Final Project

VLAN-based Segment Routing

Deadline: 2022/01/12 (WED) 23:59

Outline

- Review of Labs
- Segment Routing
  - IP Routing
  - Workflow of Segment Routing
  - Node Segment
- Final Project
  - Overview
  - Workflow
  - Requirements

Outline

- Review of Labs
- Segment Routing
  - IP Routing
  - Workflow of Segment Routing
  - Node Segment
- Final Project
  - Overview
  - Workflow
  - Requirements

Review of Labs

- Lab 4 - Unicast DHCP Application
  - Installing flow rules
  - Routing packets with global view of network
  - Configuring controller
- Lab 5 – Proxy ARP
  - Constructing packets and sending directly to switches

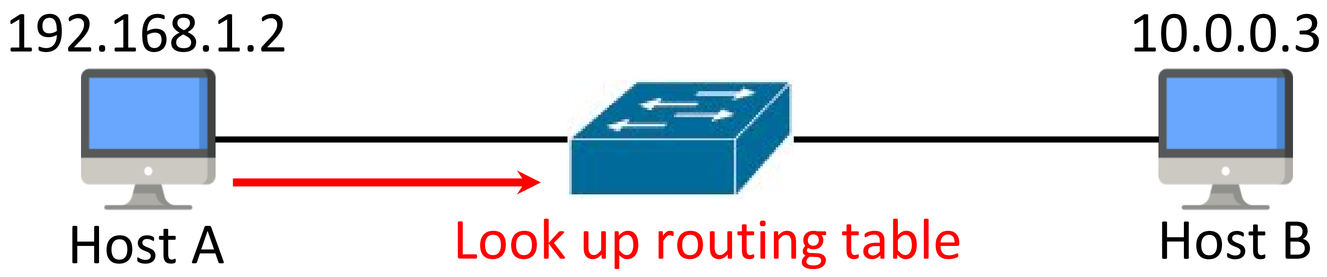
- Note: All of these labs would be used in final project

Outline

- Review of Labs
- Segment Routing
  - IP Routing
  - Workflow of Segment Routing
  - Node Segment
- Final Project
  - Overview
  - Workflow
  - Requirements

IP Routing

- Network devices route packets with IP address
  - Maintain routing information on each device
  - Look up IP table when packets arrive
- Determine paths while forwarding packets

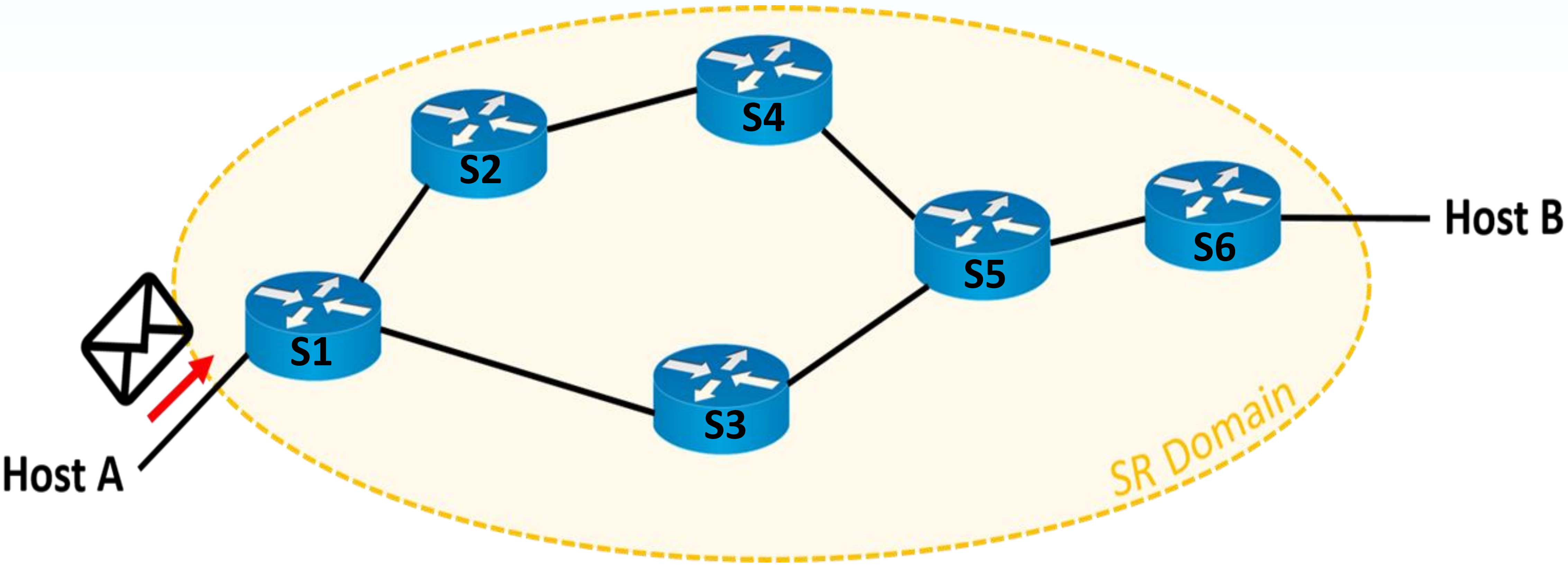


Segment Routing (SR)

- Use label (segment) switching instead of IP address
- Sender (or ingress node) of packets specifies routes of packets
- Features:
  - Sender
    - Choose a path
    - Encode it in the packet header as an ordered list of segments
  - The rest of network devices
    - Execute the encoded instructions (labels)
      - i.e., forwarding

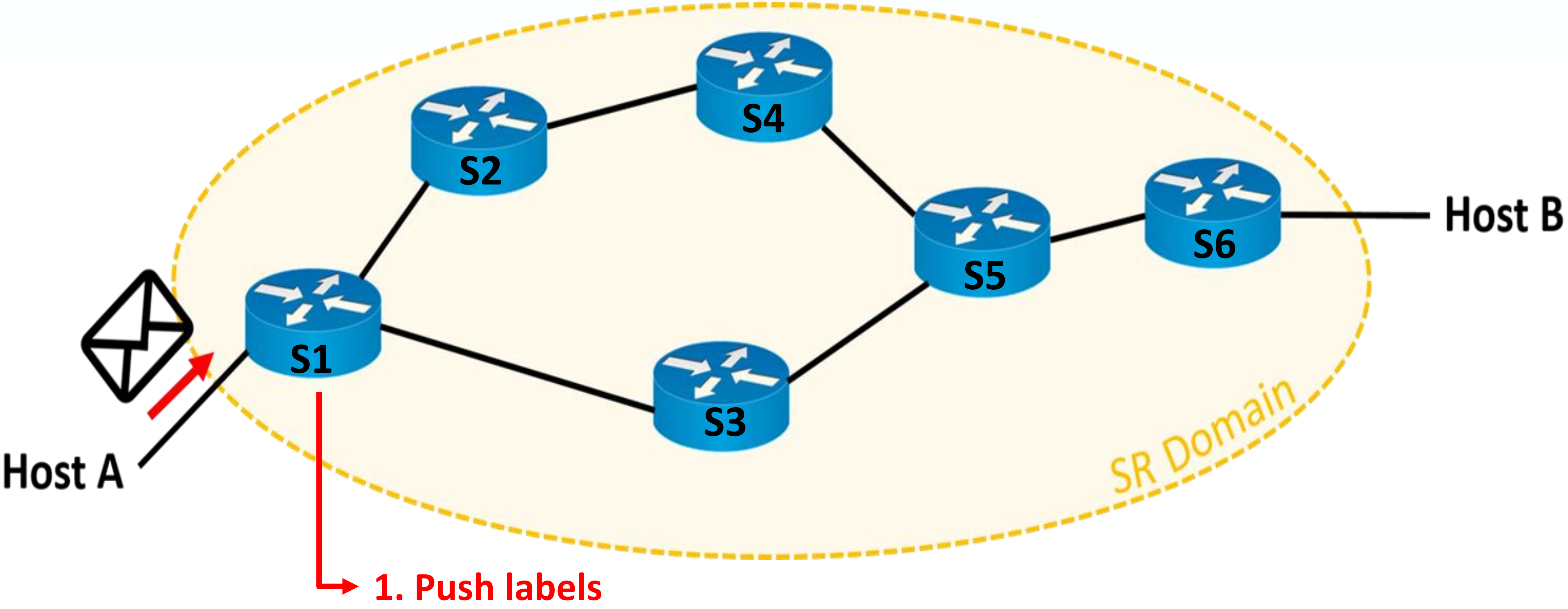
Segment Routing – Workflow (1/9)

