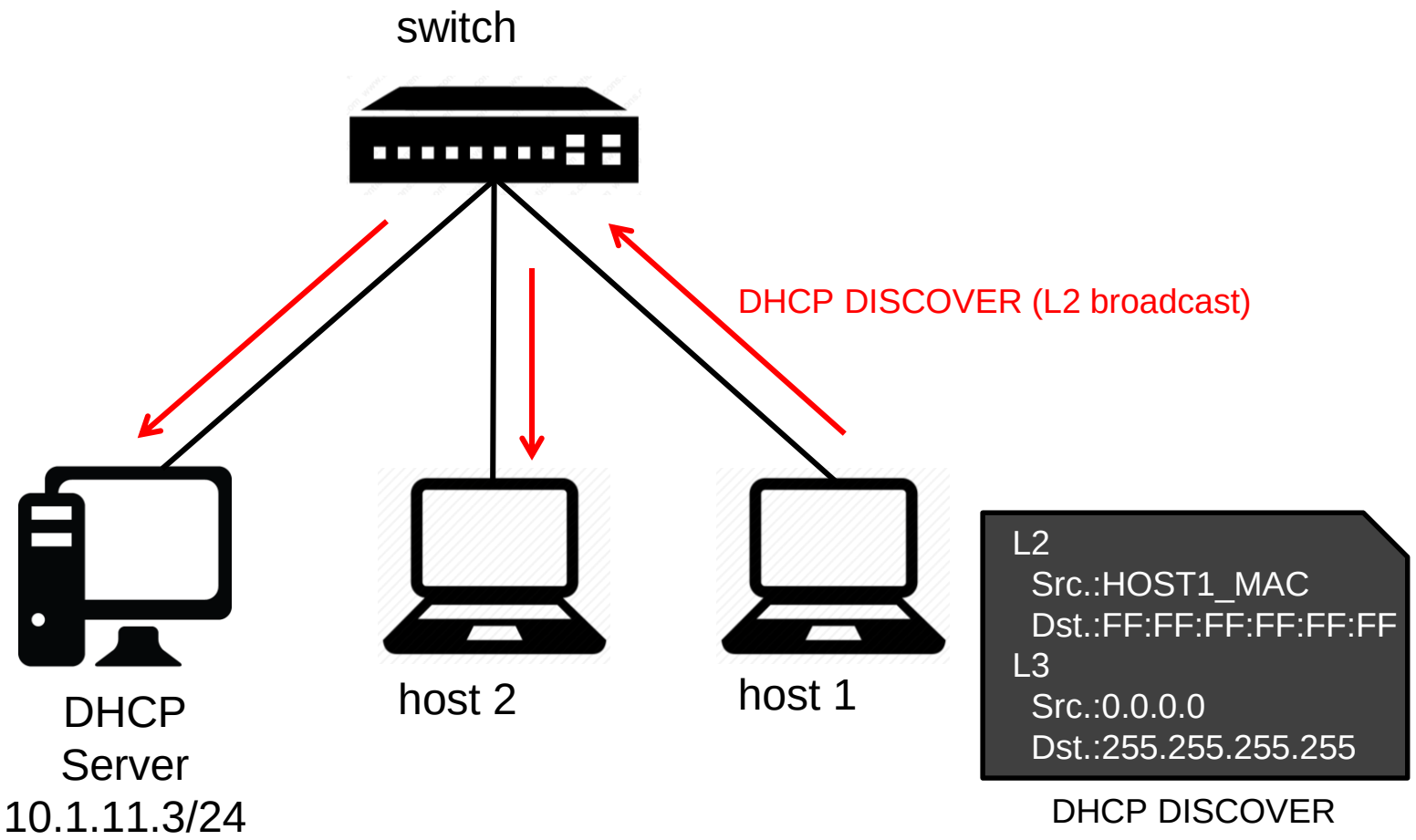
About DHCP (2/8)

9	5.676080125	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover -
10	5.677121518	10.1.11.3	10.1.11.25	DHCP	342	DHCP Offer -
11	5.677344173	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request -
12	5.678388535	10.1.11.3	10.1.11.25	DHCP	342	DHCP ACK -
13	6.197545870	02:eb:cf:c1:c2:89	LLDP_Multicast	LLDP	130	TTL = 120

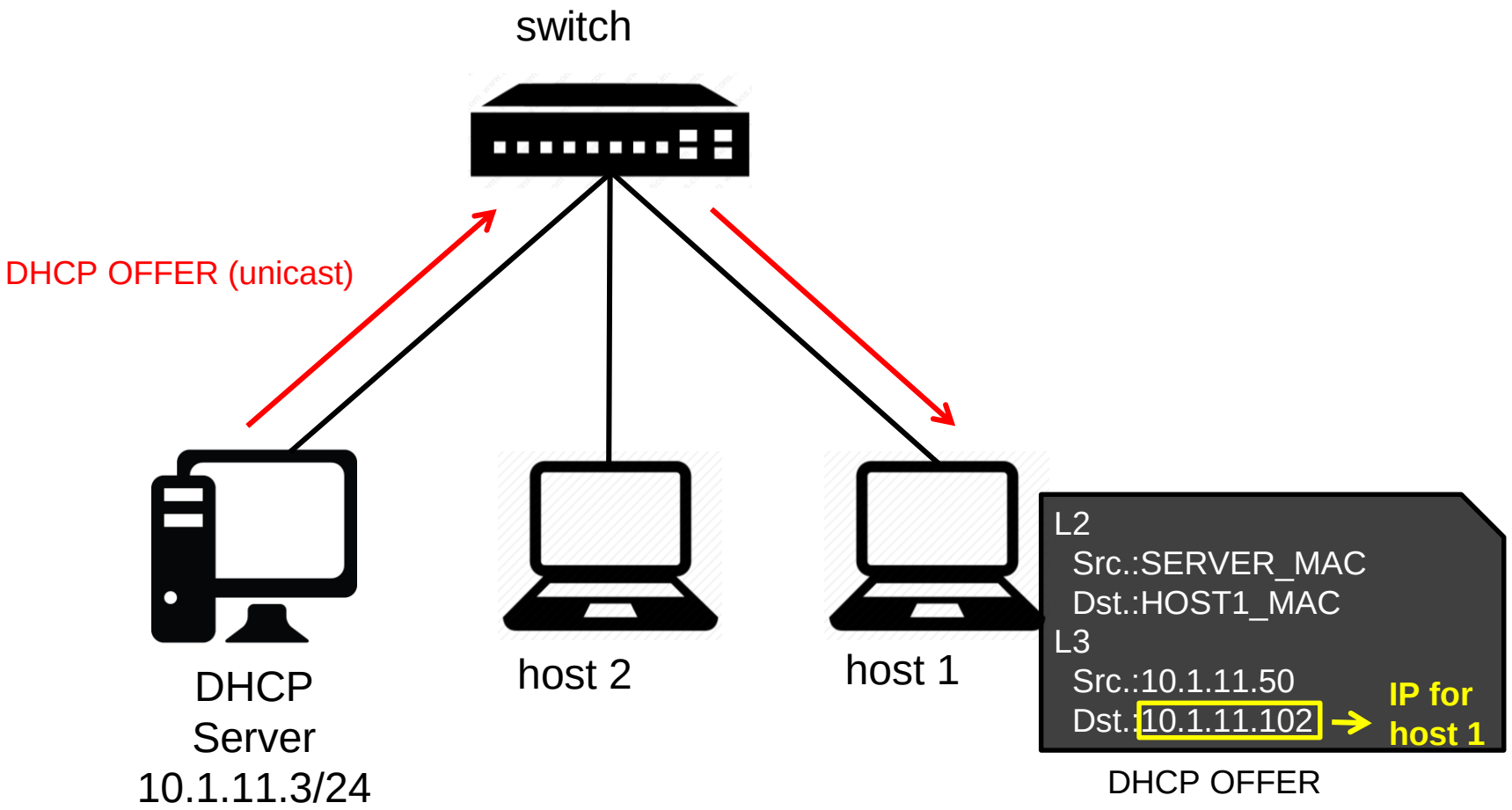
About DHCP (3/8)



