Project 5

Proxy ARP

Deadline: 2021/12/01 (WED) 23:55

Outline

- Introduction to ARP
  - What is ARP
  - ARP Request/Reply Format
- Proxy ARP
  - What is Proxy ARP
  - Workflow of Proxy ARP in SDN
- Project 5 Requirements
- Reference

Outline

- **Introduction to ARP**
  - What is ARP
  - ARP Request/Reply Format
- Proxy ARP
  - What is Proxy ARP
  - Workflow of Proxy ARP in SDN
- Project 5 Requirements
- Reference

What is Address Resolution Protocol (ARP)

- Used to discover Link Layer address (e.g. MAC) with the given Network Layer address (e.g. IPv4)
- Use flooding to discover devices
  - Destination Ethernet address of ARP Request is broadcast address
- Hosts maintain a ARP table for mapping of IPv4 to MAC address

ARP Procedure

- - - → ARP Request with h2-IP and broadcast MAC
- - - → ARP Reply with h2-IP and h2-MAC
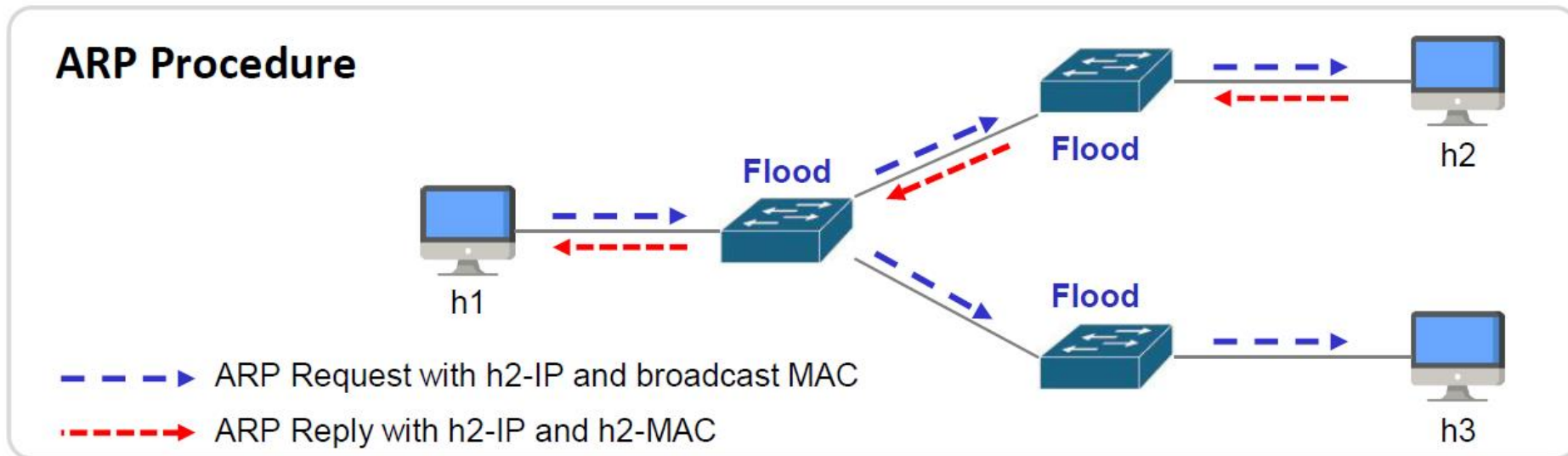
ARP Request Packet Frame

● Following table depicts ARP request packet format (Find IPv4 from MAC)

Hardware Type (Ethernet = 1)		Protocol Type (IPv4 = 0x0800)
Hardware Length (Ethernet = 6)	Protocol Length (IPv4 = 4)	Operation Code (ARP request = 1)
Sender Hardware Address (h1-MAC)		
Sender Protocol Address (h1-IP)		
Target Hardware Address (00:00:00:00:00:00)		
Target Protocol Address (h2-IP)		