- Host A sends packet to Host B



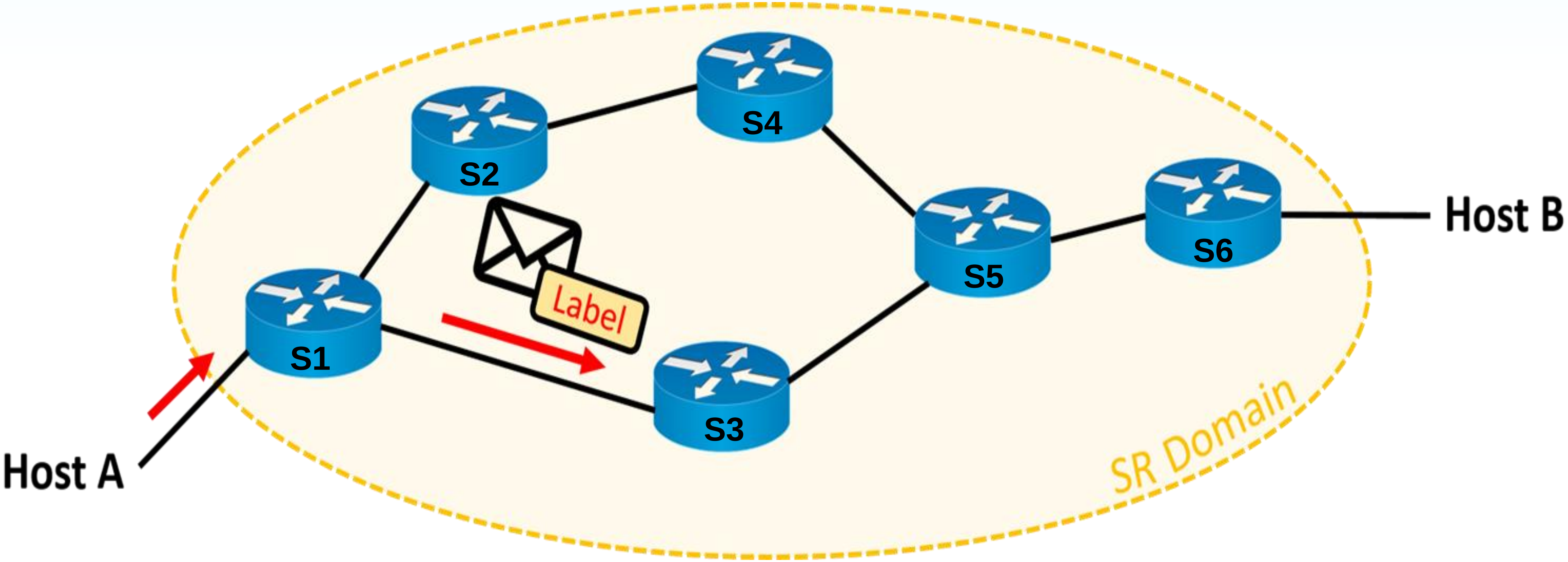
Segment Routing – Workflow (2/9)

- The edge switch S1 pushes the label of destination device S6



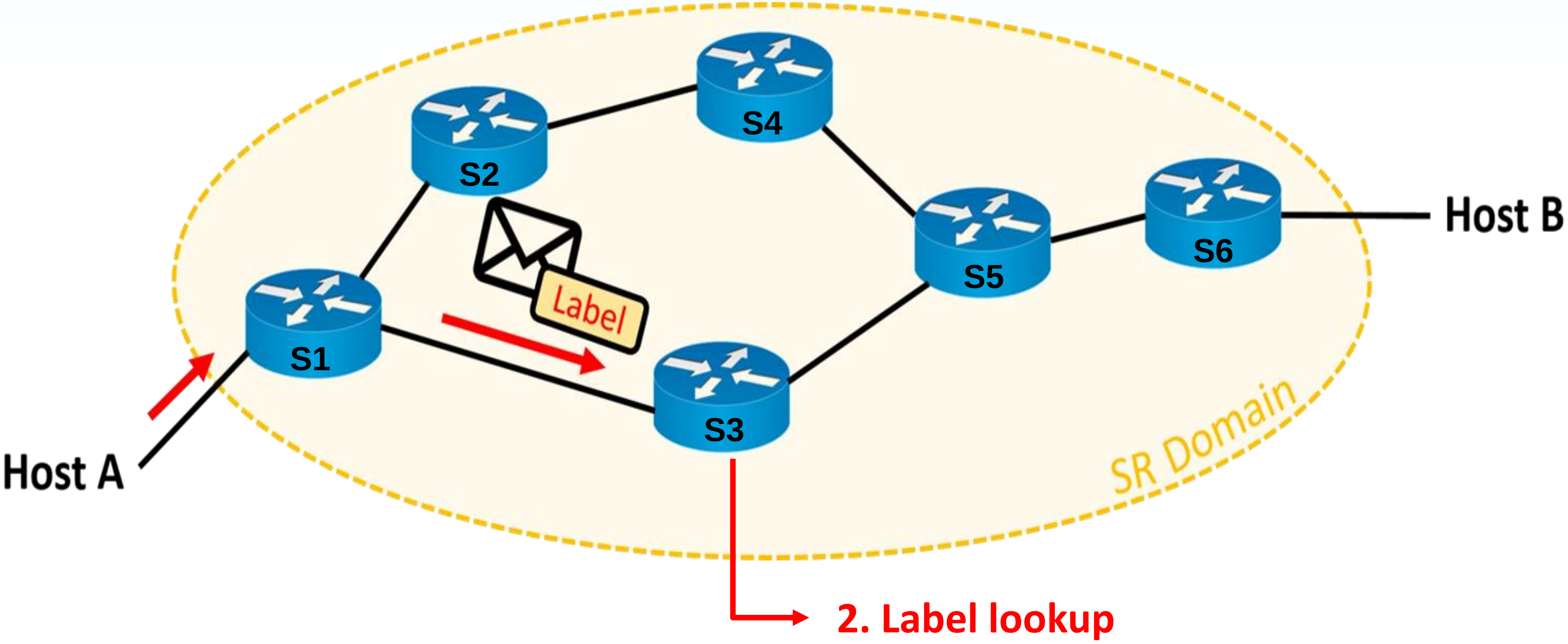
Segment Routing – Workflow (3/9)

- The edge switch S1 forwards packet with label



Segment Routing – Workflow (4/9)

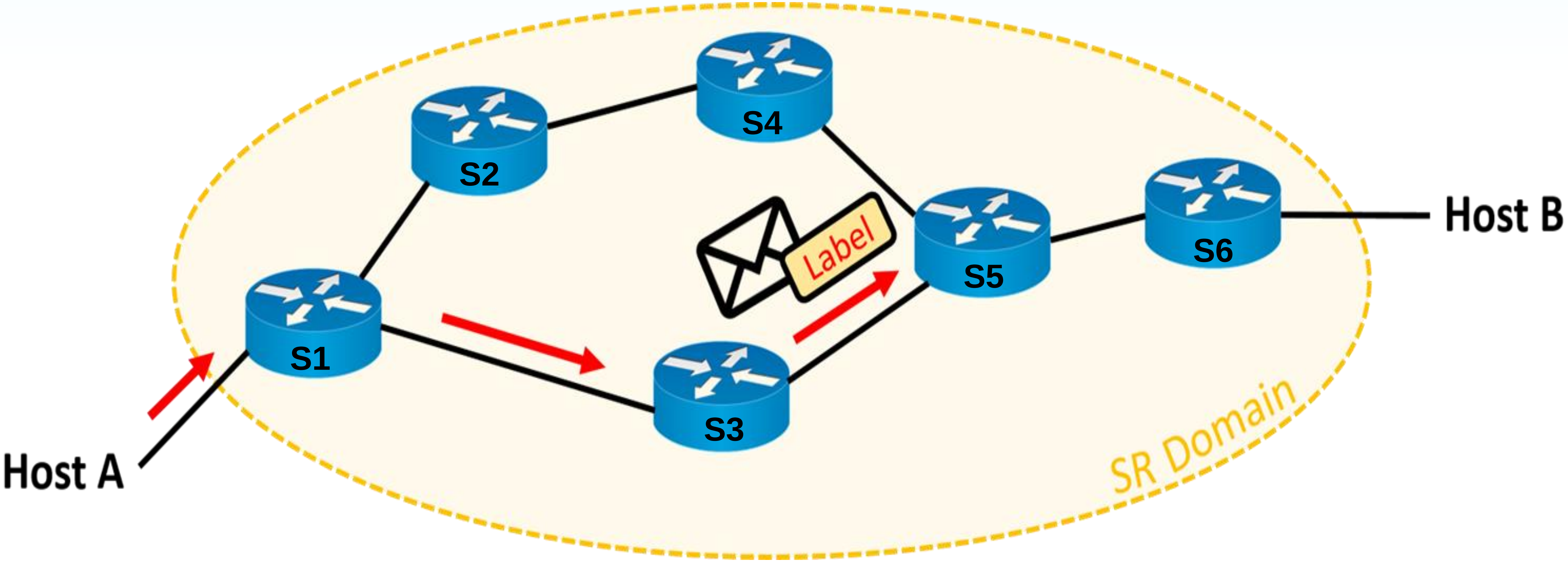
- Switch S3 receives packet with label and lookups flow table