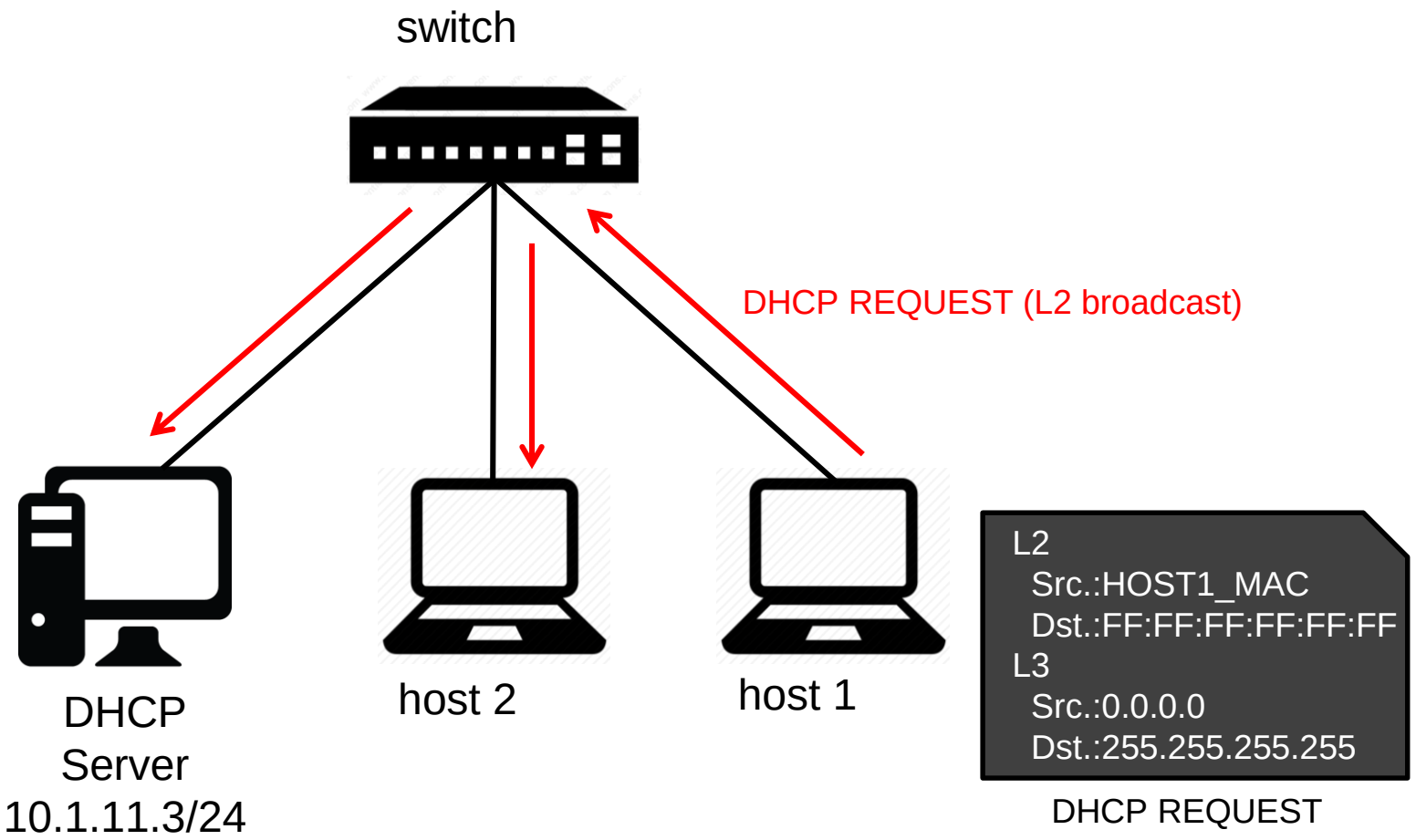
About DHCP (4/8)