ARP Reply Packet Frame

● Following table depicts ARP reply packet format (Find IPv4 from MAC)

| Hardware Type (Ethernet = 1)		Protocol Type (IPv4 = 0x0800)
Hardware Length (Ethernet = 6)	Protocol Length (IPv4 = 4)	Operation Code (ARP request = 1)
Sender Hardware Address (**h2-MAC**)		
Sender Protocol Address (h2-IP)		
Target Hardware Address (h1-MAC)		
Target Protocol Address (h1-IP)		

Outline

- Introduction to ARP
  - What is ARP
  - ARP Request/Reply Format
- Proxy ARP
  - What is Proxy ARP
  - Workflow of Proxy ARP in SDN
- Project 5 Requirements
- References

What is Proxy ARP

- A Proxy device answers ARP requests for IP address on behalf of other devices
  - The proxy device could be router, firewall, etc.
  - The replied MAC address belongs to the **proxy device**
- In the context of SDN, controller serves as proxy device
  - However, the replied MAC address belongs to the **target host**
  - Benefits:
    - Decreases workload of network devices
   - Prevent issues like broadcast storm

s1

h1   h2
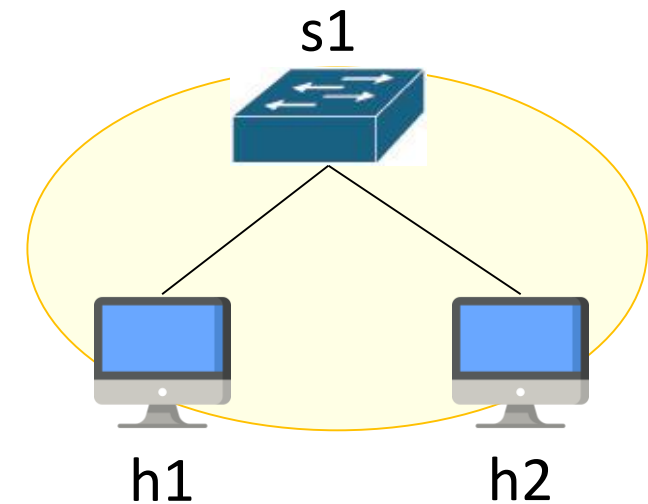
Workflow of Proxy ARP in SDN

1. Sender sends ARP request
2. Edge switch Packet-Ins the request to controller
3. Proxy ARP learns address mappings of the sender
4. Proxy ARP fetches target's MAC from ARP table
   - If mapping not exists:
     a. Flood request packet to edge ports
     b. Wait for ARP reply from the target
     c. Learn address mapping of the target
   - Else, generate an ARP reply packet
5. Proxy ARP Packet-Outs the ARP reply to the sender

- h1 sends ARP request to h2
- The very first switch (s1) performs Packet-In to controller