2. Label lookup

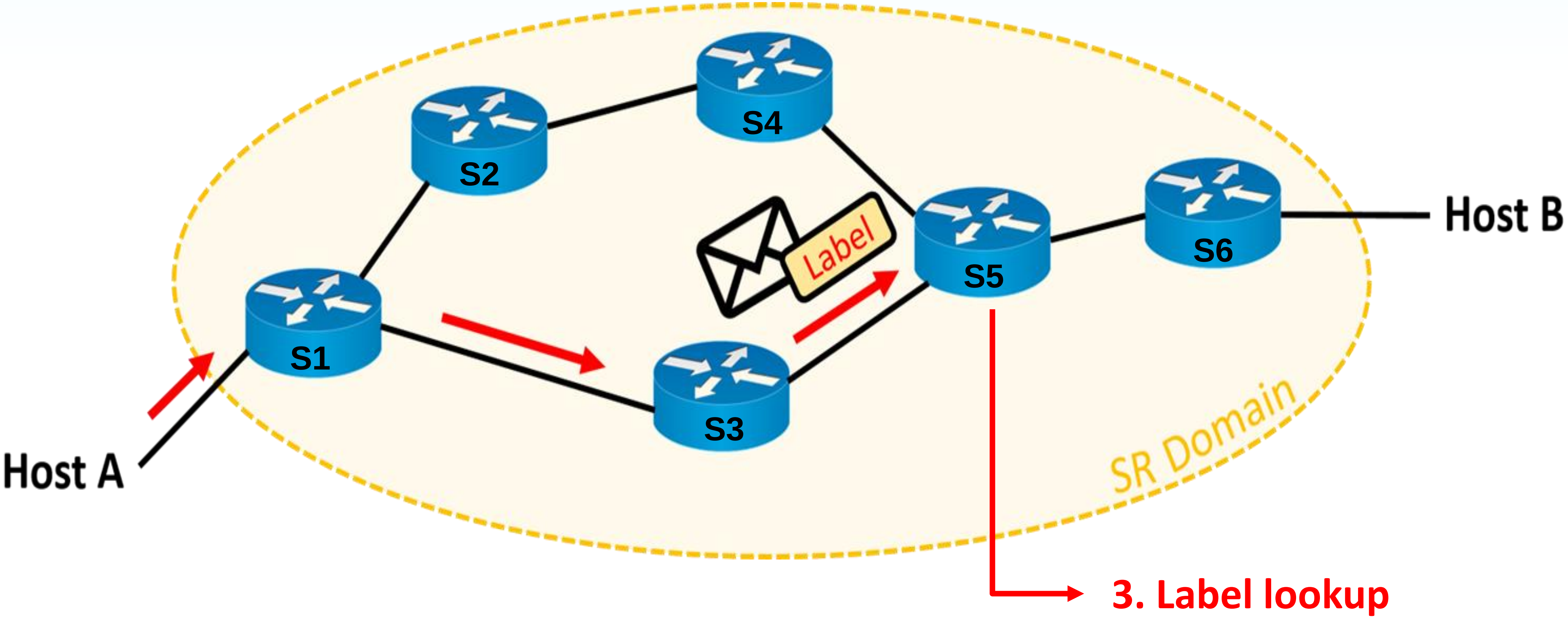
Segment Routing – Workflow (5/9)

- Switch S3 forwards packet with label



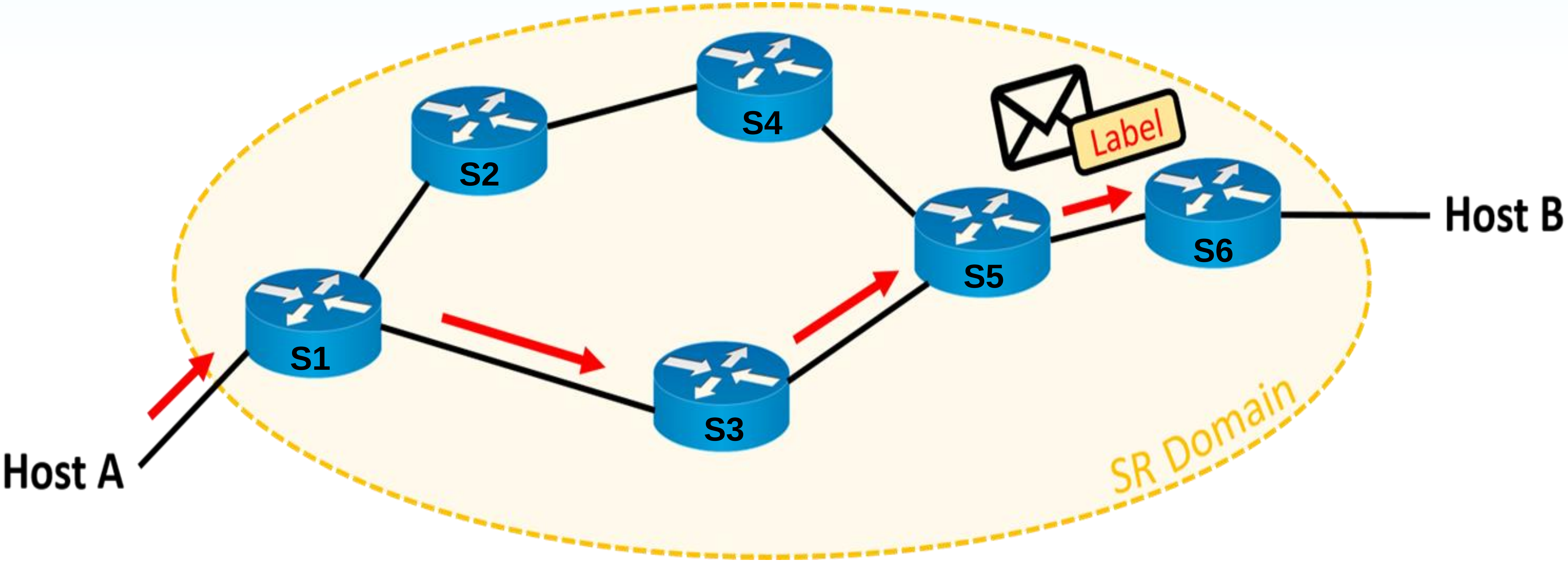
Segment Routing – Workflow (6/9)

- Switch S5 receives packet with label and lookups flow table



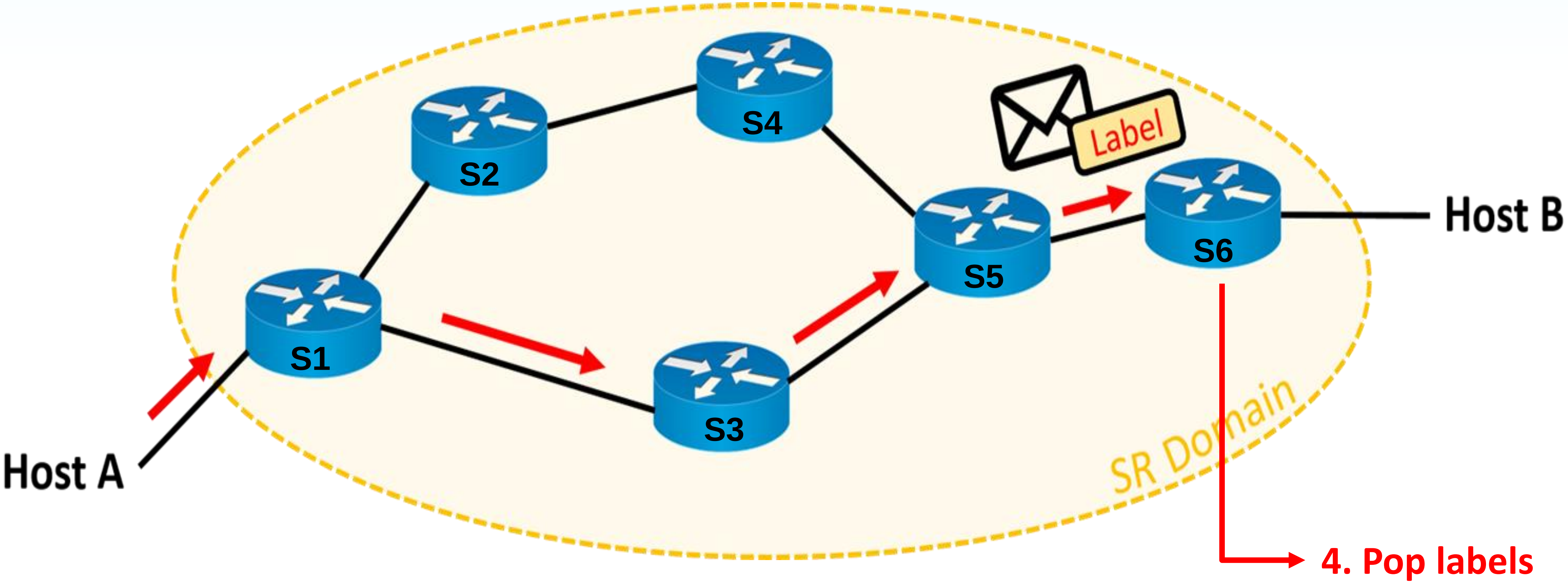
Segment Routing – Workflow (7/9)