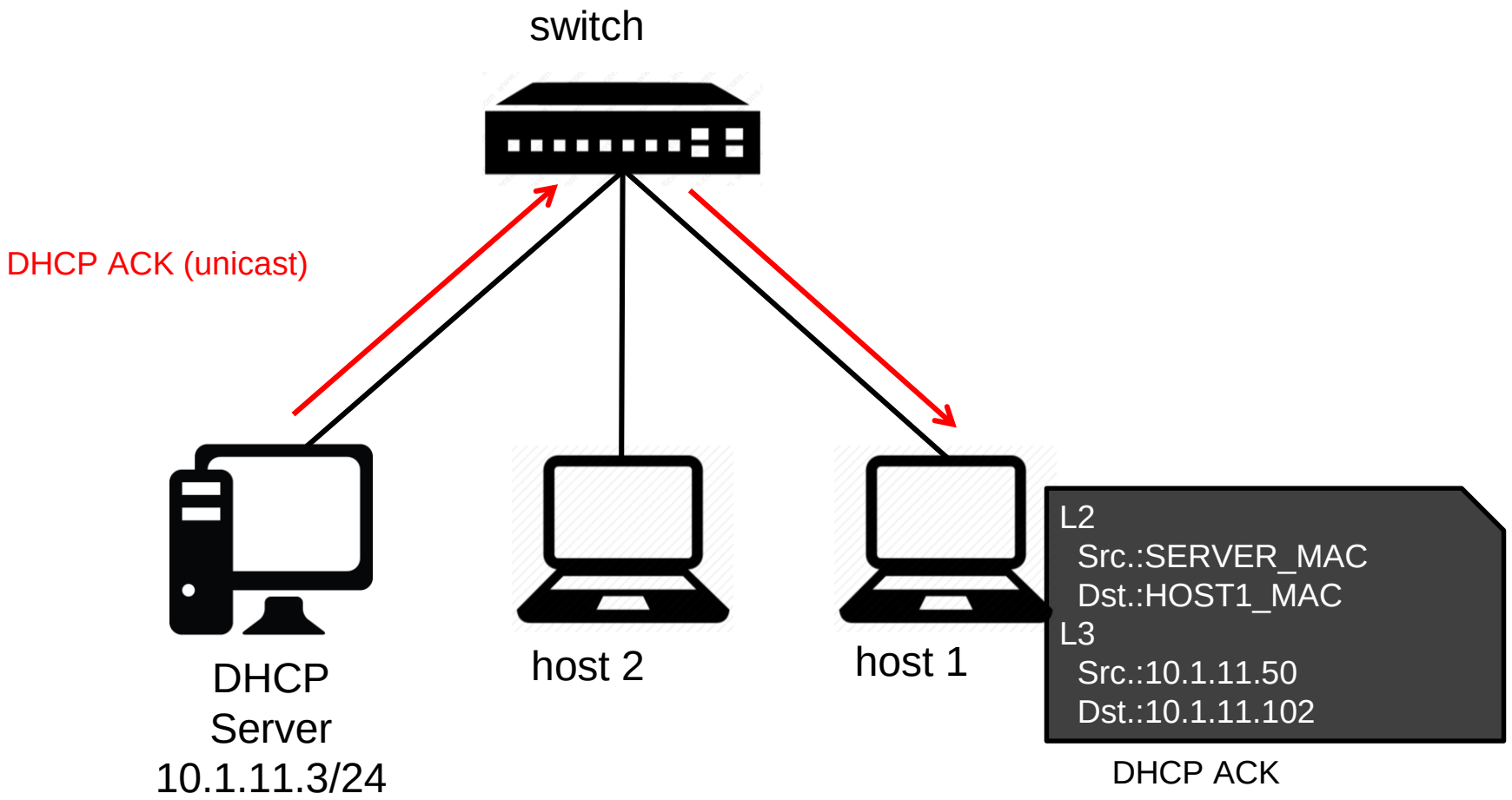
About DHCP (5/8)



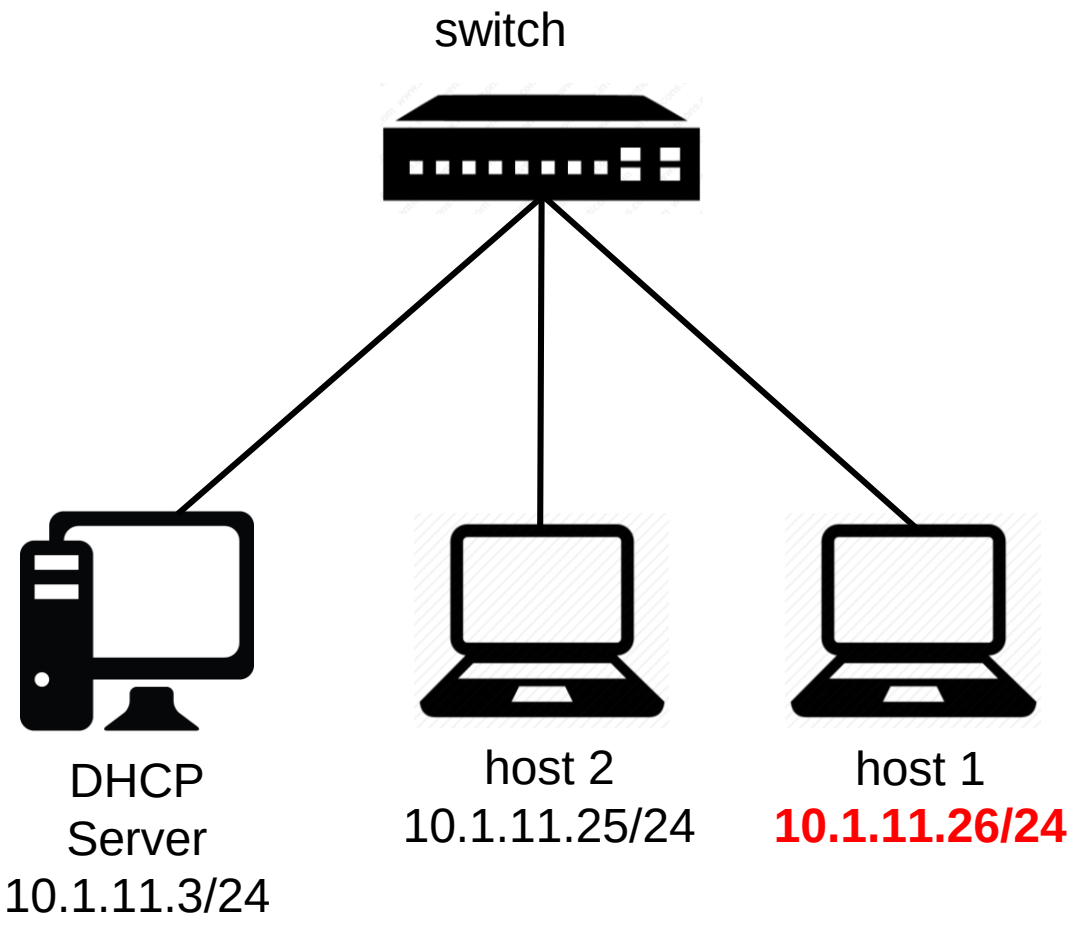
About DHCP (6/8)



About DHCP (7/8)



About DHCP (8/8)