ARP table maintained by Proxy ARP

Packet-In

controller

switches

ARP Request
ARP Reply

h1

s1
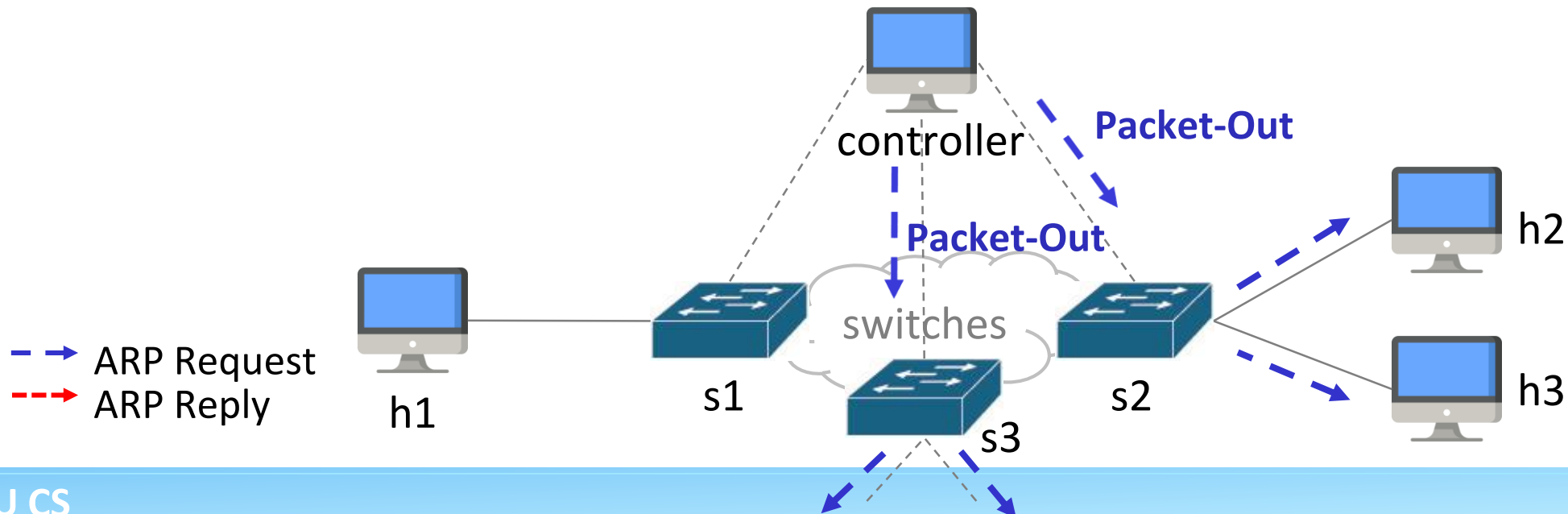
s3

s2

h2

h3
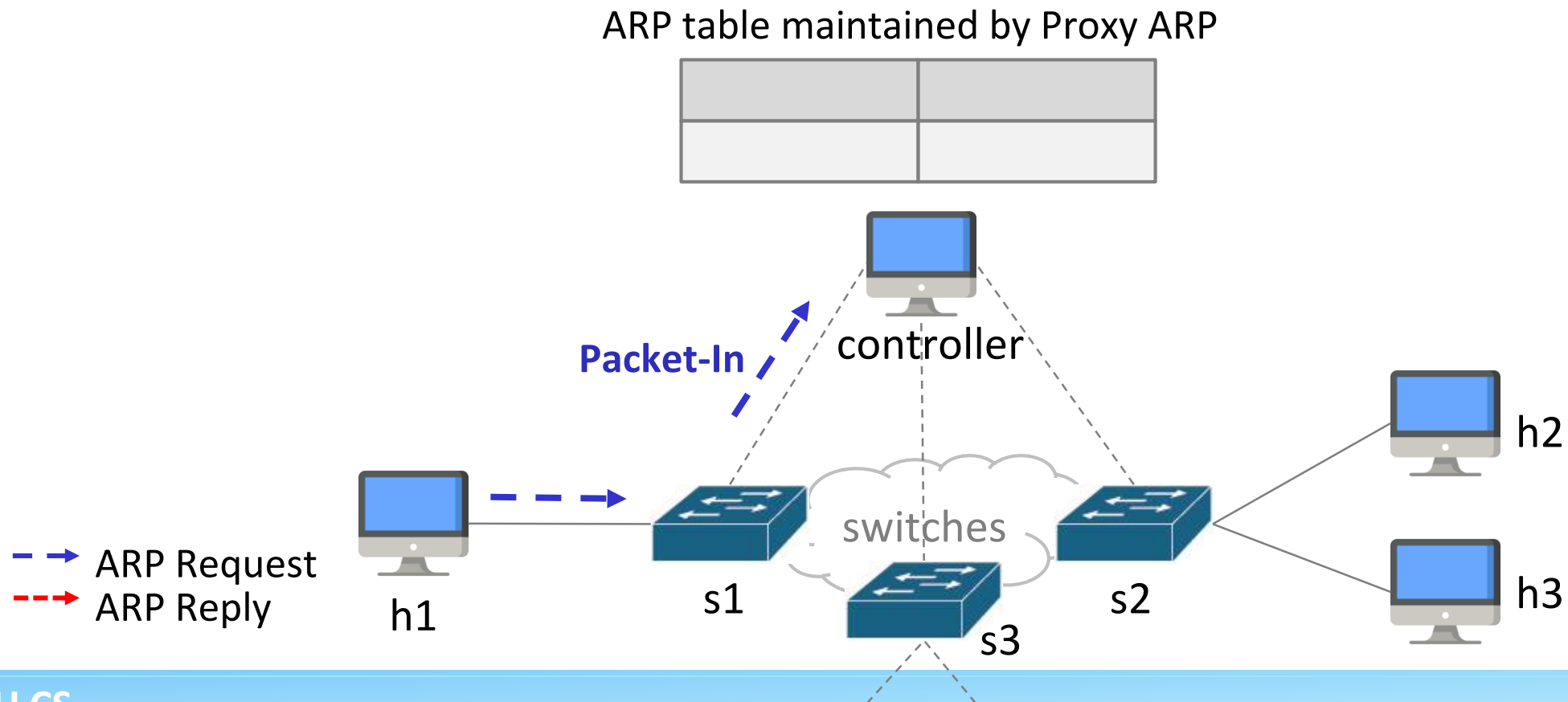
- Controller learns mapping of IP to MAC address of h1