- Switch S5 forwards packet with label



Segment Routing – Workflow (8/9)

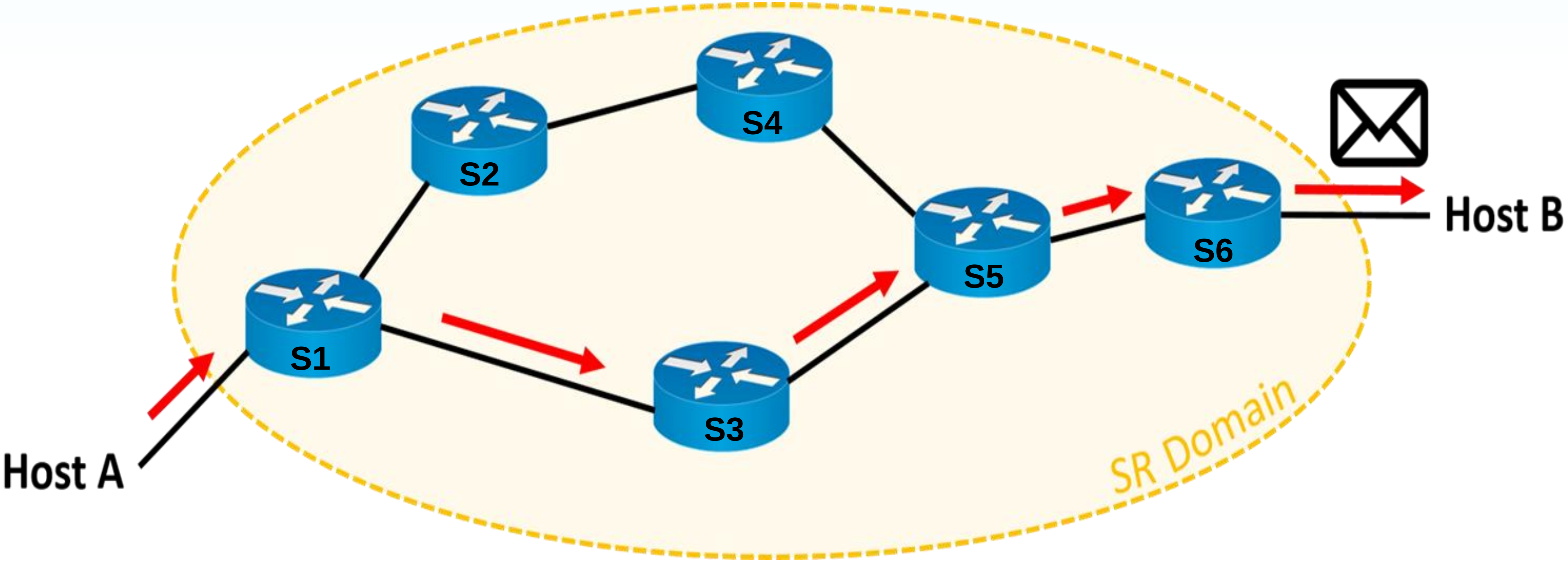
- Switch S6 receives packet with label and pops label



4. Pop labels

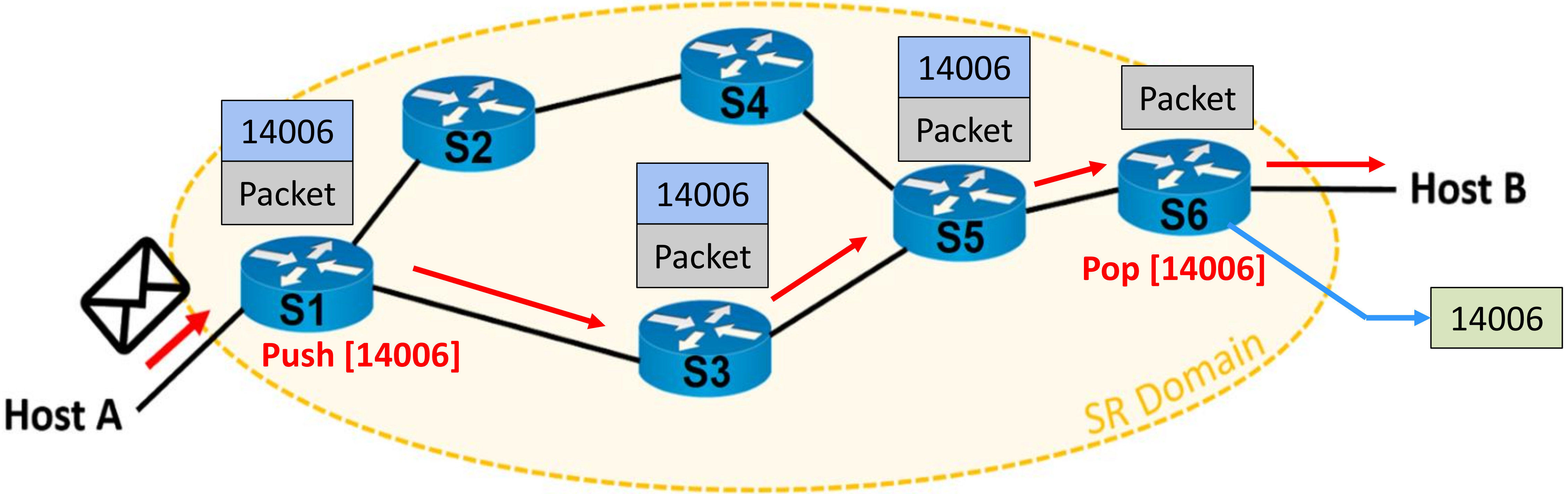
Segment Routing – Workflow (9/9)

- Switch S6 forwards the original packet to Host B



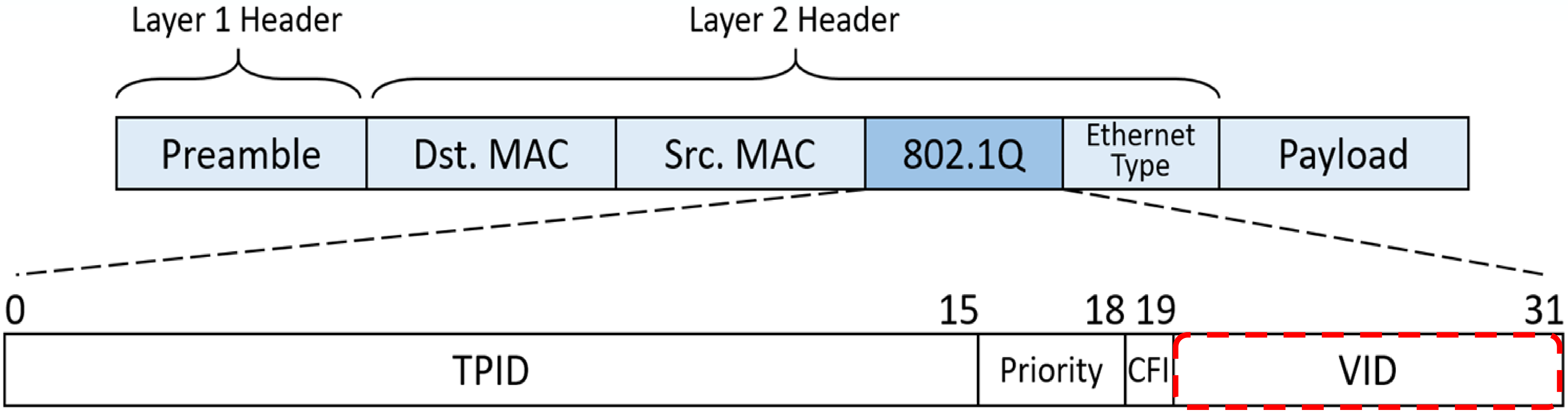
Node Segment

- Node segment ID is globally unique within a SR domain
- Typically multi-hop
  - Shortest-path first (SPF) route to designated node



VLAN

- Segment routing uses labels to route packets
- We will use **VID** field in VLAN (802.1Q) header as label



TPID: Tag protocol identifier (0x8100)