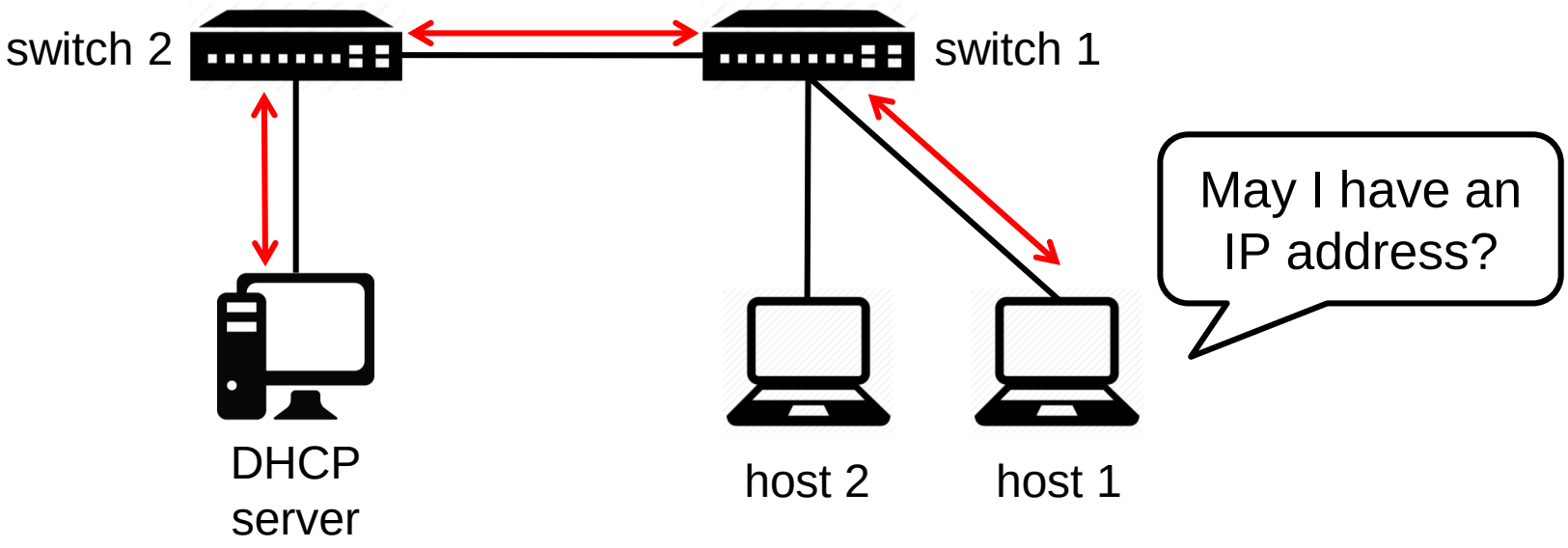


Outline

- About DHCP
- Project 6 Requirements
- Hints

Project 6 Requirement

- In this project, you need to implement an unicast DHCP application that support:
  1. Dynamically set DHCP server's connect point through REST API (configuration service).
  2. Compute path between DHCP client and DHCP server.
  3. Install flow rules to forward DHCP transaction traffic.

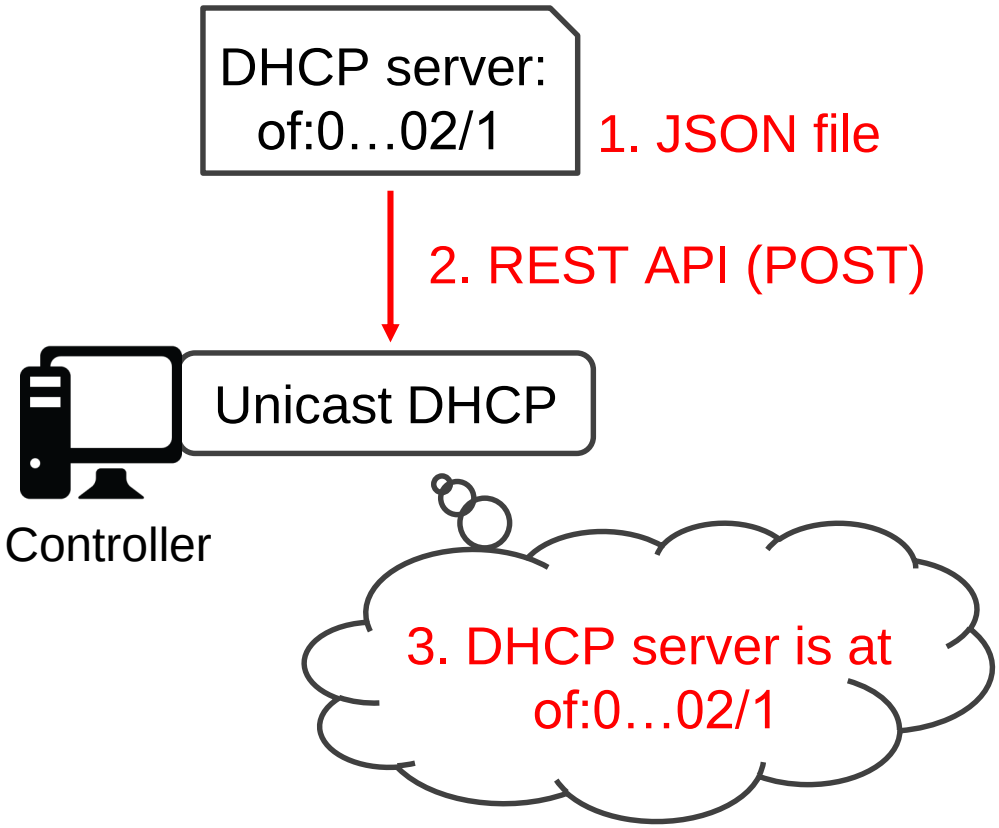


Project 6 Requirement - Workflow (1/7)

1. Before DHCP transaction
   a. Network manager upload configuration file to ONOS
   b. Your application should install a flow rule to request DHCP packets
2. DHCP DISCOVER will be packet-in to controller by switch
3. Controller compute path between DHCP client and server
4. Controller install flow rules to forward DHCP packets
5. Done

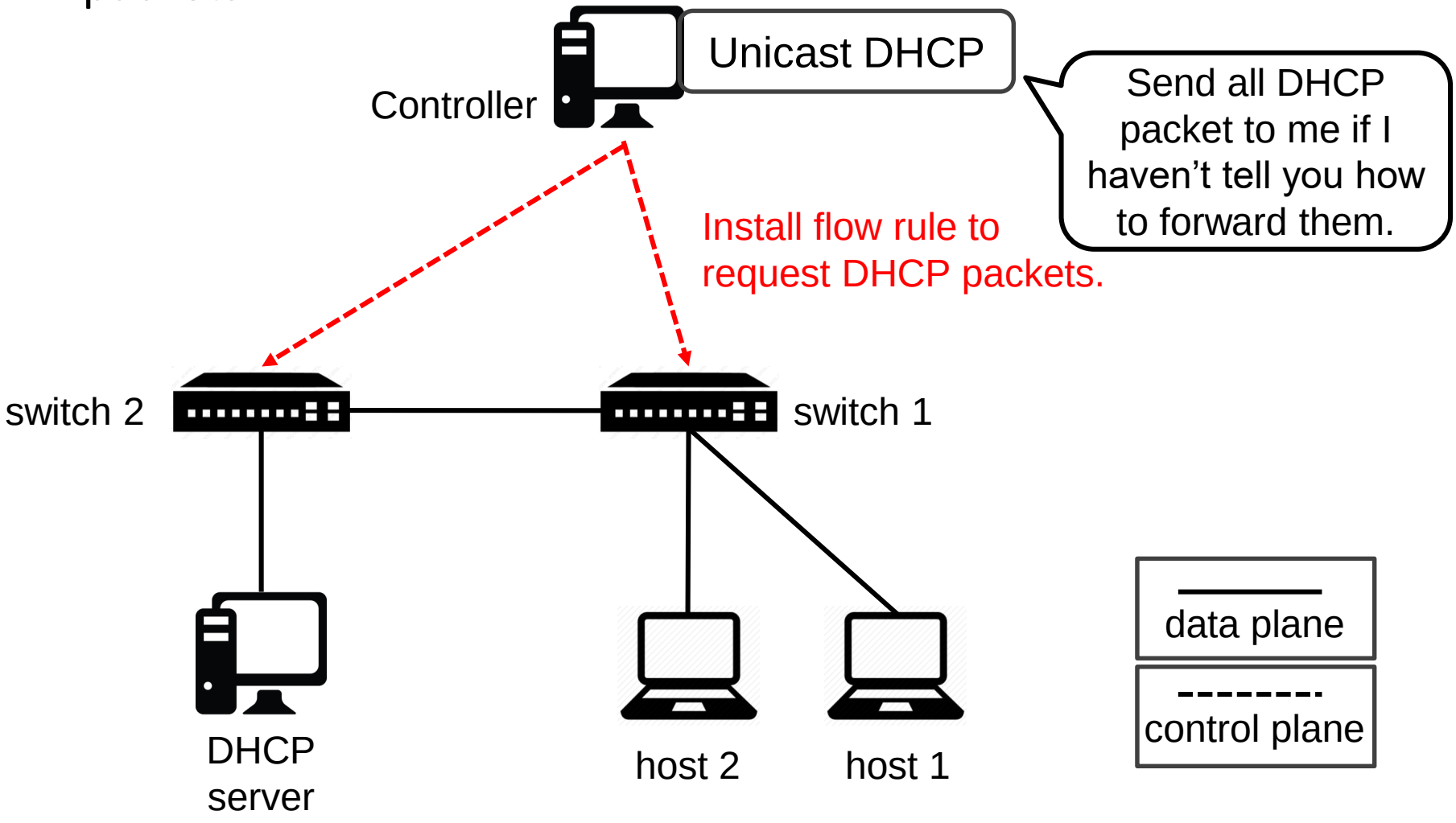
Project 6 Requirement - Workflow (2/7)