- With mapping of h2 absent, controller needs to discover h2
  - By Packet-Outing the ARP request to all edge ports (ports connecting to host) other than h1's
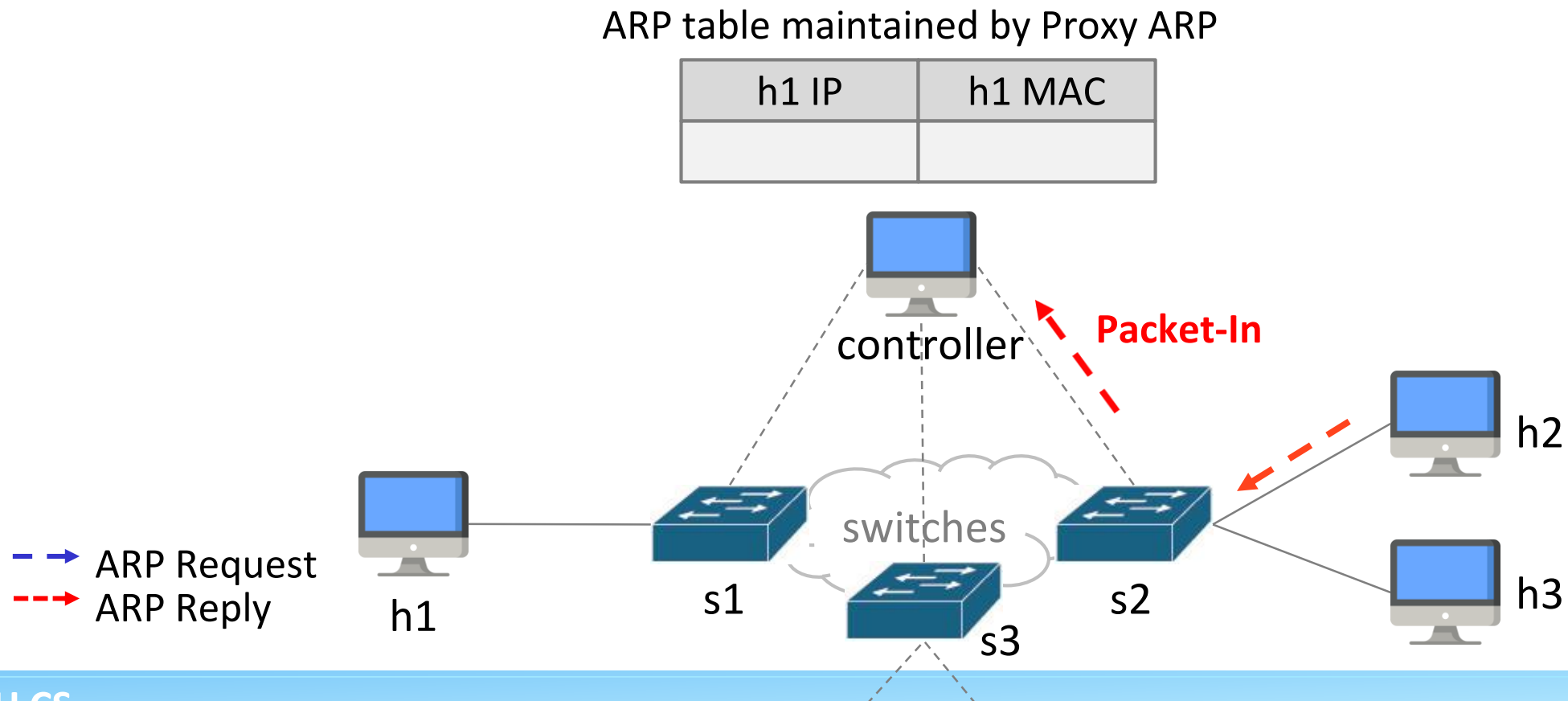
ARP table maintained by Proxy ARP