CFI: Canonical Format Indicator

VID: VLAN Identifier

Outline

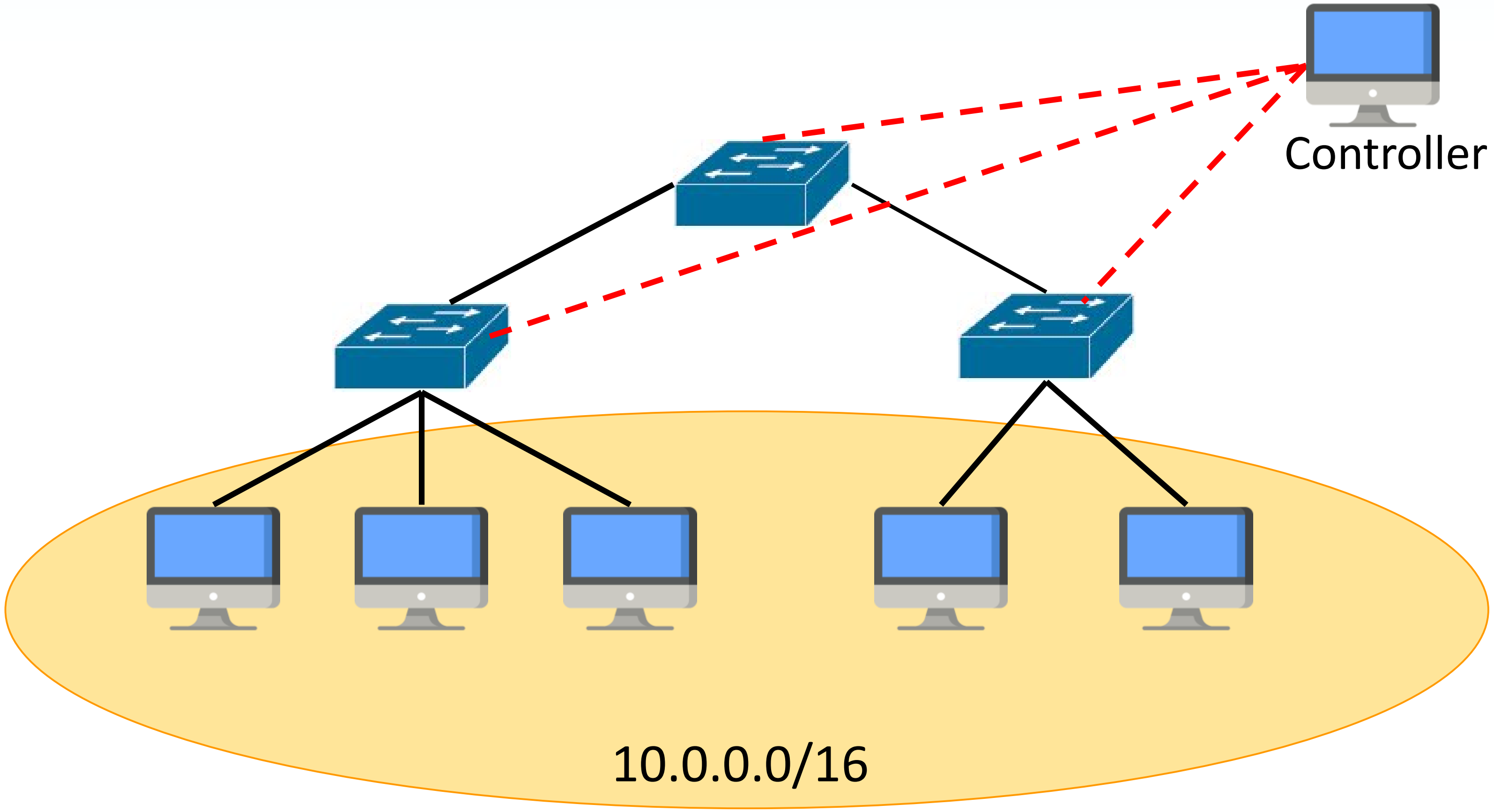
- Review of Labs
- Segment Routing
  - IP Routing
  - Workflow of Segment Routing
  - Node Segment
- Final Project
  - Overview
  - Workflow
  - Requirements

Overview

- You need to implement an VLAN-based segment routing app
  - Configure network
    - DHCP server location,
    - segment ID for each switch, and
    - subnet attached to edge switches
  - Compute path to each edge switches
    - Install flow rules to forward packets