1. a. Network manager upload configuration file to ONOS



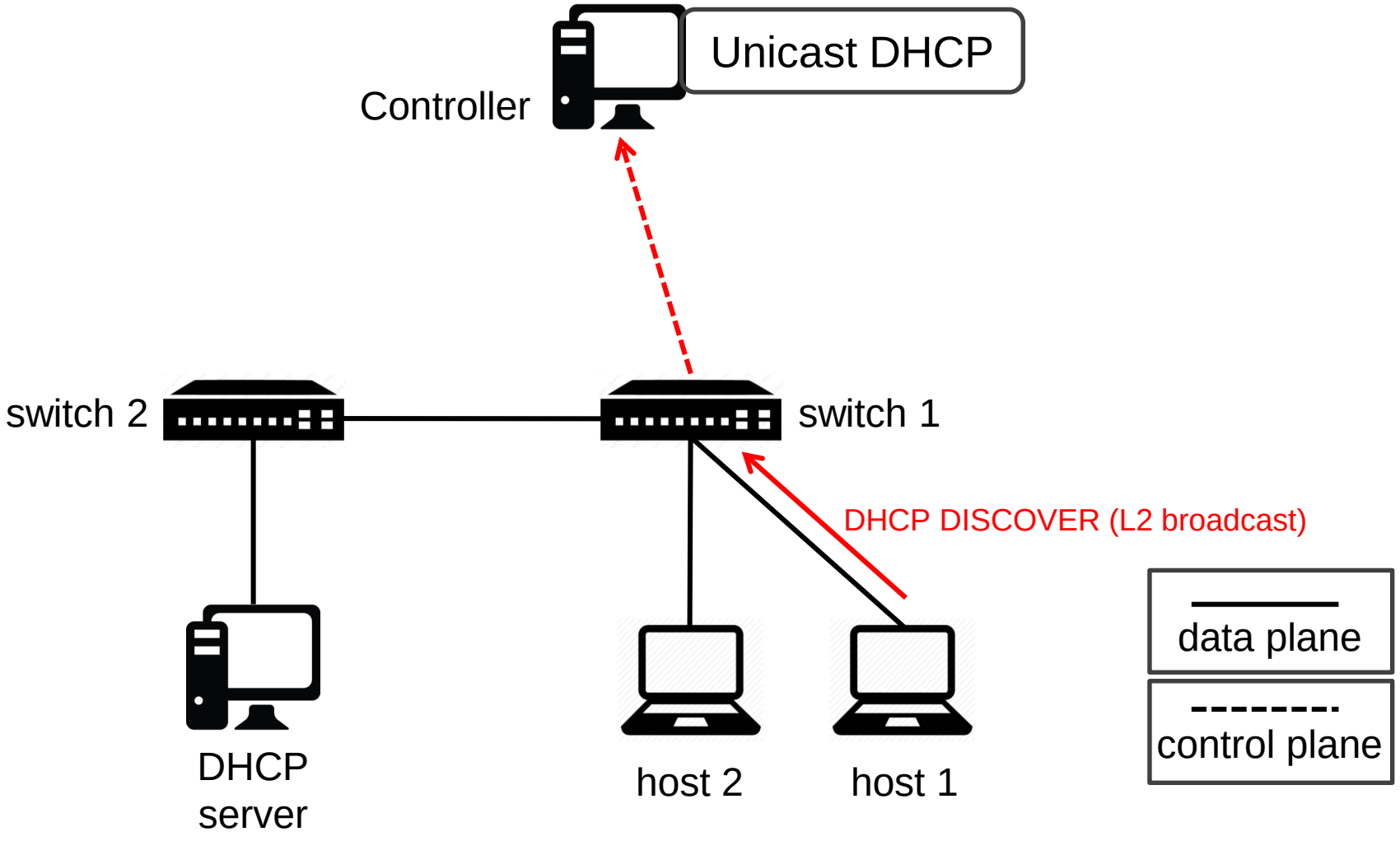
Project 6 Requirement - Workflow (3/7)

1. b. Your application should install a flow rule to request DHCP packets



Project 6 Requirement - Workflow (4/7)