h1 IP	h1 MAC

controller

Packet-Out

Packet-Out

switches

h1

s1

s2

s3

h2

h3

ARP Request
ARP Reply

- All other hosts receive ARP requests, but only h2 will send ARP reply
- Again, the very first switch (s2) performs Packet-In to controller

ARP table maintained by Proxy ARP

h1 IP	h1 MAC

Packet-In

controller

switches

s1

s2

s3

h1

h2

h3

- - → ARP Request
- - → ARP Reply

- Controller learns mapping of requested IP to MAC address of h2
- Controller Packet-Outs the ARP reply to the switch connected to h1

ARP table maintained by Proxy ARP

h1 IP	h1 MAC
h2 IP	h2 MAC

controller

Packet-Out

switches

h2

h3

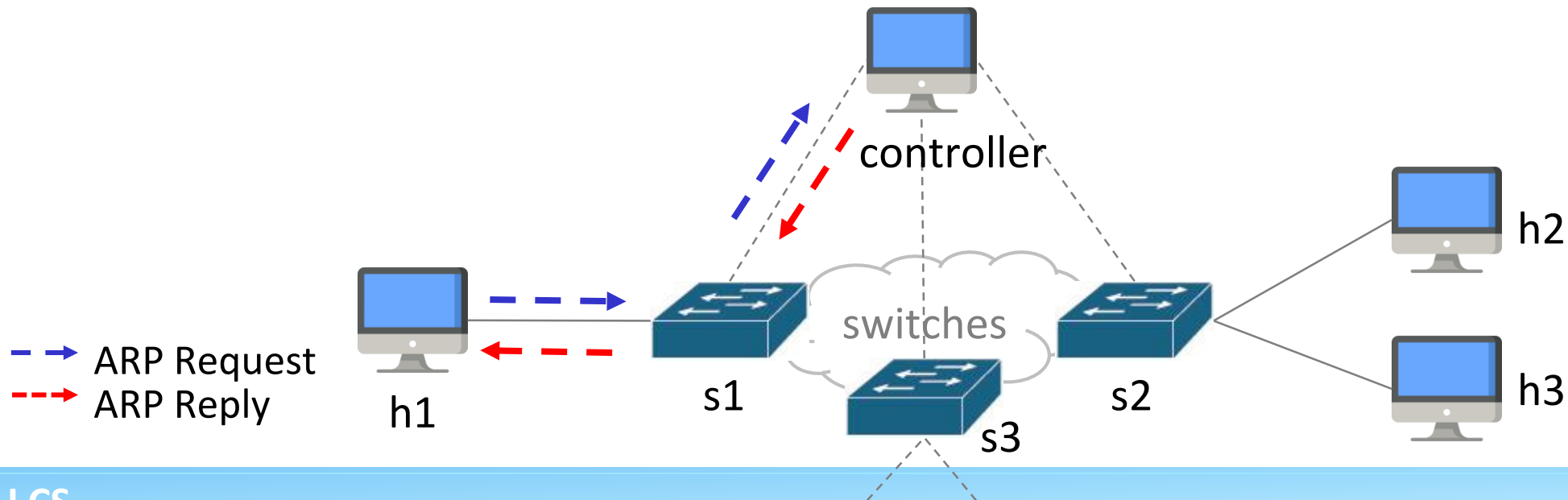ARP Request
ARP Reply

h1

s1

s2

s3
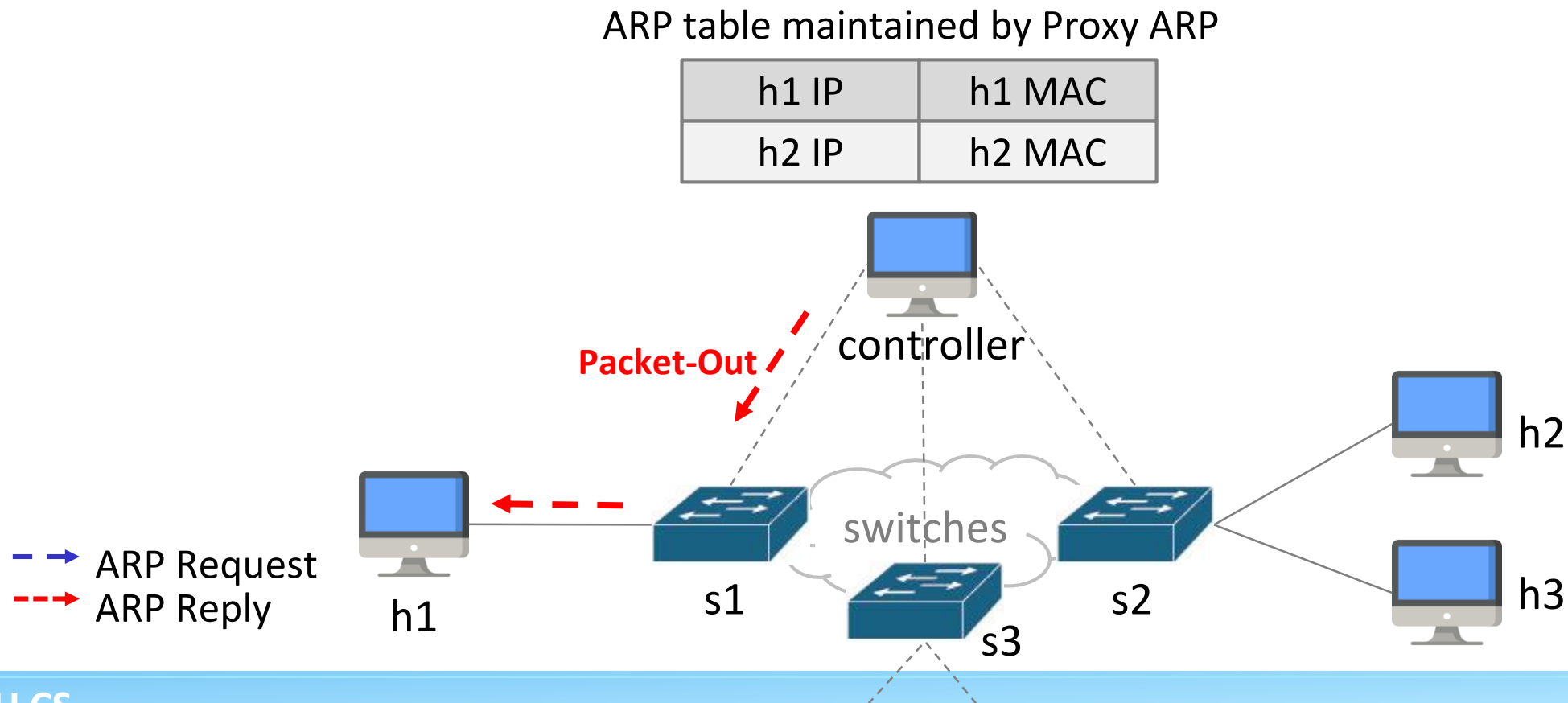
Subsequent ARP Request

- If h1 sends ARP request to h2 again
  - Since controller has learned the address mapping of h2, it just generates and Packet-Out the ARP reply to the ingress port