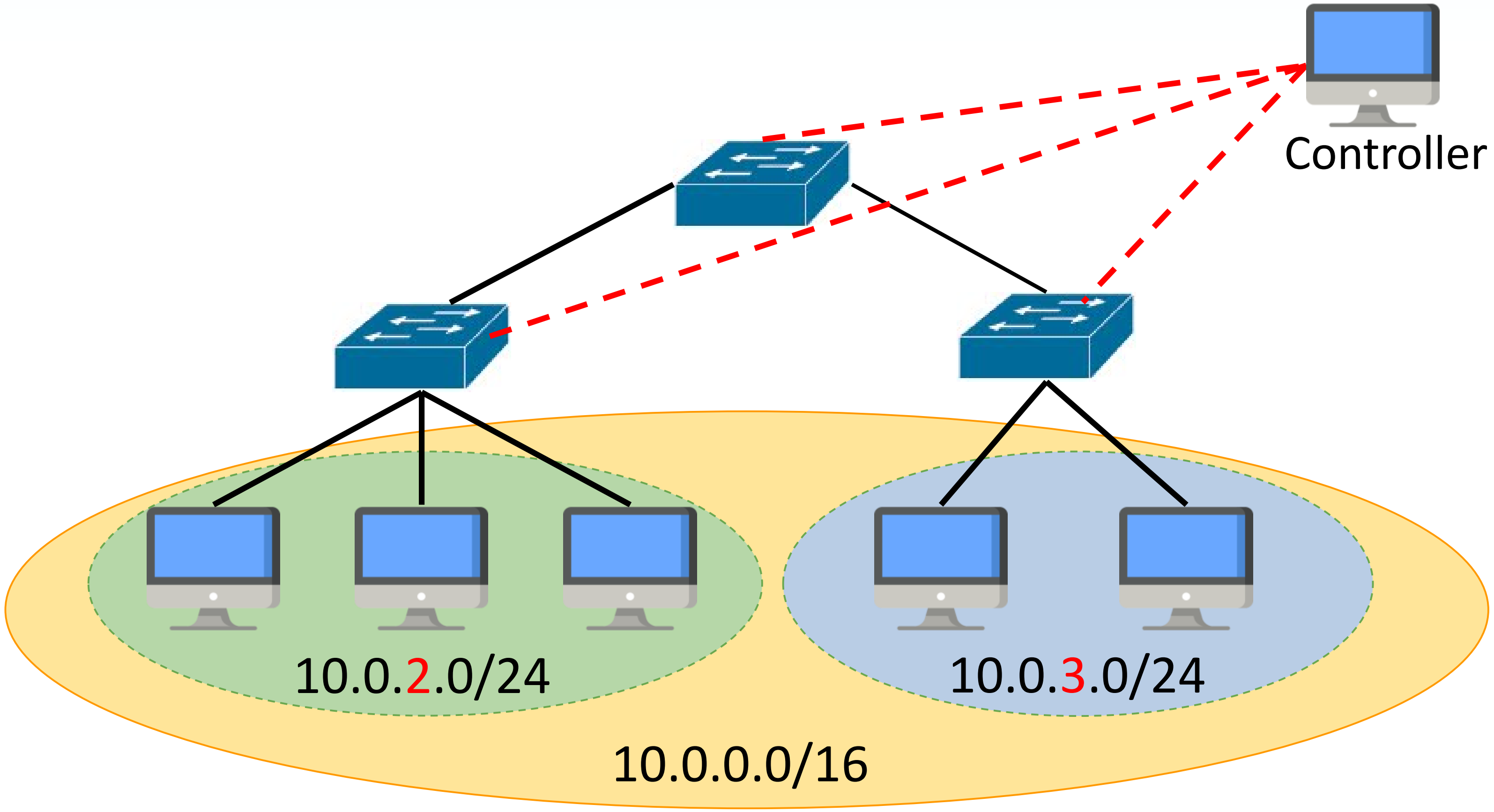
Configure Network – in Mininet

- Mininet topology
  - Hosts are configured under the same subnet



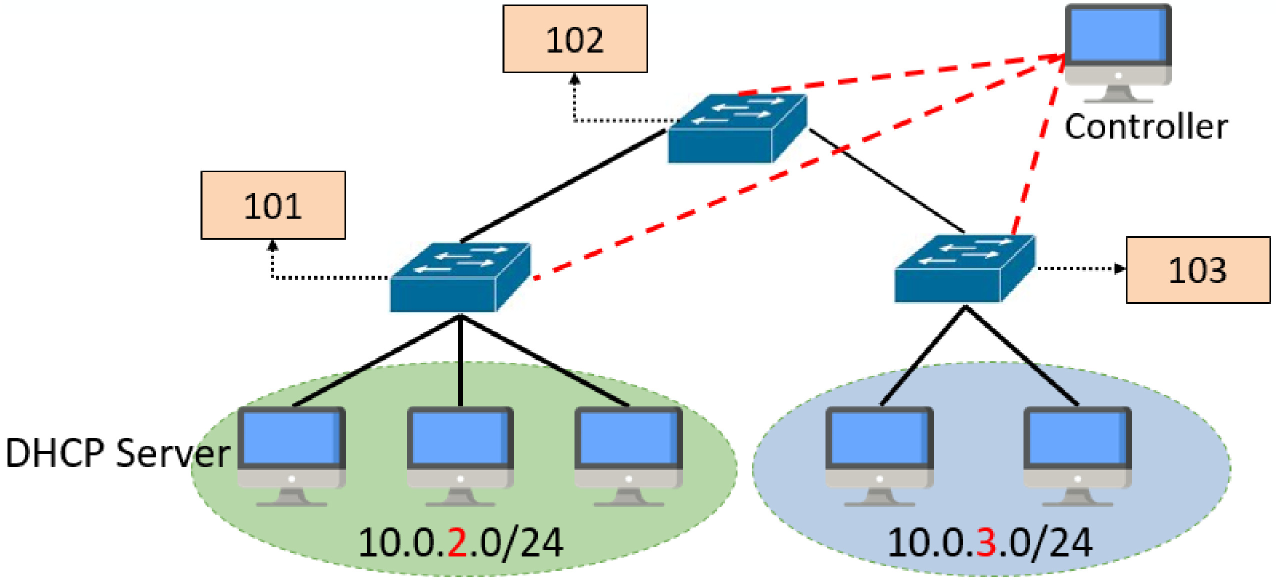
Configure Network – for Controller

- Controller view
  - Hosts are configured under the **different** subnet



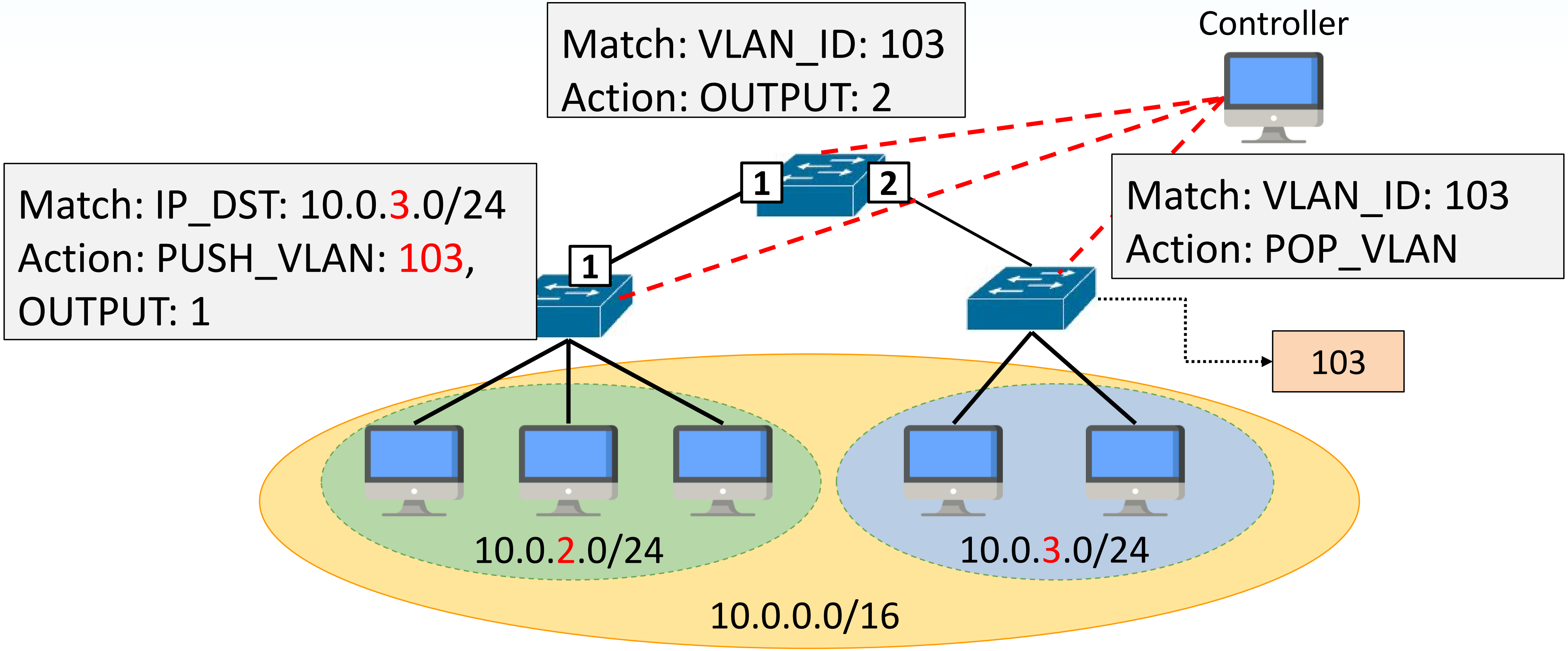
Configure Network – Upload Configuration

- Upload configuration to controller
  - DHCP server location
  - Segment ID for switch (node segment)
  - IP subnet on edge switch
  - Other configuration as you wish
    - E.g., indication of edge switch (to make implementation easier)