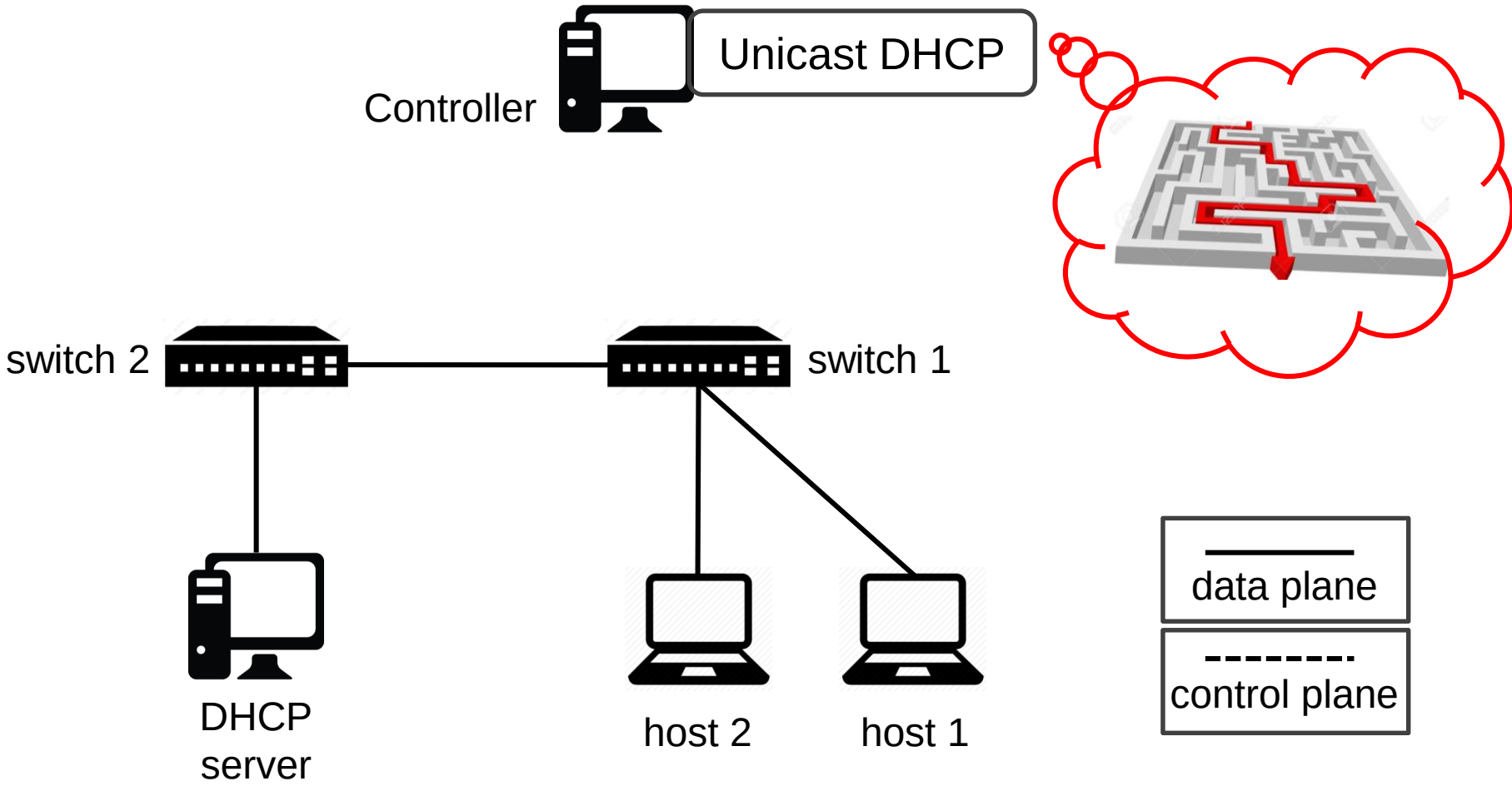
2. DHCP DISCOVER will be packet-in to controller by switch



Project 6 Requirement - Workflow (5/7)

3. Controller compute path between DHCP client and server

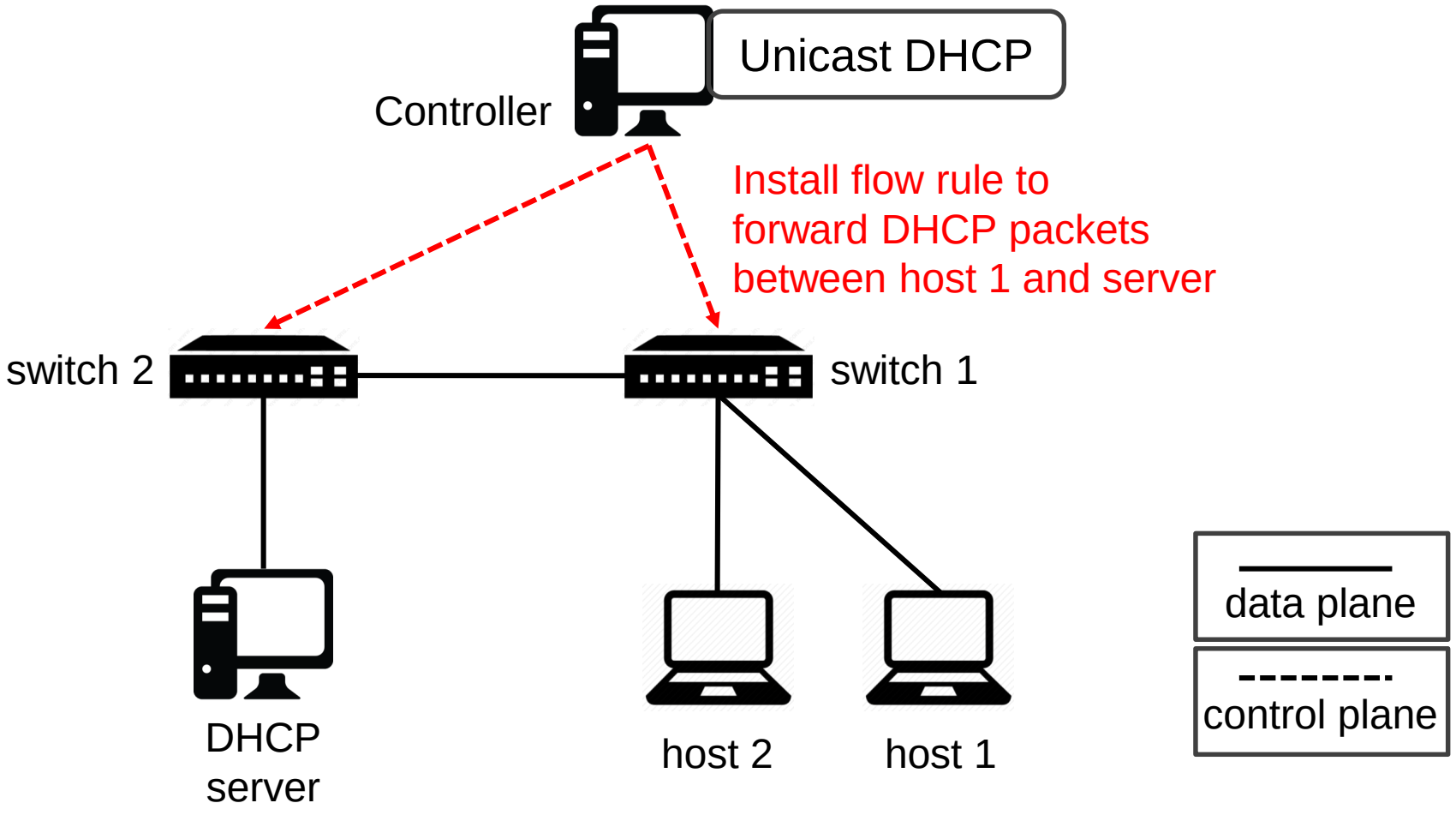
   (Note: a bit like what you did in project 4)



Project 6 Requirement - Workflow (6/7)