ARP table maintained by Proxy ARP

h1 IP	h1 MAC
h2 IP	h2 MAC

controller

switches

ARP Request
ARP Reply

h1

s1

s3

s2

h2

h3

Outline

- Introduction to ARP
  - What is ARP
  - ARP Request/Reply Format
- Proxy ARP
  - What is Proxy ARP
  - Workflow of Proxy ARP in SDN
- **Project 5 Requirements**
- References

Project 5 Descriptions

- In this project, you need to implement a Proxy ARP application
- **No** flow rule should be installed by your application
- Your implementation **should comply to workflow** introduced earlier
- Note:
  - The sender should **not** receive the packet-out when table miss
  - When table miss, only packet-out to **edge ports**

Test Your Application

- Once you activating your application and Mininet, executing **arping** in Mininet to check ARP functionality

```
mininet> h1 arping h2
```

- Correct result would look like:

```
mininet> h1 arping h2 -c 3
ARPING 10.0.0.2 from 10.0.0.1 h1-eth0
Unicast reply from 10.0.0.2 [D6:B5:82:B5:23:0E]  15.850ms
Unicast reply from 10.0.0.2 [D6:B5:82:B5:23:0E]  4.267ms
Unicast reply from 10.0.0.2 [D6:B5:82:B5:23:0E]  4.370ms
Sent 3 probes (1 broadcast(s))
Received 3 response(s)
mininet>
```

Restriction

- ONOS application activation
  - You are only allowed to activate **your *ProxyARP*** and the following ONOS applications:

```
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 nctu.winlab.ProxyArp               1.0.SNAPSHOT ONOS OSGi bundle archetype
```

Naming Requirement

- You should follow the Maven project naming format below
- Otherwise, your project will result in 10 points deduction
  - `<groupId>:` `nctu.winlab`
  - `<artifactId>:` `ProxyArp`
  - `<version>:` `<use default> (1.0-SNAPSHOT)`
  - `<Package>:` `nctu.winlab.ProxyArp`

Project 5 Scoring Criteria

- (10%) Project naming convention
- (30%) Print messages in following events:
  - ARP table miss

```
| 209 - nctu.winlab.ProxyArp - 1.0.0.SNAPSHOT | TABLE MISS. Send request to edge ports
```

  - ONOS receives ARP reply from host

```
| 209 - nctu.winlab.ProxyArp - 1.0.0.SNAPSHOT | RECV REPLY. Requested MAC = 06:4F:F1:84:A5:EA
```

  - ARP table hit

```
| 209 - nctu.winlab.ProxyArp - 1.0.0.SNAPSHOT | TABLE HIT.  Requested MAC = 06:4F:F1:84:A5:EA
```

- (60%) Work properly at least in tree (depth=3, fanout=3) topology
  - All hosts are able to **arping** to each other

Submission

- Files
  - All files of your application
- Submission
  - Upload ".zip" file to New e3
    - Named: **project5_<studentID>.zip**
  - Incorrect naming convention or format will result in 10 points deduction

Outline

- Introduction to ARP
  - What is ARP
  - ARP Request/Reply Format
- Proxy ARP
  - What is Proxy ARP
  - Workflow of Proxy ARP in SDN
- Project 5 Requirements
- References

References

- For fetching network topology status
  - http://api.onosproject.org/2.2.0/apidocs/org/onosproject/net/host/HostService.html
  - http://api.onosproject.org/2.2.0/apidocs/org/onosproject/net/edge/EdgePortService.html
- For packet manipulation
  - http://api.onosproject.org/2.2.0/apidocs/org/onlab/packet/Ethernet.html
  - http://api.onosproject.org/2.2.0/apidocs/org/onlab/packet/ARP.html