Compute Path and Install Flow Rules

- Compute a path for each pair of edge nodes
- Install flow rules for all paths



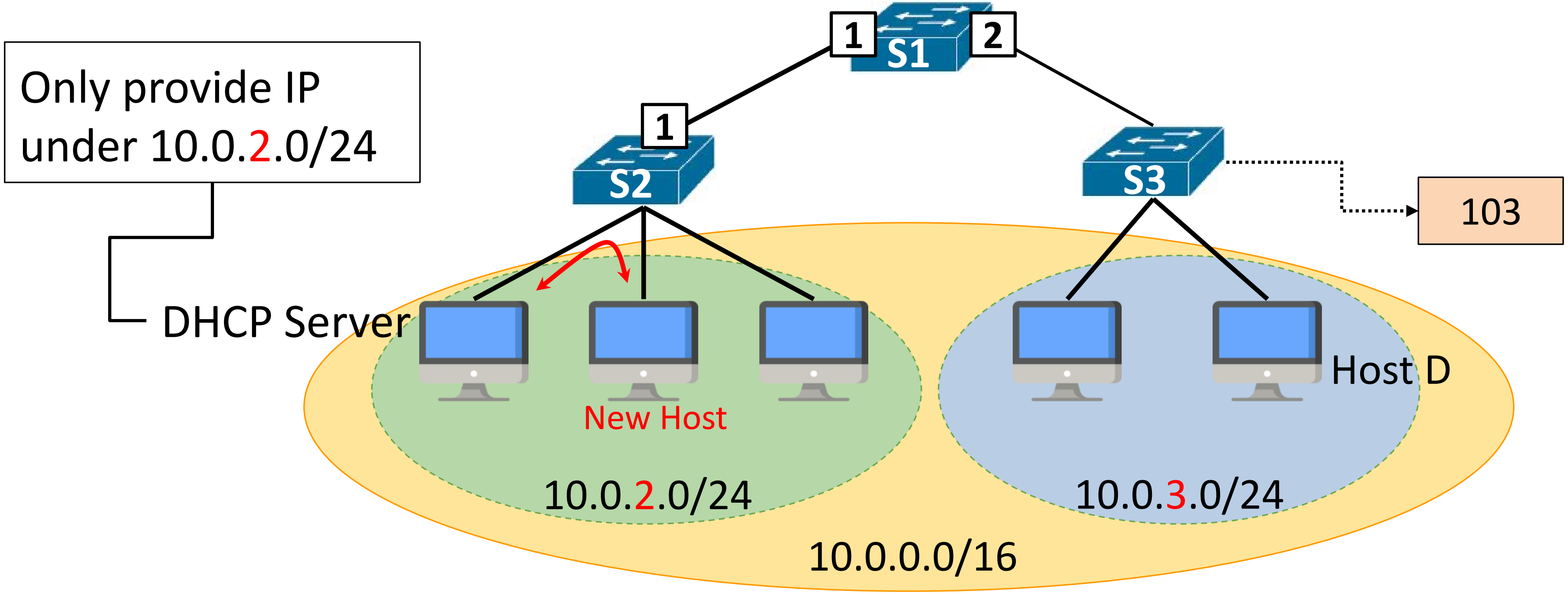
Match: VLAN_ID: 103
Action: OUTPUT: 2

Match: IP_DST: 10.0.3.0/24
Action: PUSH_VLAN: 103,
OUTPUT: 1

Match: VLAN_ID: 103
Action: POP_VLAN

Workflow (1/6) – Integration with Unicast DHCP

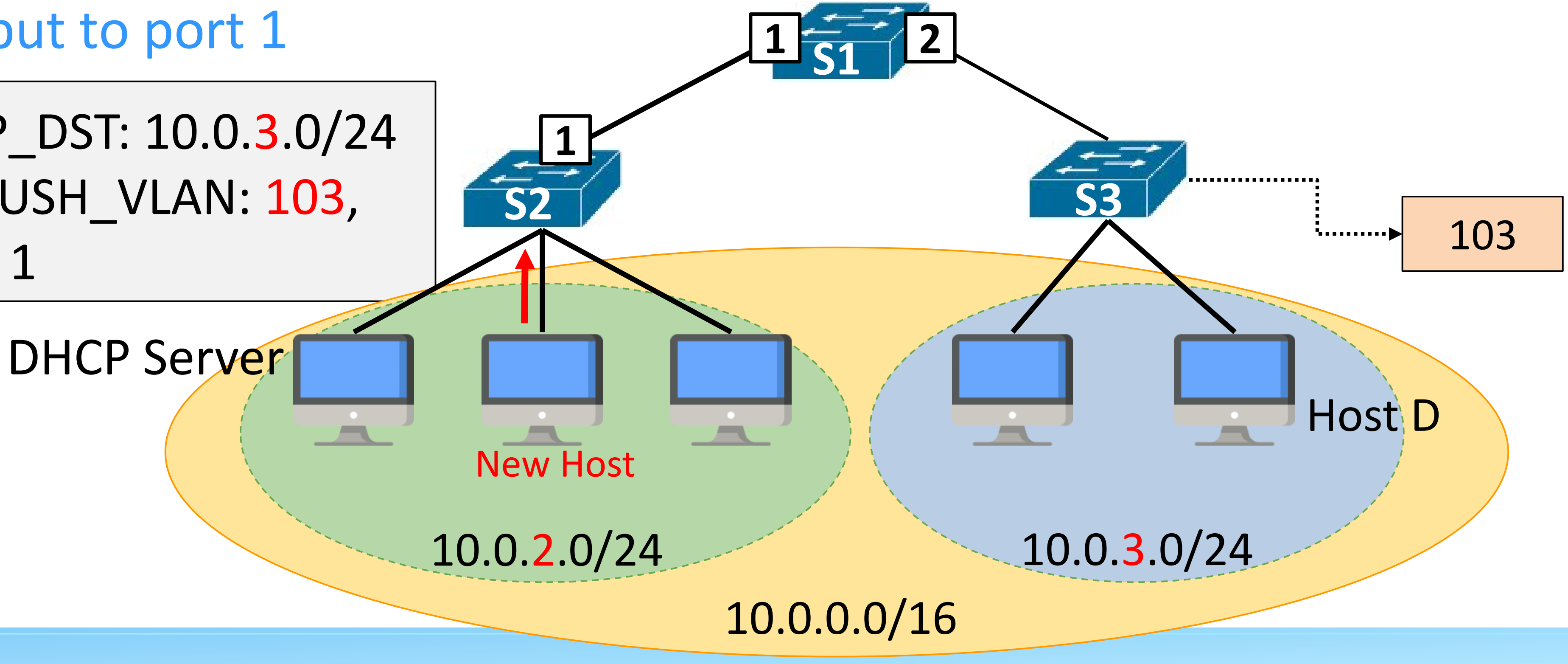
- New host could request IP address from DHCP server
  - This is what you done in lab 4



Workflow (2/6) – Host to Host Communication

- Assume new host sends packet to Host D
- Edge Switch S2:
  - Match subnet of destination address
  - Push VLAN tag to packet according to destination edge node
  - Then output to port 1

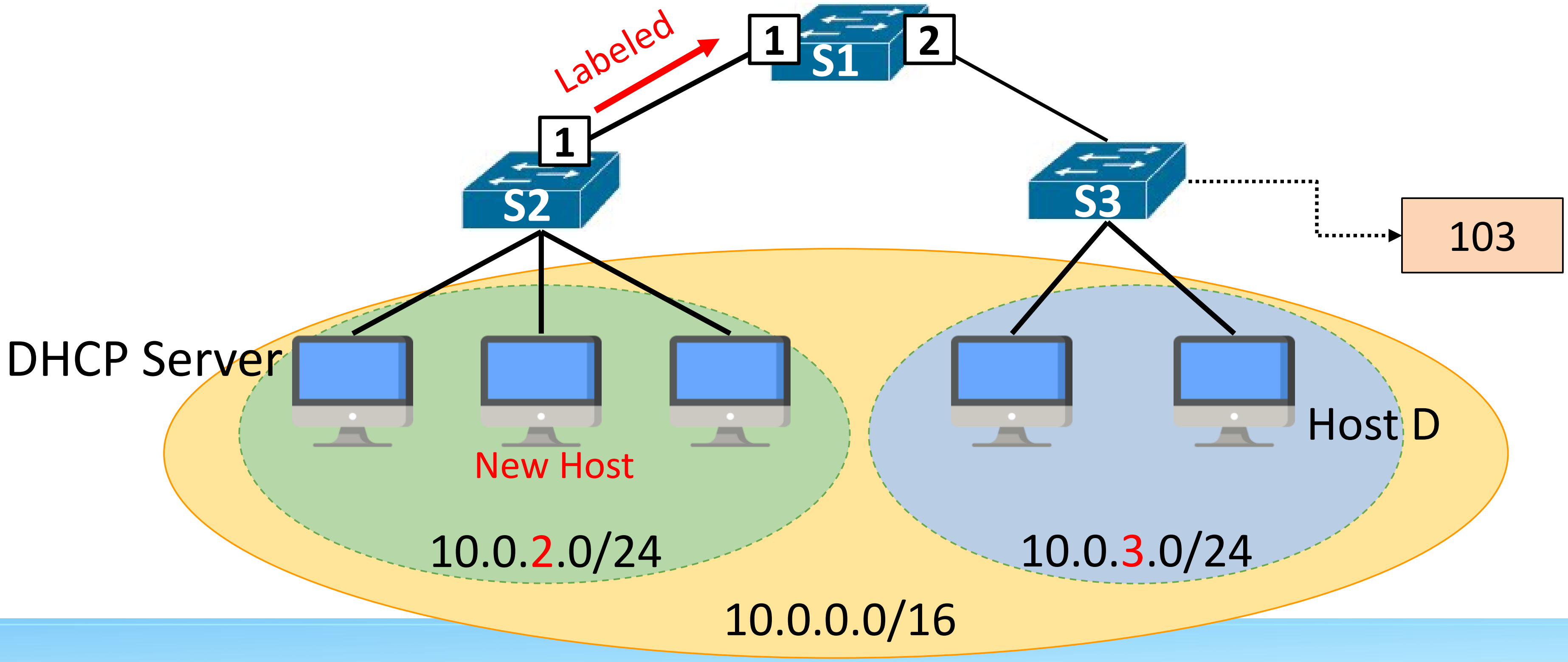
Match: IP_DST: 10.0.3.0/24
Action: PUSH_VLAN: 103,
OUTPUT: 1



Workflow (3/6) – Label Lookup

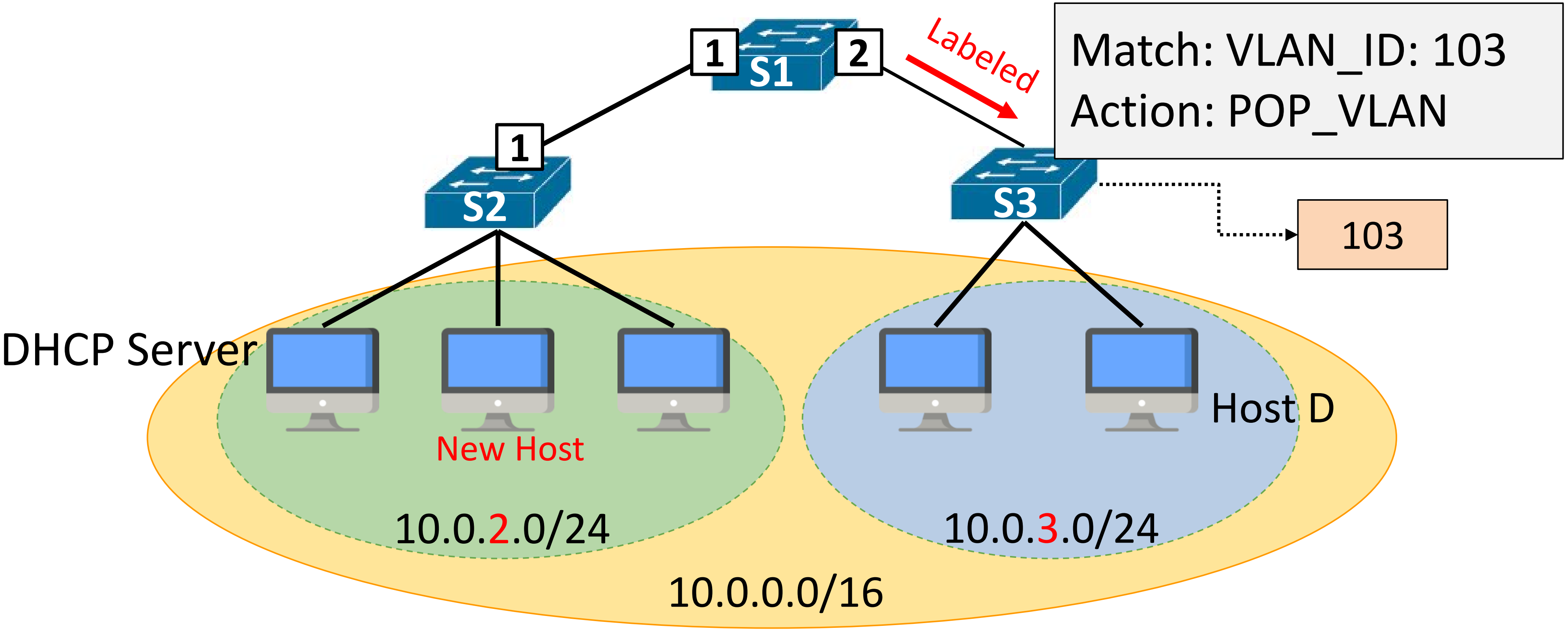
- S1 receives VLAN tagged packet
  - Match VLAN tag and forward tagged packet via port 2