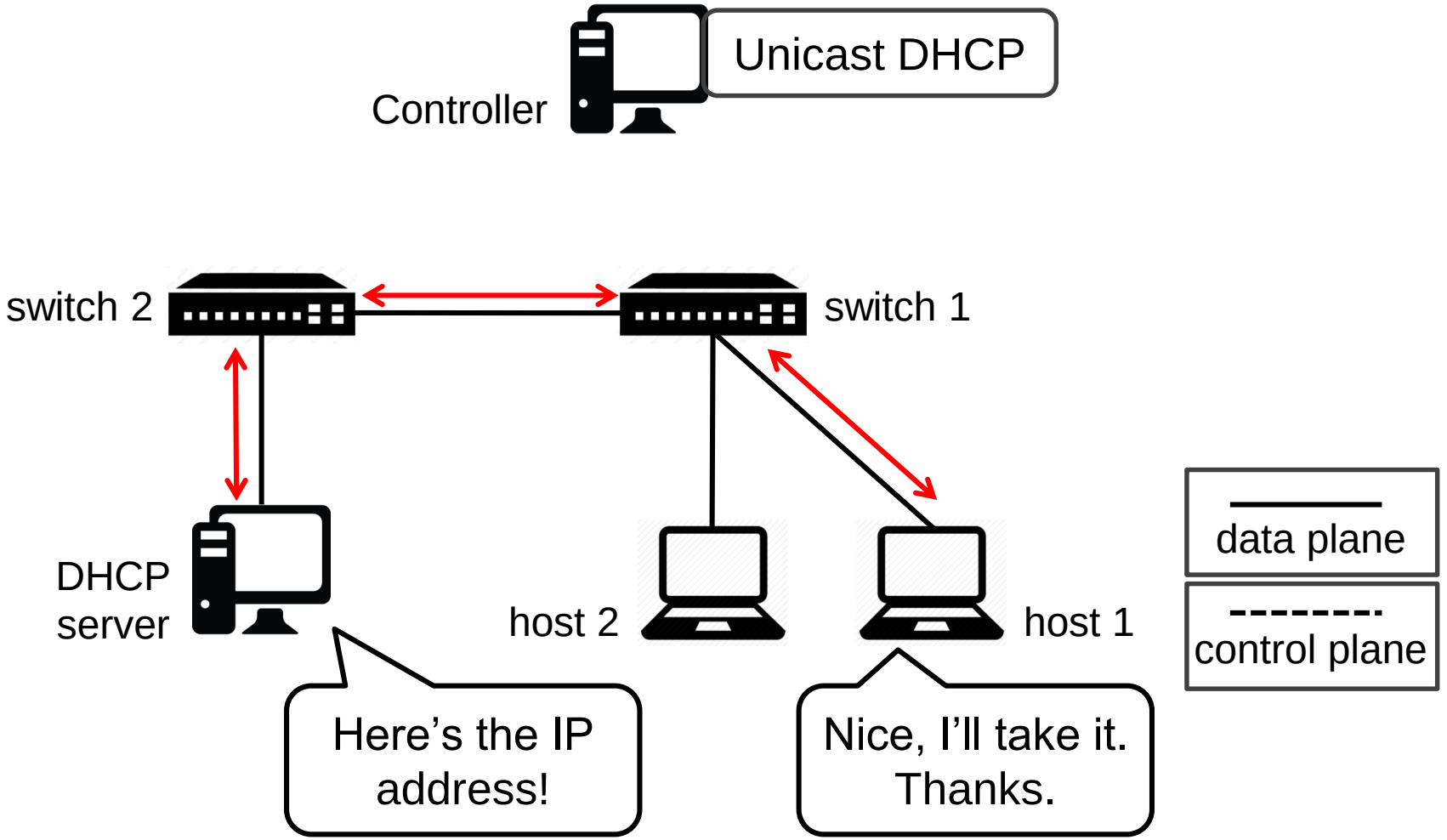
4. Controller install flow rules to forward DHCP packets

   (Note: proactive and reactive forwarding are both acceptable)



Project 6 Requirement - Workflow (7/7)

5. Done

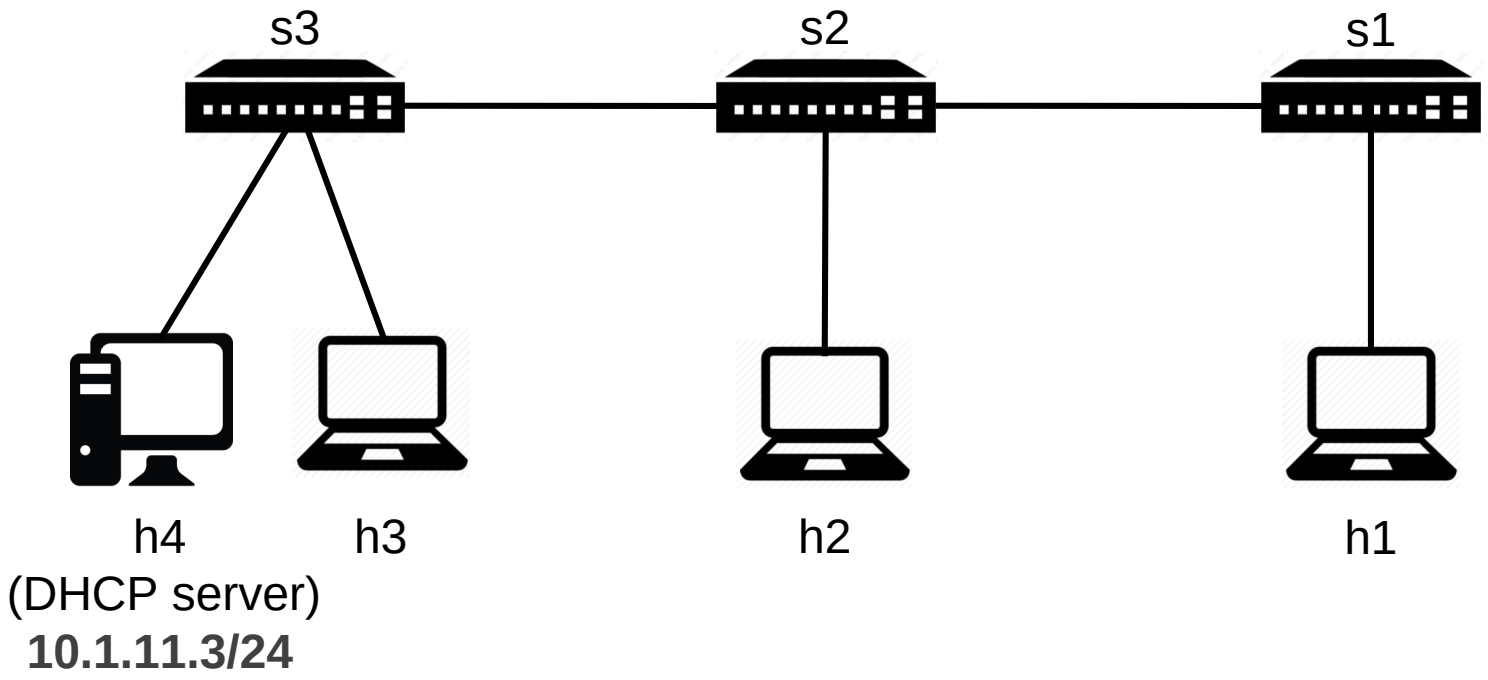


Project 6 Requirement - Provided File

- "project_6.zip" includes two directories and following files:
  1. **EchoConfig**: (reference for ONOS users)
     - EchoConfig is an example application that read configuration file through REST API and echo it to log file.
     - Sample configuration file is also provided.
     - This application is provided with ONOS version only.
  2. **SDN-NFV-Project6**:
     a. **topo.py**: mininet topology
     b. **dhcpd.conf**: configuration file for mininet topology
     c. **unicastdhcp.json**: configuration file for unicast DHCP app (example for ONOS users)