Match: VLAN_ID: 103
Action: OUTPUT: 2



Workflow (4/6) – Label Popping

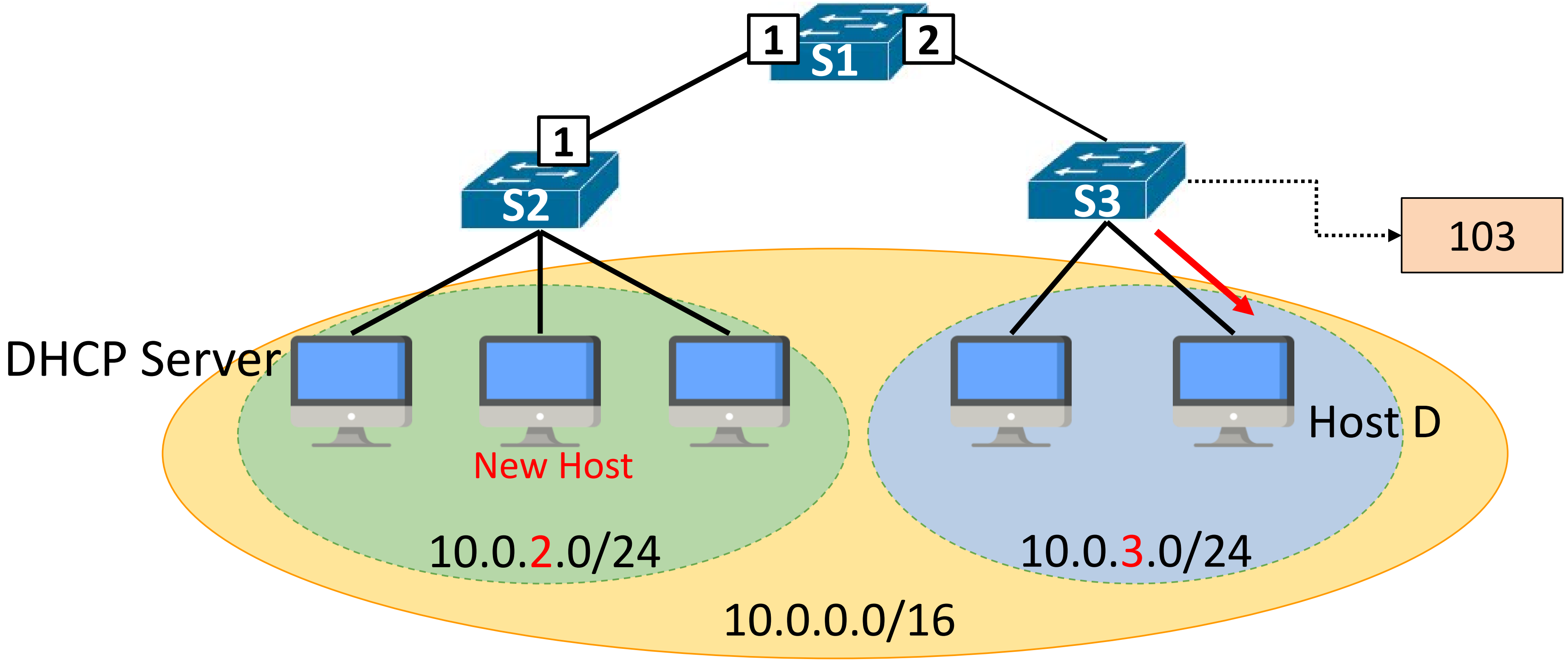
- When destination edge switch S3 receives tagged packet
  - Pop VLAN tag
  - Forward original (untagged) packet to Host D (next slide)



Match: VLAN_ID: 103
Action: POP_VLAN

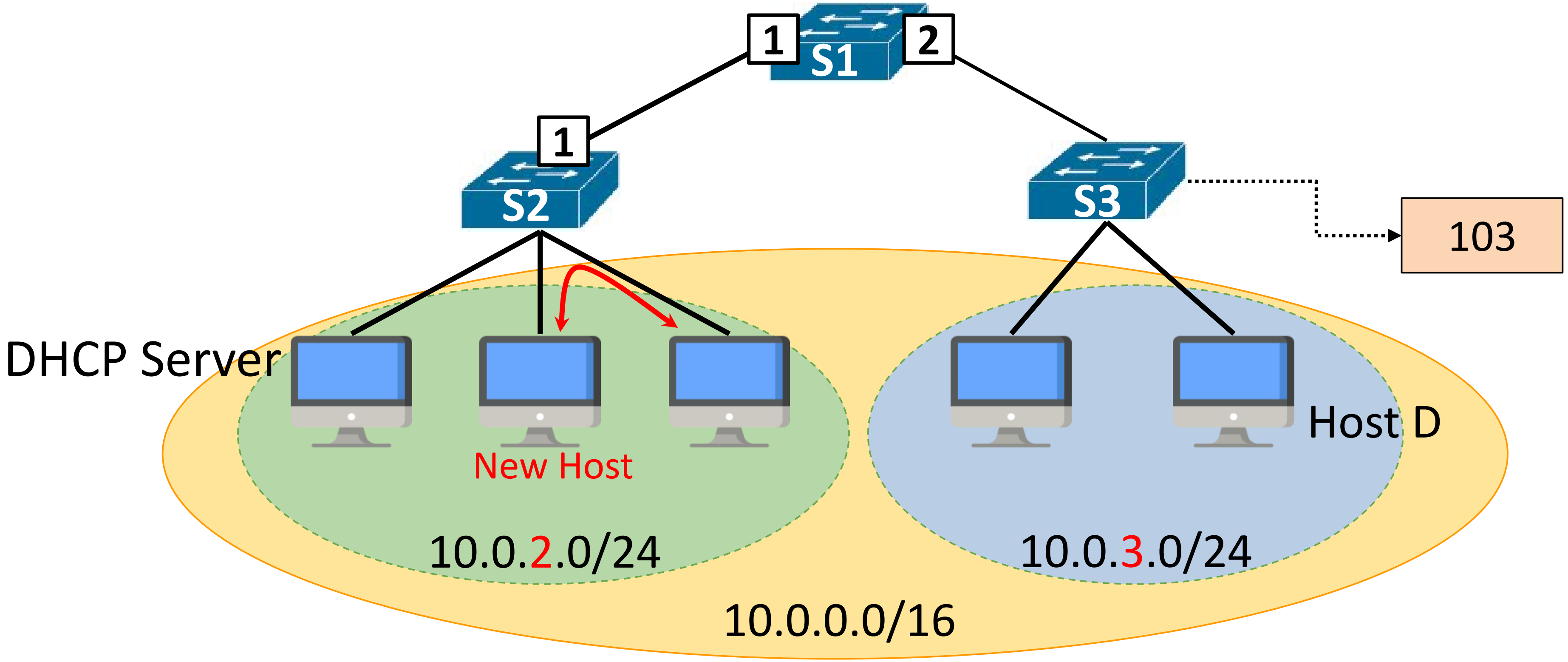
Workflow (5/6) – IP/MAC Forwarding

- After label popping, S3 forwards original packet by matching Layer 2 or L3 address



Workflow (6/6) – Intra-device Forwarding

- Intra-device packet could be forwarded by matching Layer 2 or L3 address directly



Requirements

- The following three applications should be activated
  - DHCP Unicast (lab 4)
  - Proxy ARP (lab 5)
  - VLAN-based Segment Routing (this project)
- You **should not activate** any other application except **OpenFlow-related applications** on ONOS
- Install all flow rules when controller receives configuration
- Flow rules for forwarding packets must **match VLAN tag**
  - Except for intra-subnet forwarding

Issues

- Controller may not be able to install correct flow rules when receiving configuration
  - Since controller does not know hosts at the beginning
- First solution
  - Send packets to let controller know host information before uploading configuration
- Second solution
  - Add host information in configuration
- You can figure out other solutions on your own

Naming Requirement

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

Scoring Criteria

- Report: Previous Labs Parts
  - DHCP Unicast and Proxy ARP (5%)
- Report: Final Project Part
  - Activated applications (10%)
    - Only activate DHCP unicast, Proxy ARP, vlanbasedsr, and OpenFlow-related applications
  - Flow rules (25%)
    - Must use VLAN tag to forward packets
  - Connectivity (10%)
    - Hosts under different subnets can send labeled packets to each other
- Demo (50%)
  - TA will produce a different topology to test your app

Submission

- Files:
  - All files in Vlan-based Segment Routing app
  - No need to submit DHCP Unicast and Proxy ARP app
- Submit:
  - Upload a ".zip" file to e3
    - Named: final_<studentID>.zip
  - Incorrect naming convention or format will not be scored