Project 6 Requirement - Topology

- The topology provided in ”project_6.zip” consist of:
  - 3 switches
  - 3 hosts
  - 1 DHCP server



Project 6 Requirement - Commands (1/2)

- Remember to install isc-dhcp-server before you start this project:

```
$ sudo apt-get install isc-dhcp-server
```

- To use dhcpd, we should disable/modify AppArmor (only need to be done in the first time)

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

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

Project 6 Requirement - Commands (1/2)

- Use following command to start the topology:

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

- In mininet CLI, use following command to ask an IP for a host (which means to start an DHCP transaction):

```
mininet> h1 dhclient {interface_name}
```

Project 6 Requirement - Configuring

- Use following command to upload a config file by REST API:
  - For ONOS:

```
$ onos-netcfg {controller_IP} {json_file_name}
```

  - For RYU:
    - https://osrg.github.io/ryu-book/zh_tw/html/rest_api.html

Submit to e3

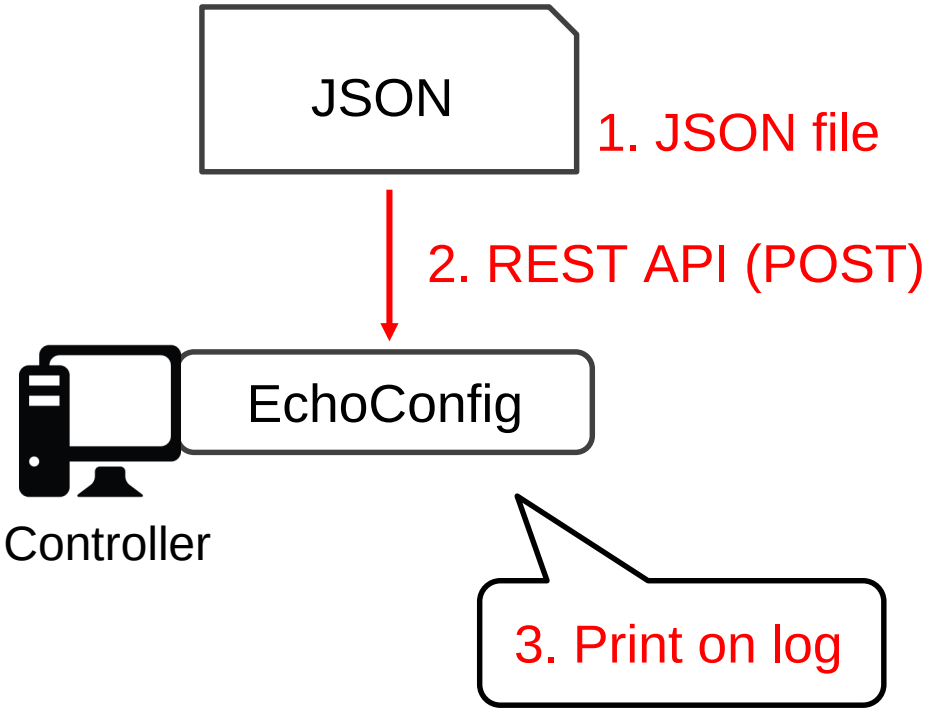
- Files
  - All files of your application
  - Config file you used in provided topology
- Submit
  - Upload “.zip” file to e3
    - Named: **project6_studentID.zip**
  - Wrong file name or format would not be scored

Outline

- About DHCP
- Project 6 Requirements
- Hints
  1. EchoConfig

Hints - EchoConfig

- EchoConfig is an example application that read configuration file through REST API and echo it to log file.
- We will explain this application for ONOS and RYU separately.

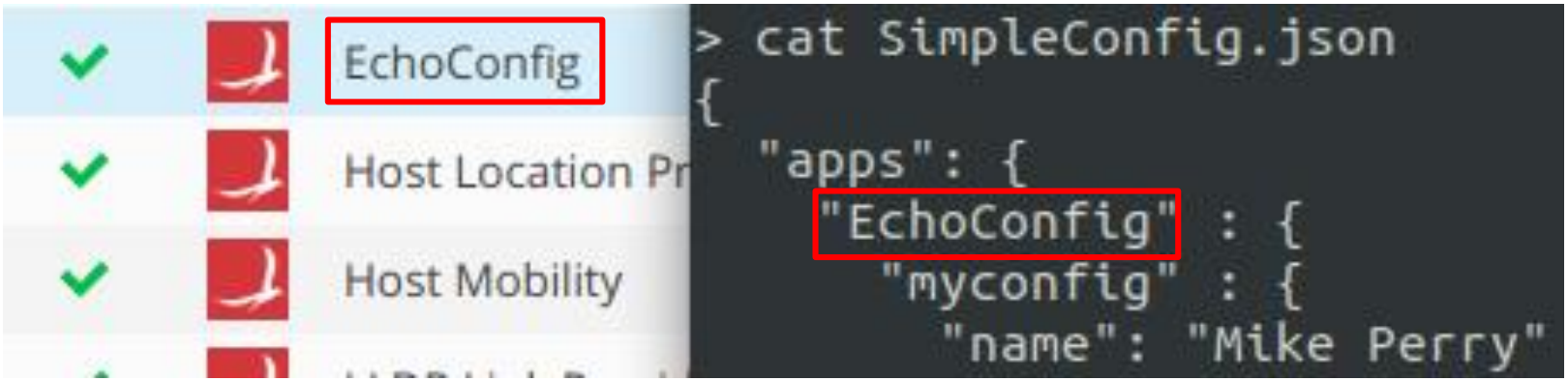


Hints – EchoConfig ONOS version (1/2)

- Including:
  1. 2 source code files
     - AppComponent.java: main source code file for the application
     - MyConfig.java: the config class which will be imported in "AppComponent.java".
  2. 1 sample config file
- Create your own EchoConfig application according to provided source code.
  - ➢ You may need to do some modification according to your package name.
- Compile your application and install into ONOS

Hints – EchoConfig ONOS version (2/2)

- Check that the application title is correct in configuration file:



- Upload the config file
- The EchoConfig application will echo the config file in ONOS log: