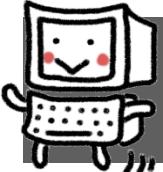


# Final Project

*VLAN-based Segment Routing*

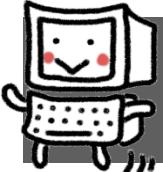
Date: 2019/06/06 (Thu.)

Deadline: 2019/06/23 (Sun.)



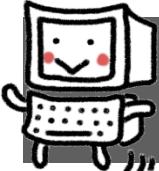
# Outline

- ❑ Review of Projects
- ❑ Segment Routing
- ❑ Implementation
- ❑ Final Project Requirements
- ❑ Grading
- ❑ Hints



# Outline

- Review of Projects
- Segment Routing
- Implementation
- Final Project Requirements
- Grading
- Hints



# Review of Projects

## □ Project 4 – Path Service

- Installing flow rules
- Routing packets with global view of network

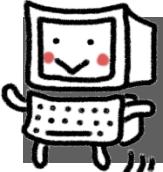
## □ Project 6 – DHCP Unicast

- Configuring controller

## □ Project 7 – Proxy ARP

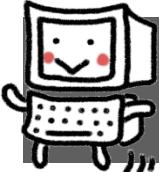
- Constructing packets and sending directly to switches

*Note: All of these projects would be used in final project*



# Outline

- ❑ Review of Projects
- ❑ Segment Routing
- ❑ Implementation
- ❑ Final Project Requirements
- ❑ Grading
- ❑ Hints

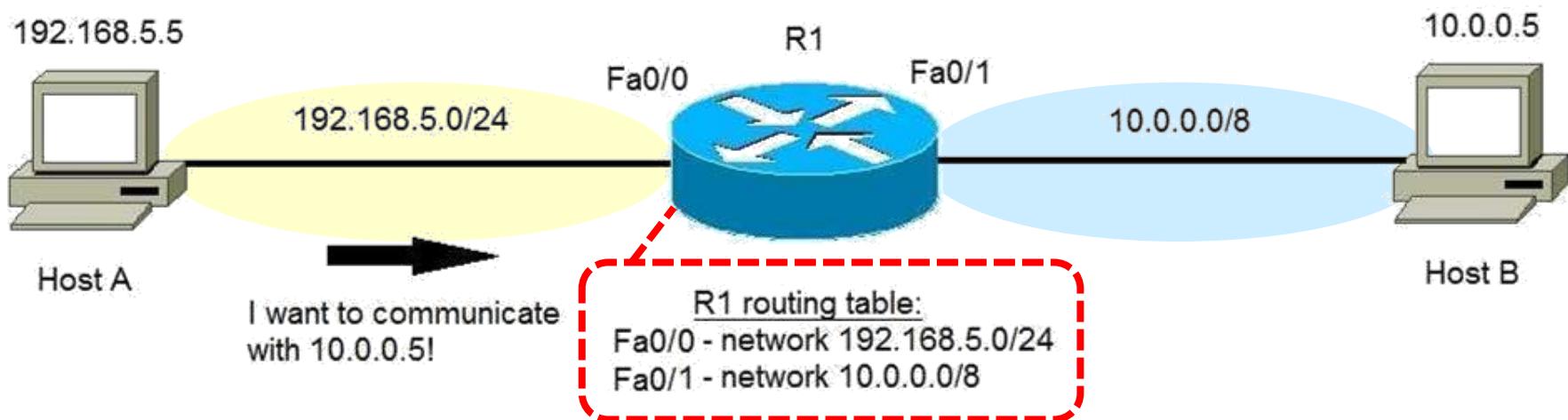


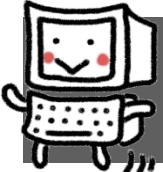
# IP Routing

- Network devices route packets with IP address

- Maintain routing information on each device
- Look up IP table when packets arrive

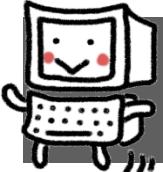
- Paths are determined while packets are forwarding





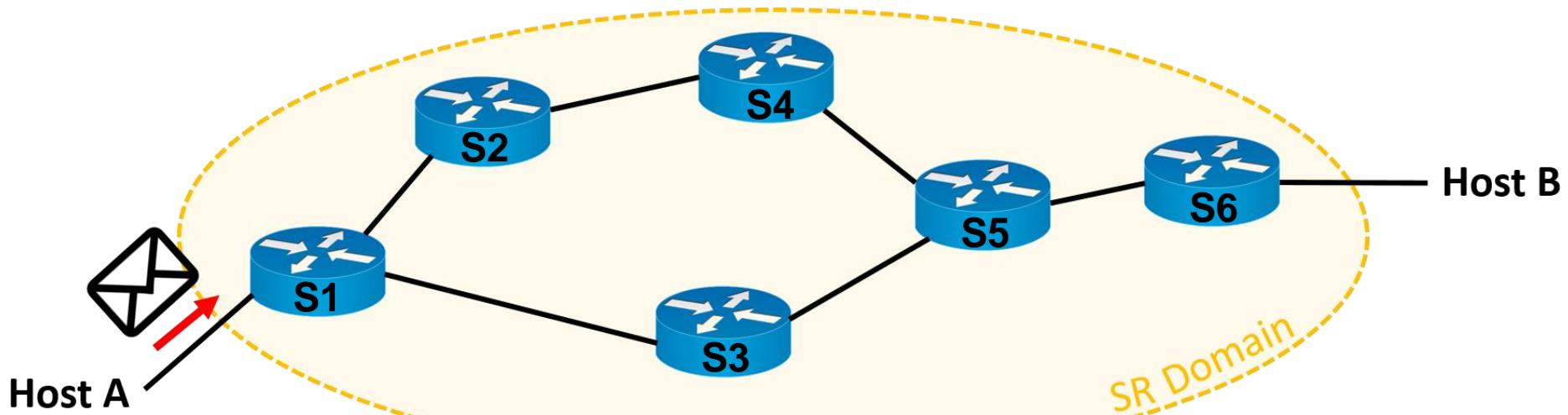
# Segment Routing (SR)

- Addressed by IETF SPRING (Source Packet Routing In NetworkinG)
  - RFC 8402
- Use label (segment) switching instead of IP address
  - MPLS
  - IPv6

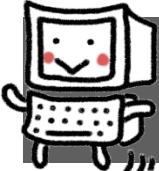


# Segment Routing (I)

- Host A sends packet to Host B

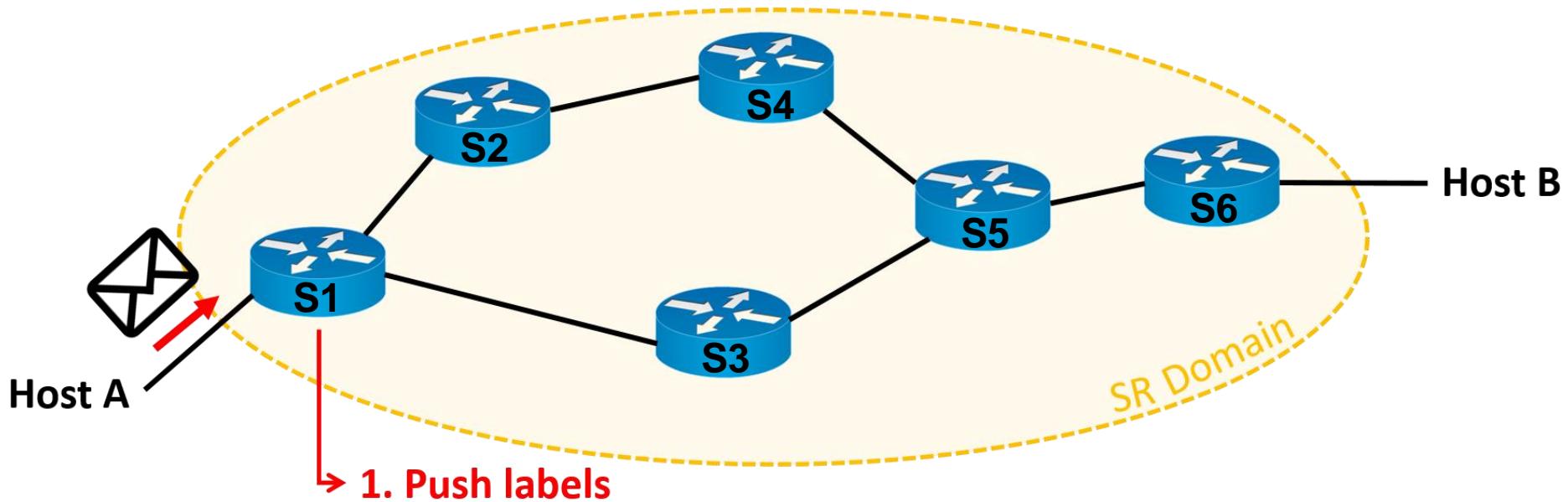


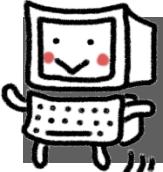
*Note: Segment Routing implemented in final project is simplified as the following procedure*



# Segment Routing (II)

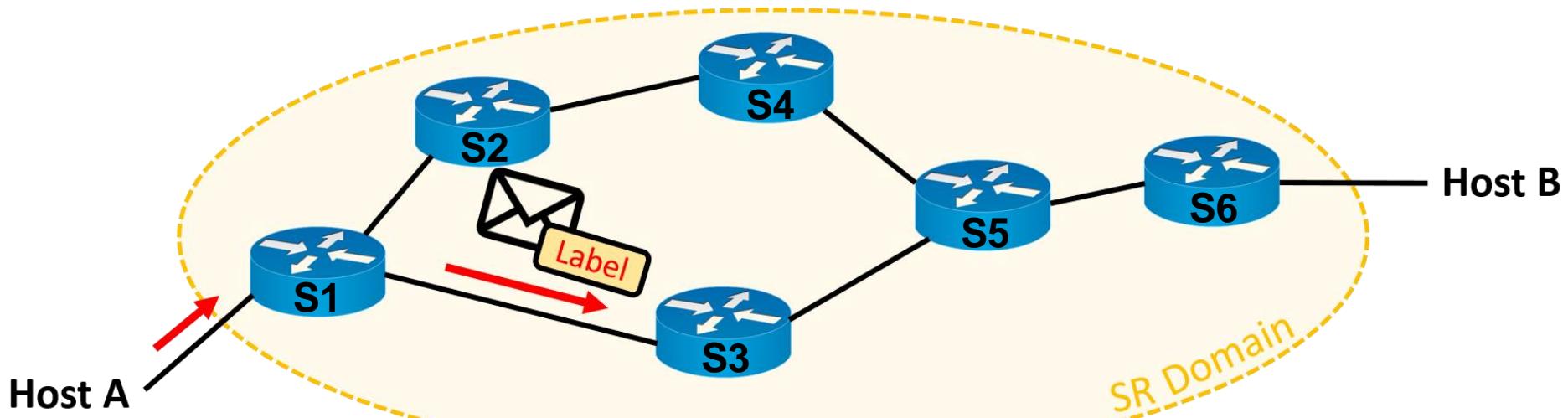
- The edge switch S1 pushes label of destination device S6

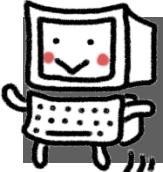




# Segment Routing (III)

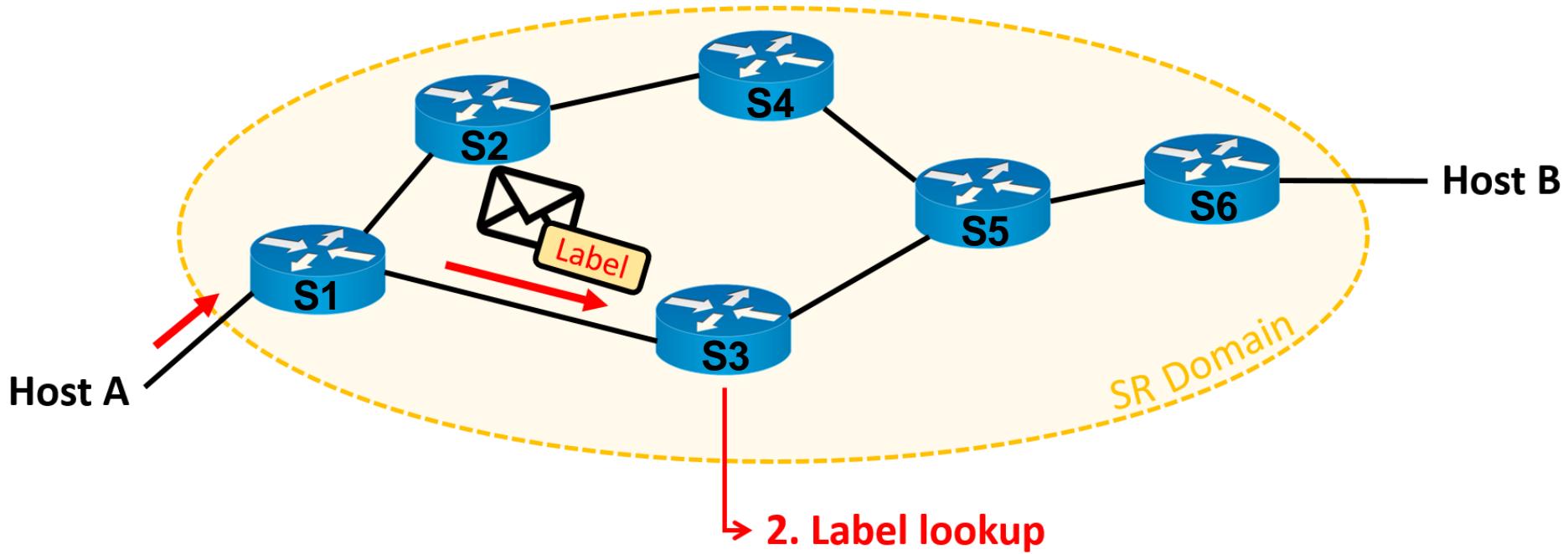
- The edge switch S1 forwards packet with label

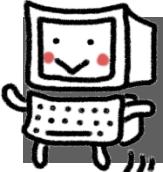




# Segment Routing (IV)

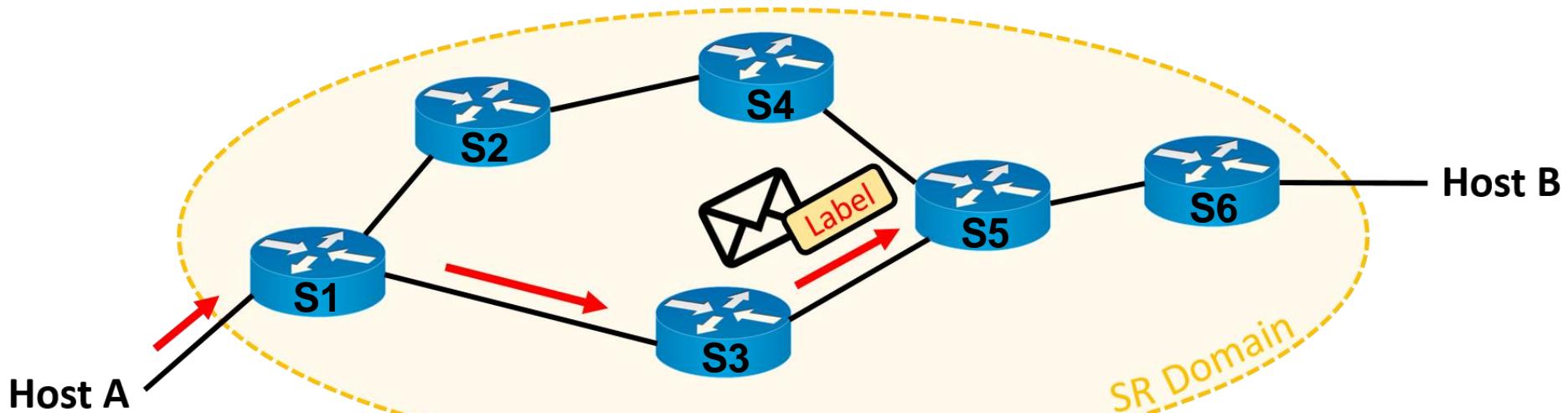
- Switch S3 receives packet with label and lookups flow table

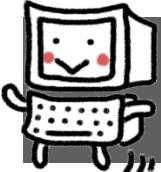




# Segment Routing (V)

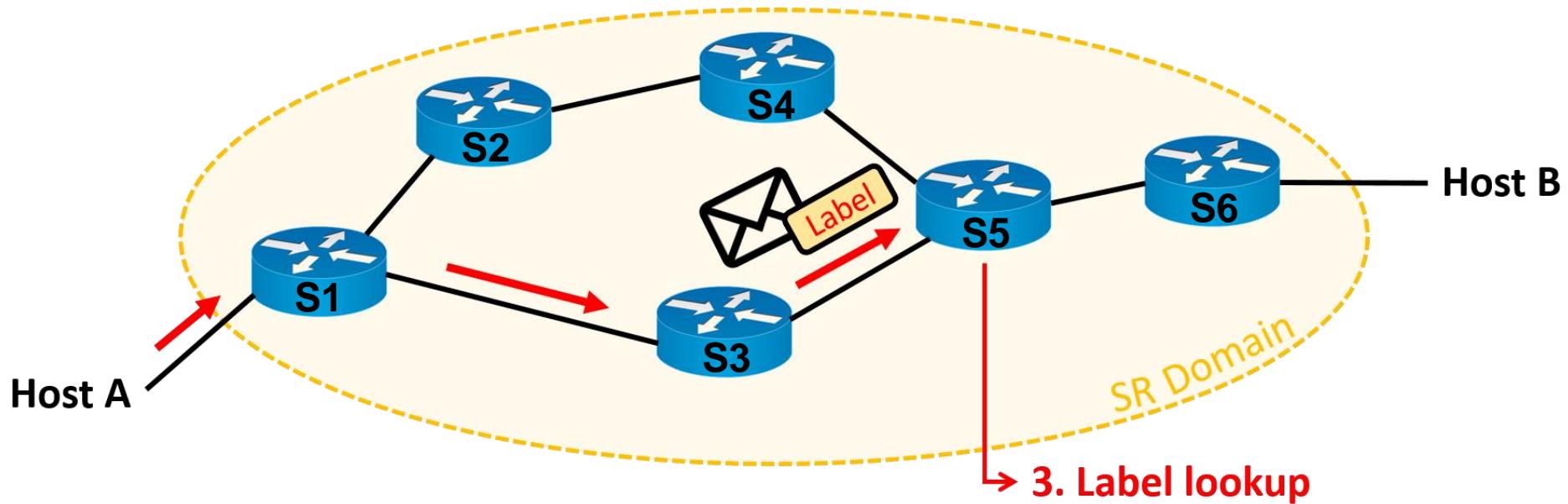
- Switch S3 forwards packet with label

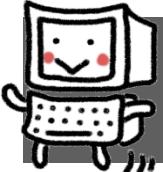




# Segment Routing (VI)

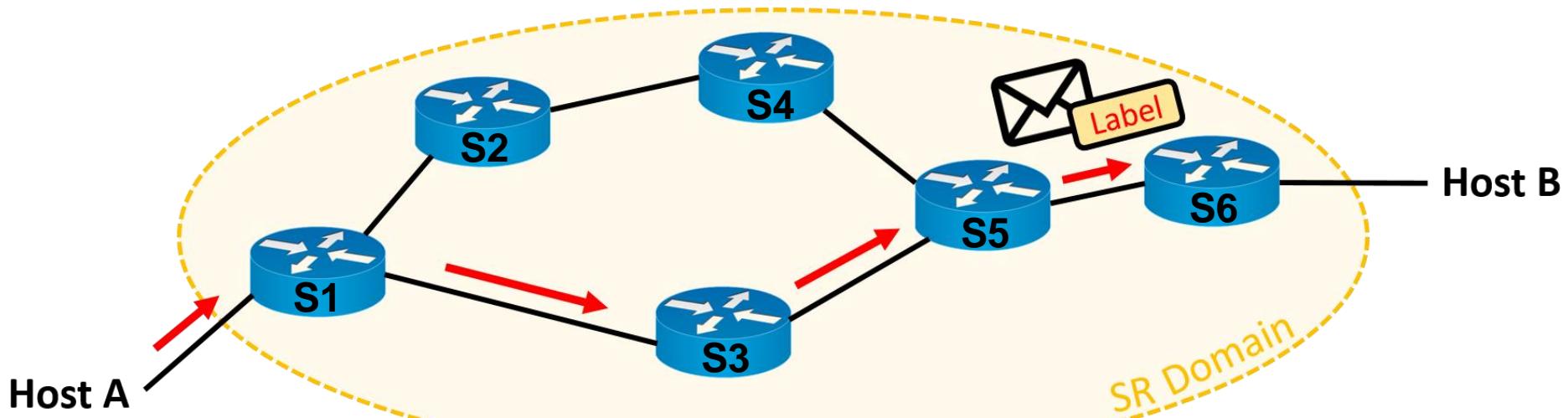
- Switch S5 receives packet with label and lookups flow table

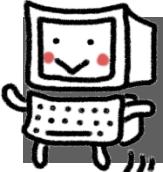




# Segment Routing (VII)

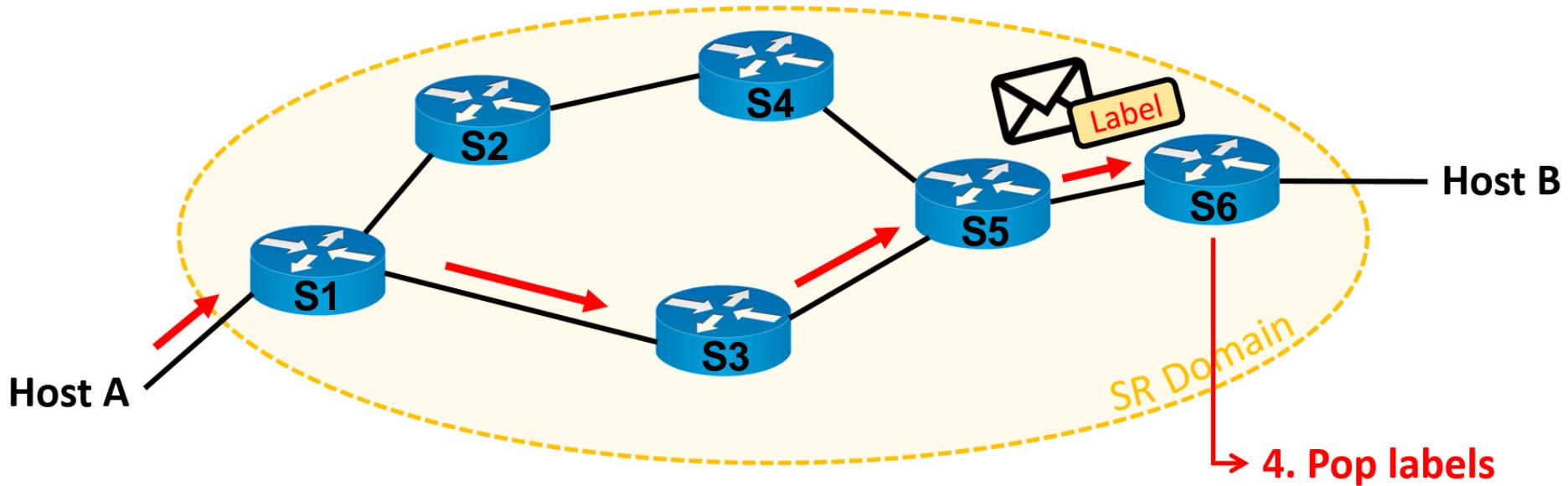
- Switch S5 forwards packet with label

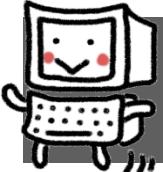




# Segment Routing (VIII)

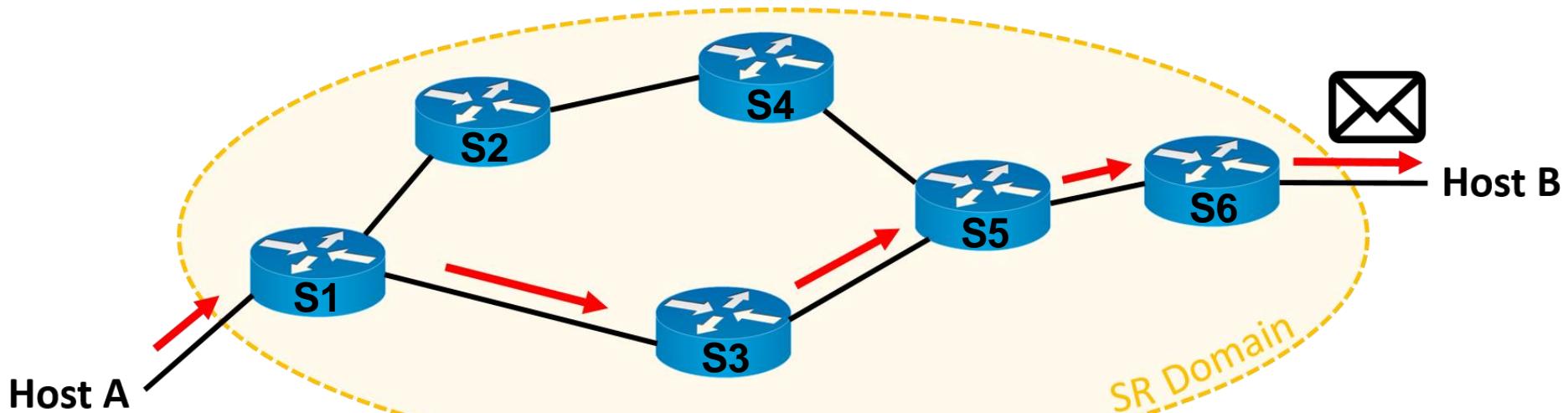
- Switch S6 receives packet with label and pops label

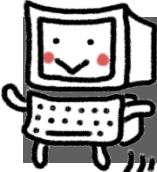




## Segment Routing (VIII)

- Switch S5 forwards the original packet





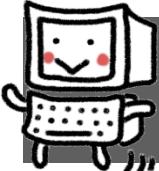
# Characteristics of Segment Routing (SR)

- A source routing mechanism
  - Support traffic engineering
  - Without requiring mid-point state
    - Such as per-flow states for traffic engineering

*"A source-routing architecture that seeks the right **balance between distributed intelligence and centralized optimization**.*

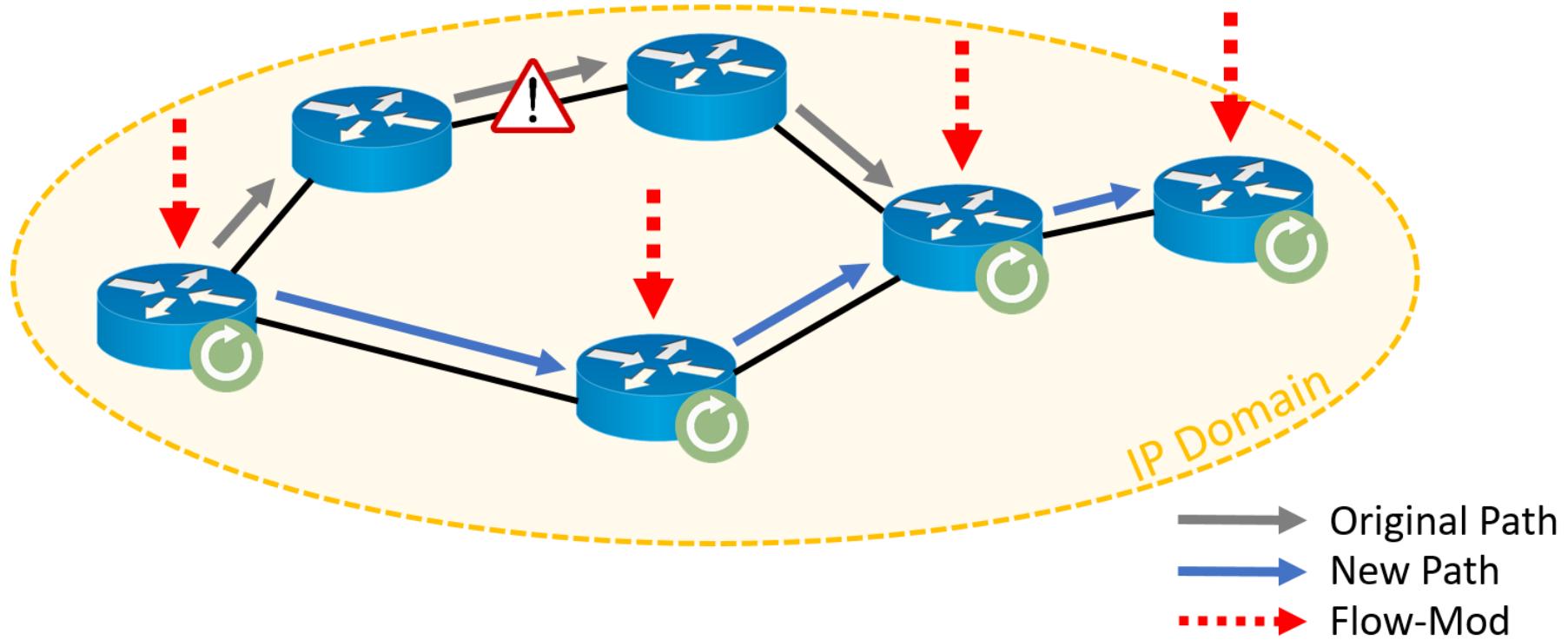
*The application steers its packets through an ordered list of instructions and realizes end-to-end policy **without creating any per-flow state** in the network."*

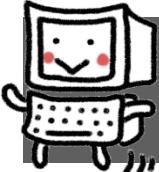
Segment Routing



# Comparison – IP Routing

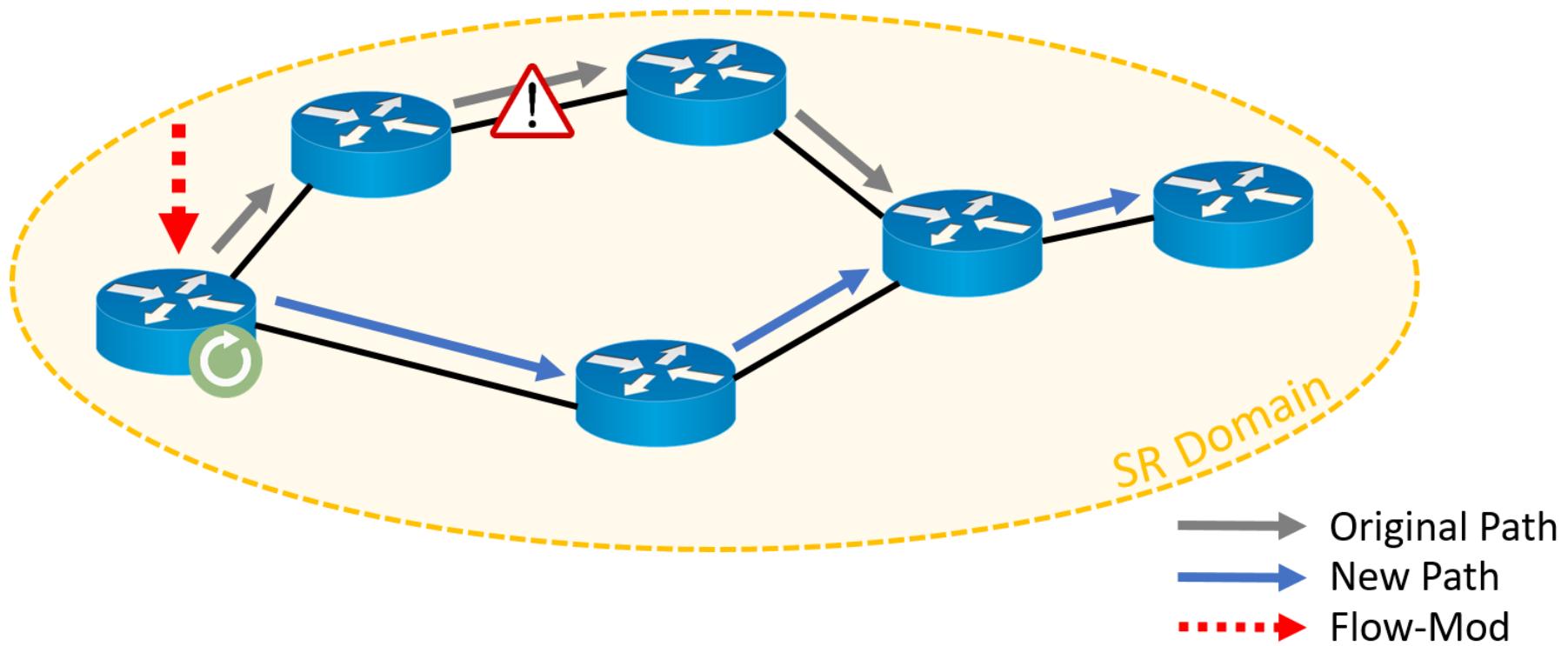
- Each of network devices on new path should be modified when
  - A link on existing flow path is down
  - New policy of traffic engineering is executed

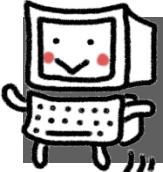




# Comparison – Segment Routing

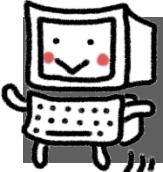
- Only edge device would be modified
  - Source routing mechanism is used
  - Flow rules could be pre-installed on mid-point switch
- Traffic engineering is more flexible and scalable





# Outline

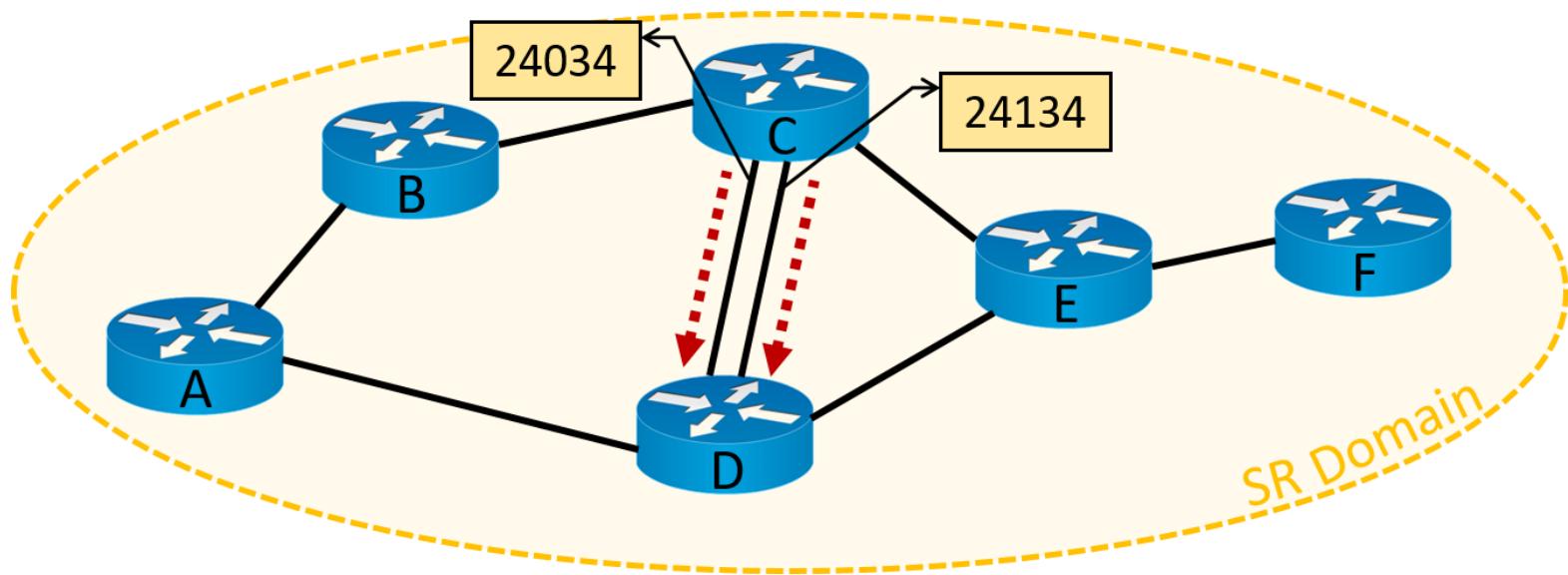
- ❑ Review of Projects
- ❑ Segment Routing
- ❑ Implementation
- ❑ Final Project Requirements
- ❑ Grading
- ❑ Hints

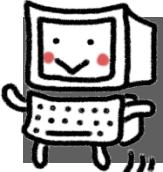


# Adjacency Segment

- ❑ Adjacency segment ID is locally unique within a SR router
- ❑ Typically single-hop

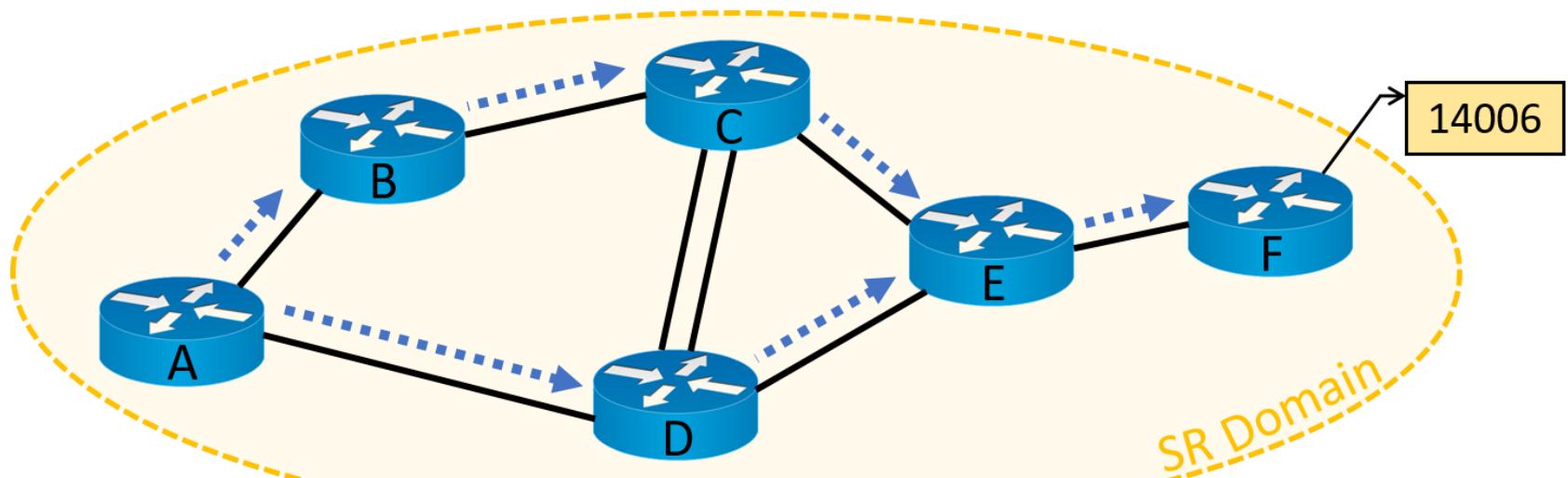
*Note: In final project, Adjacency Segment would not be used*

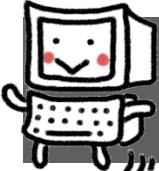




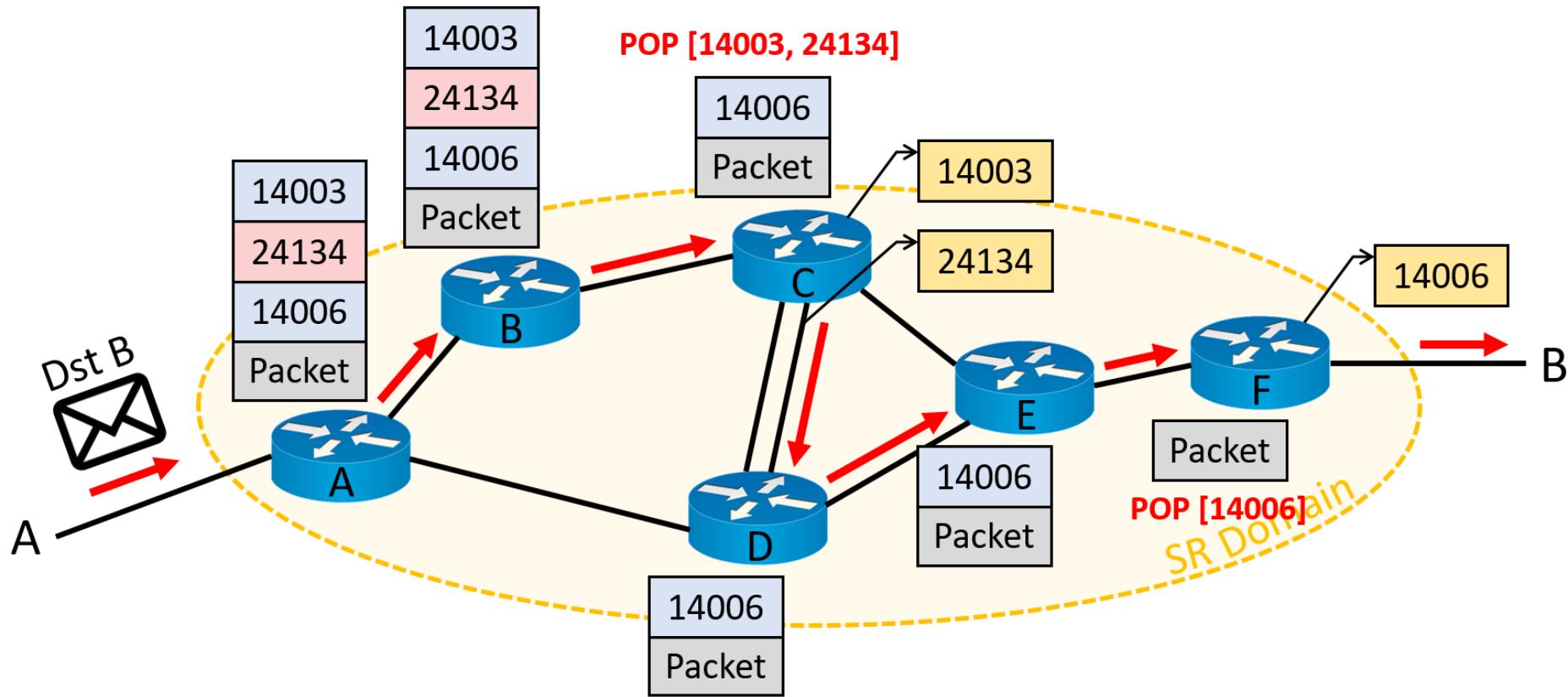
# Node Segment

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



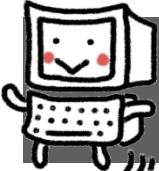


# Combining Segment



*Note: In final project, Adjacency Segment would not be used and only single Node Segment would be pushed on edge instead of segment stack*

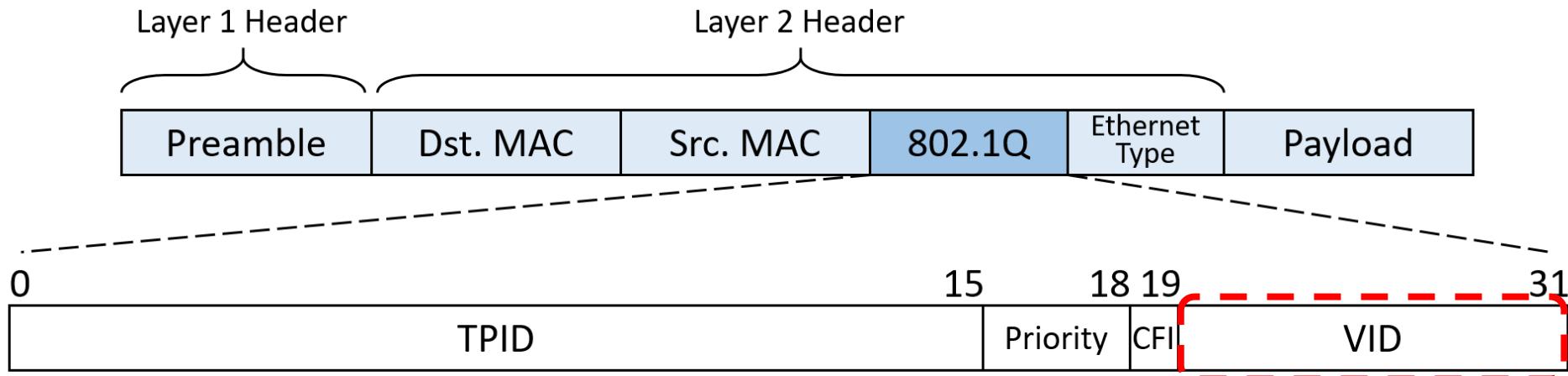
Actually in Segment Routing, penultimate hop popping is used. Therefore, the last label should be popped at the penultimate switch instead of the last switch.



# IEEE 802.1Q

- The networking standard that supports virtual LANs (VLANs)

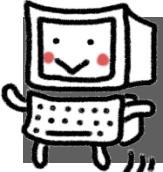
*Note: In final project, VID in VLAN tags are used as segment ID*



TPID: Tag protocol identifier (0x8100)

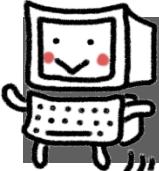
CFI: Canonical Format Indicator

VID: VLAN Identifier



# Outline

- ❑ Review of Projects
- ❑ Segment Routing
- ❑ Implementation
- ❑ Final Project Requirements
- ❑ Grading
- ❑ Hints



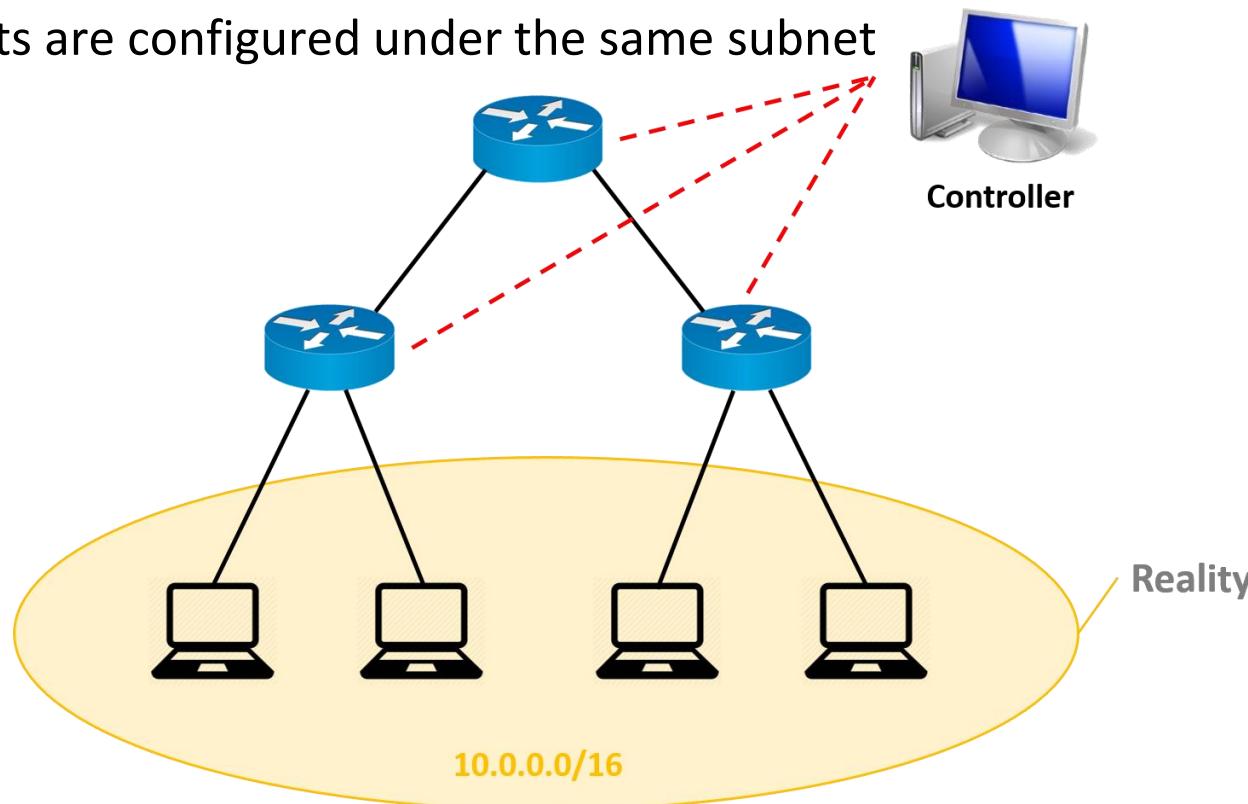
# Subnet Configuration (I)

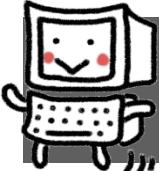
- In final project, we do not implement Layer 3 network functions

- Replacement of Layer 2 address
  - Decrement of TTL

- In Mininet topology

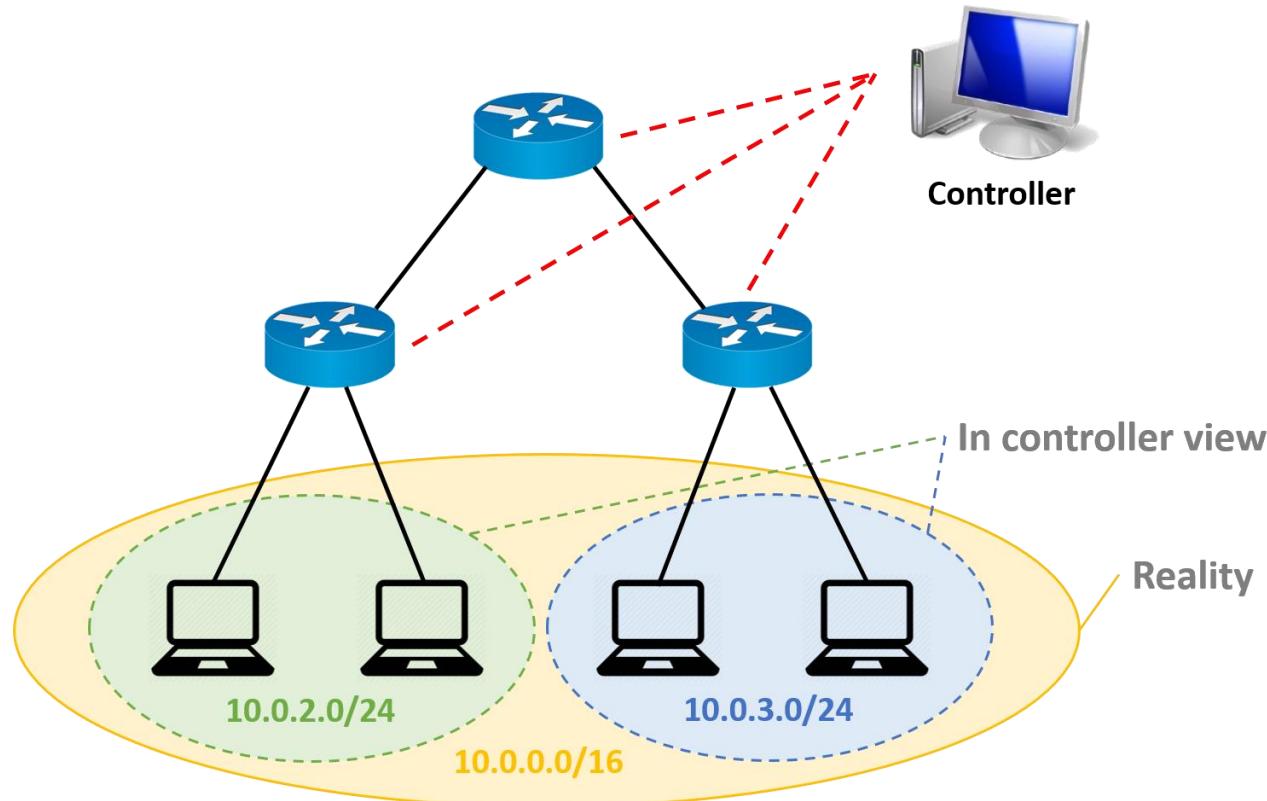
- Hosts are configured under the same subnet

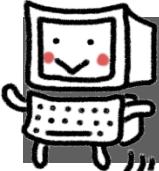




# Subnet Configuration (II)

- Segment Routing assigns subnet to Segment ID of attaching edge node
  - Controller needs a virtual subnet
- In controller view
  - Hosts are configured with different subnets

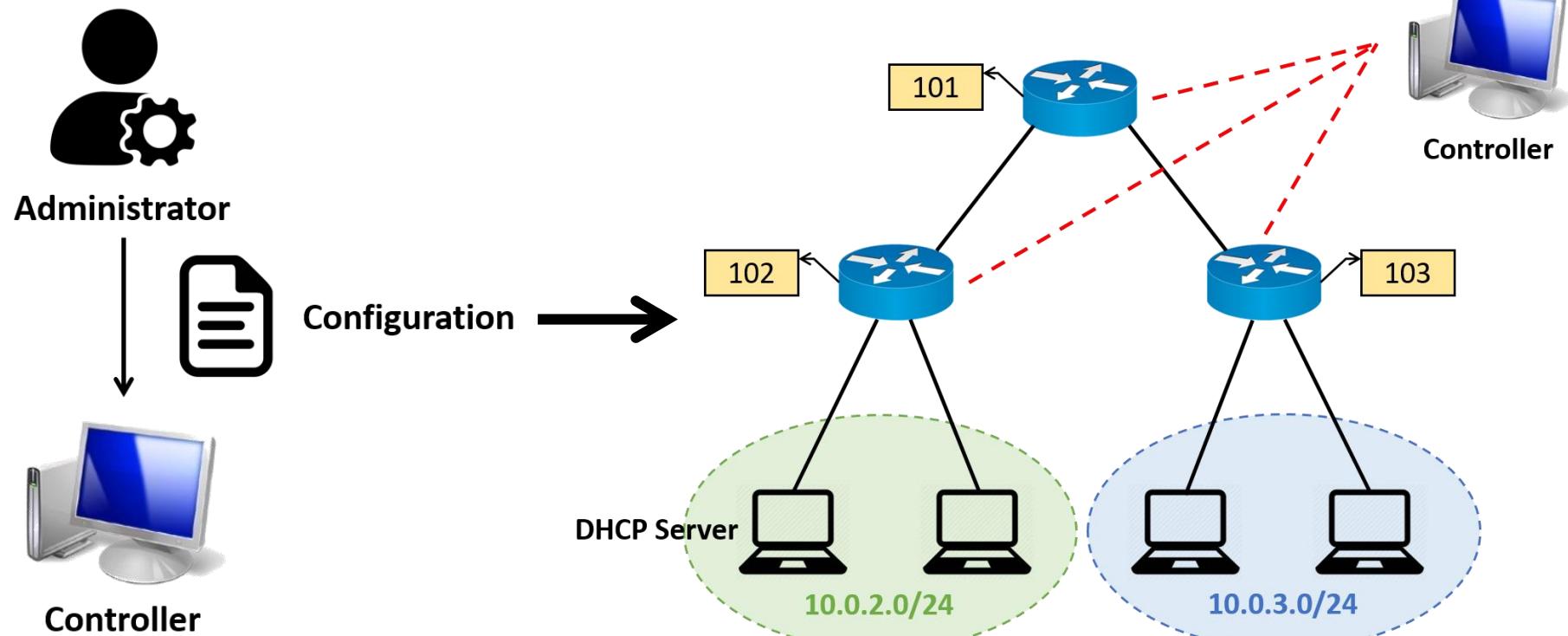


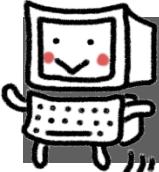


# Workflow (I) – Configuring

- Administrator uploads configuration to controller

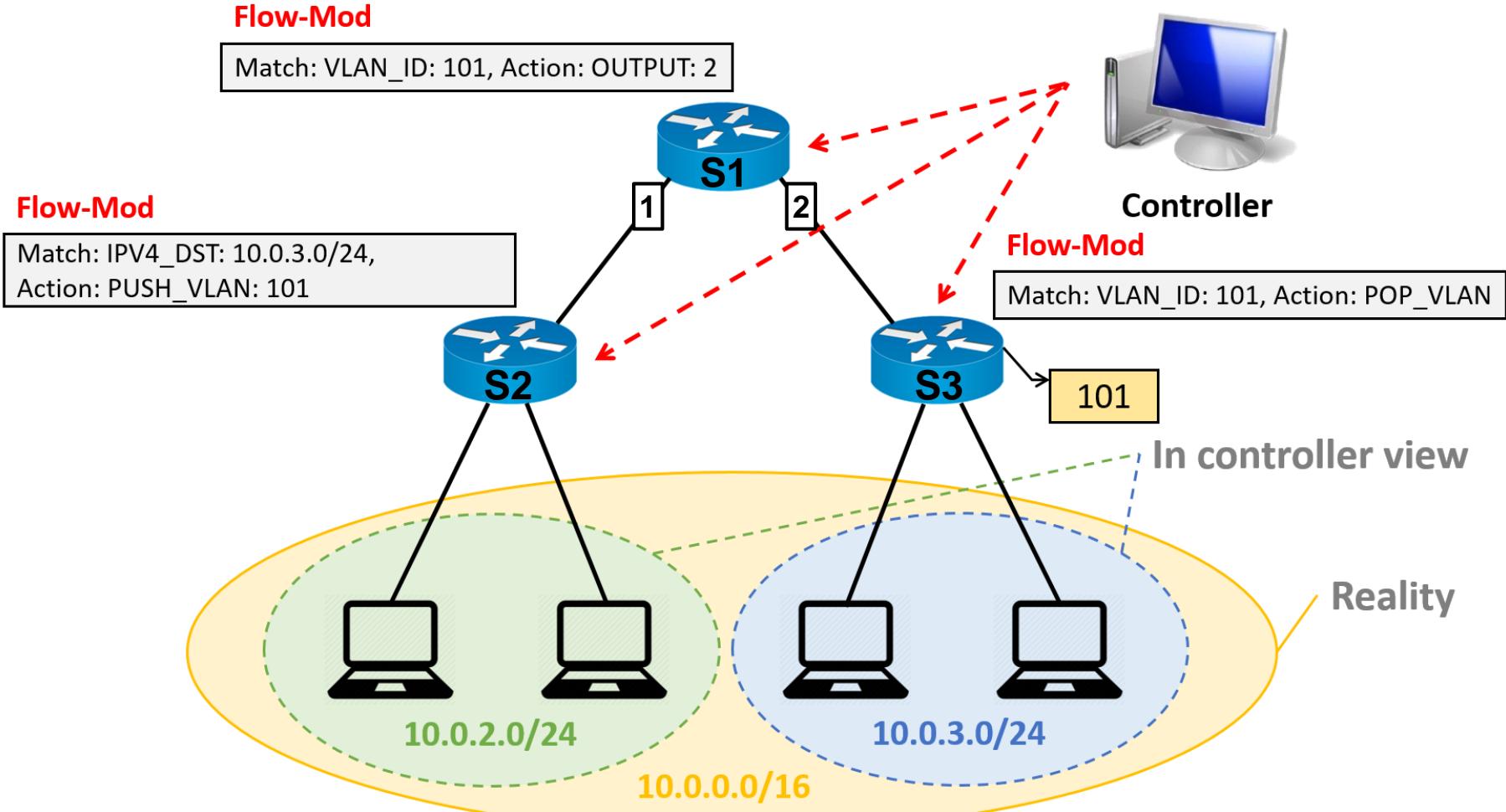
- DHCP server location
- Segment ID for switch (node segment)
- IP subnet on edge switch
- Other configuration as you wish, e.g. indication of edge devices

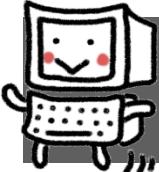




# Workflow (II) – Flow Rules Pre-installing

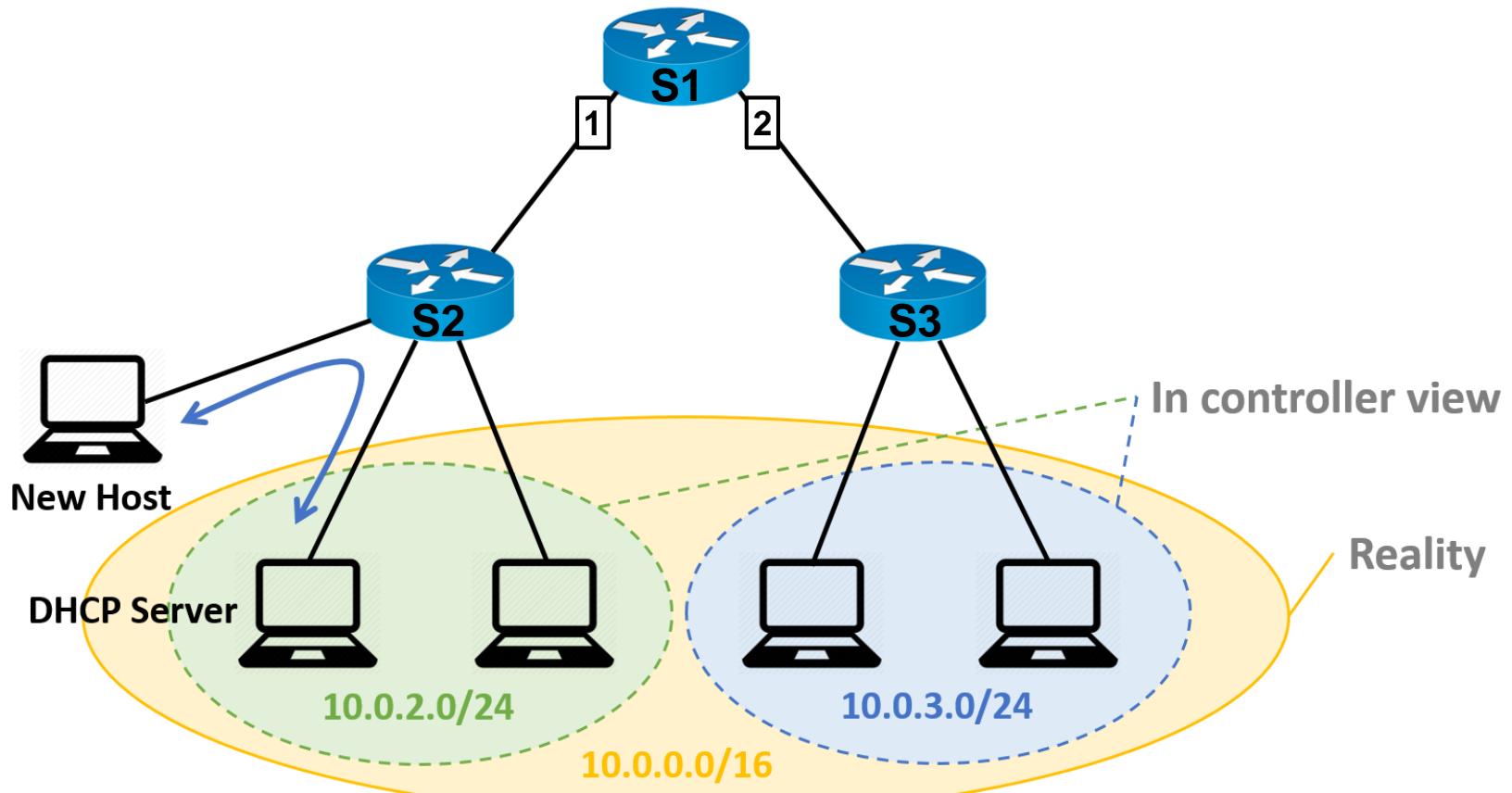
- Controller would install flow rules once being configured

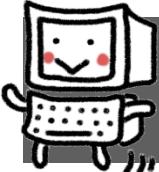




## Workflow (III) – New Host Attaching

- ☐ New host could request IP address from DHCP server
  - This is what you done in Project 6



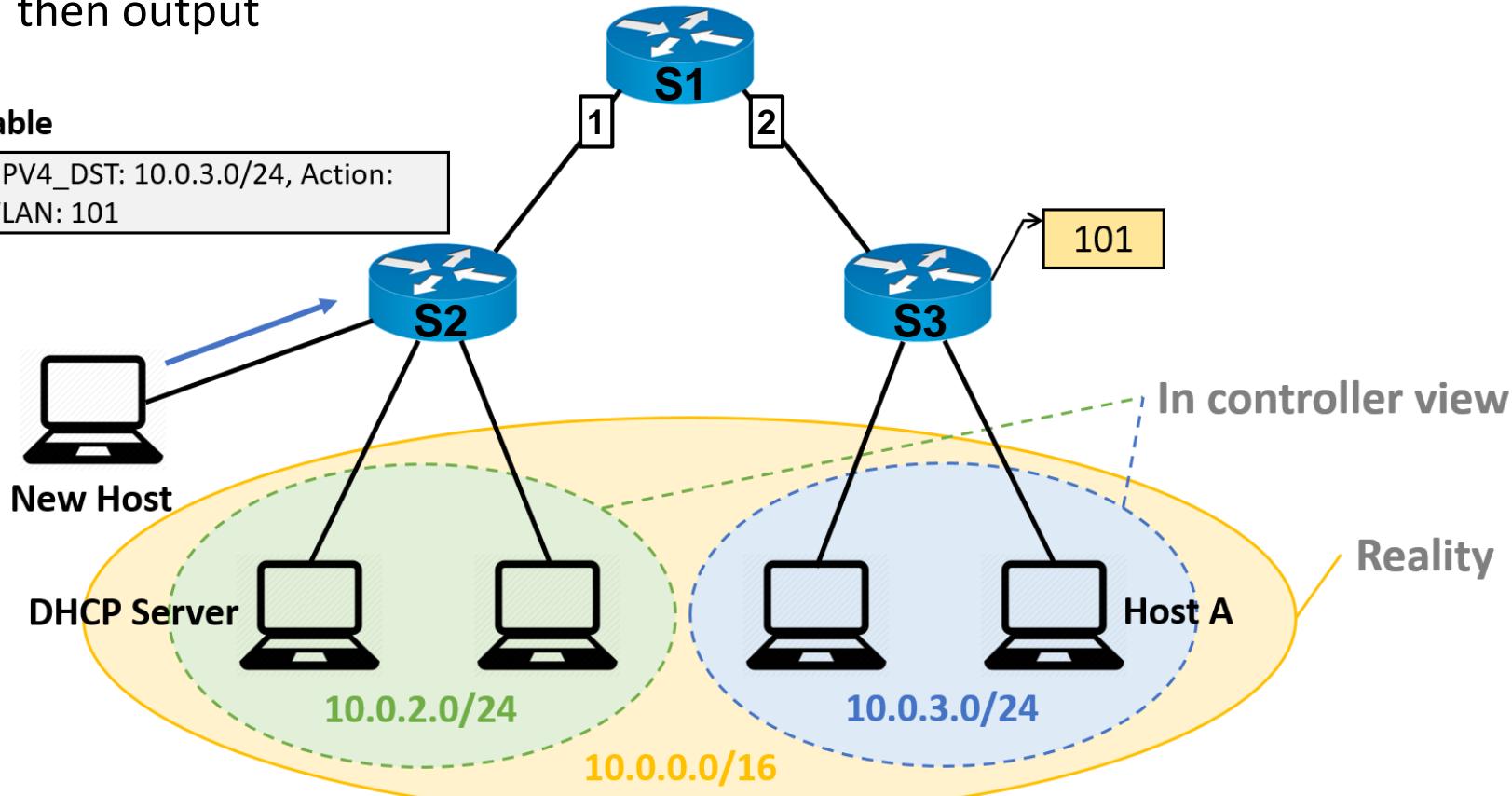


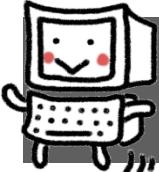
# Workflow (IV) – Label Pushing

- When new host sends packet to Host A
  - Subnet of destination address is matched
  - Packet is pushed with VLAN tag corresponding to destination edge and then output

Flow Table

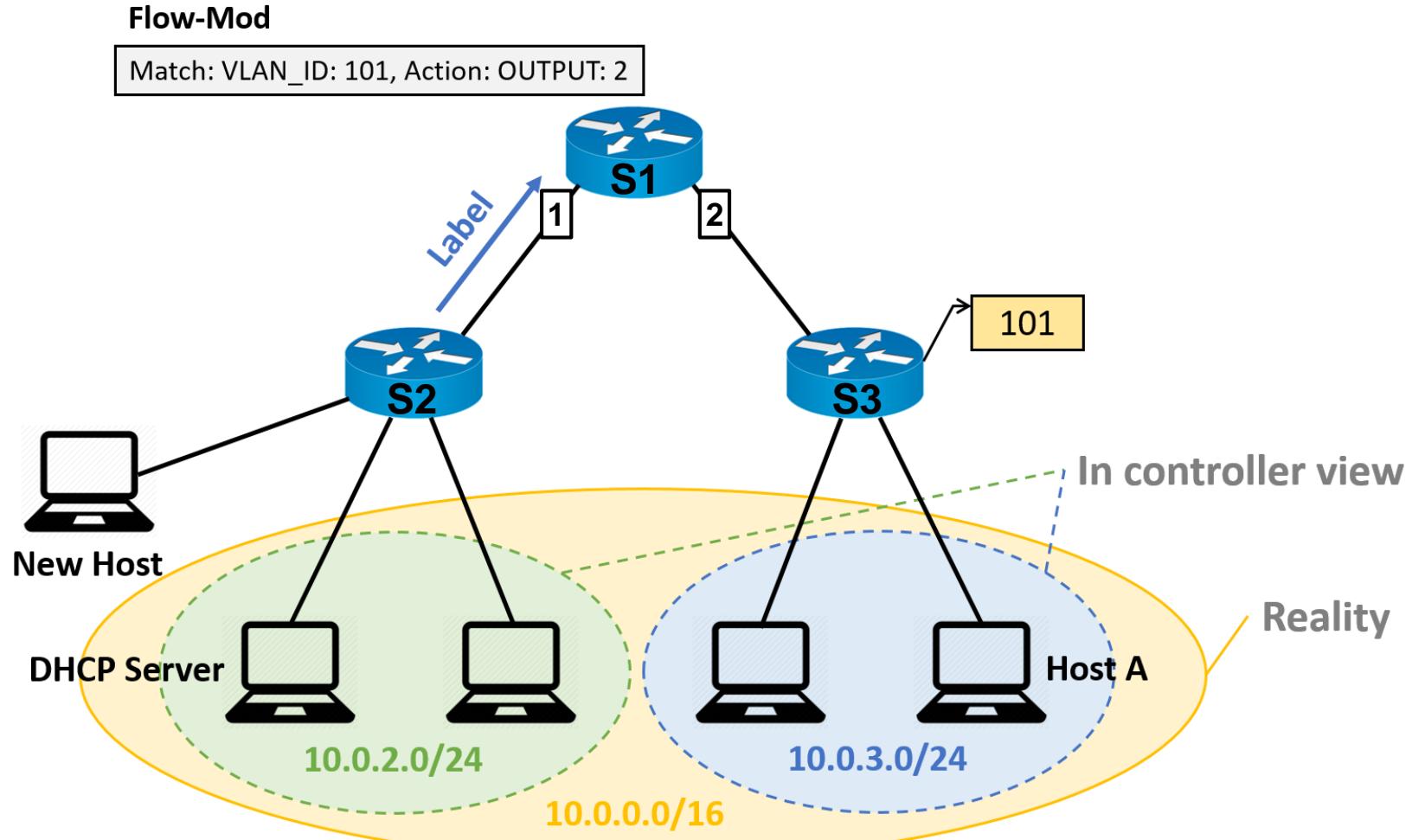
Match: IPV4\_DST: 10.0.3.0/24, Action:  
PUSH\_VLAN: 101

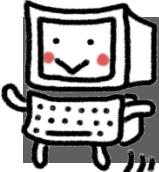




# Workflow (V) – Label Lookup

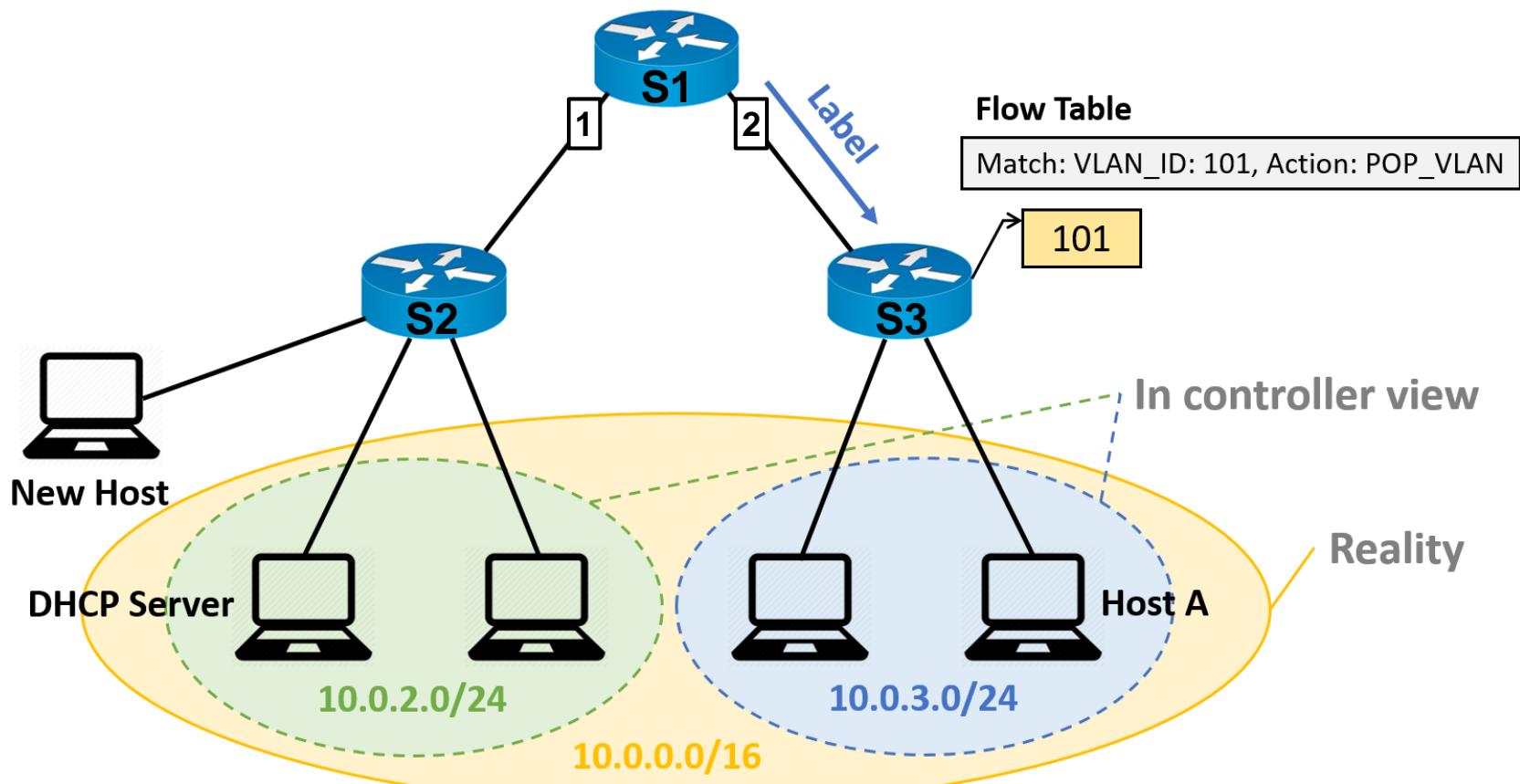
- Packet with VLAN tag is received by mid-point devices and forwarded

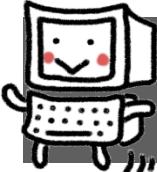




# Workflow (VI) – Label Popping

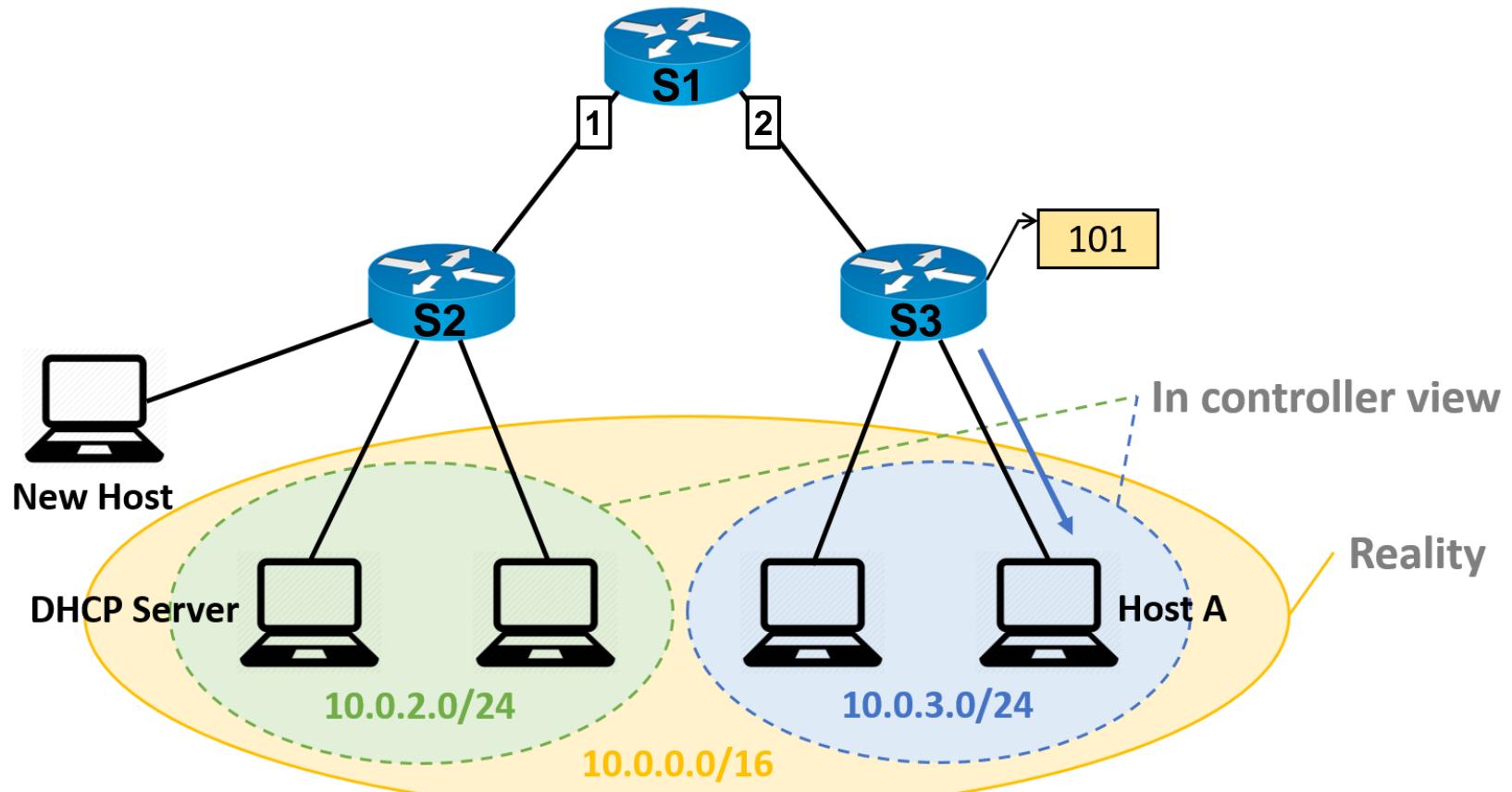
- When destination edge receives packet with label
  - Pop VLAN tag

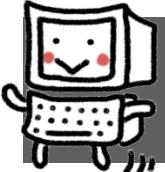




# Workflow (VII) – IP/MAC Forwarding

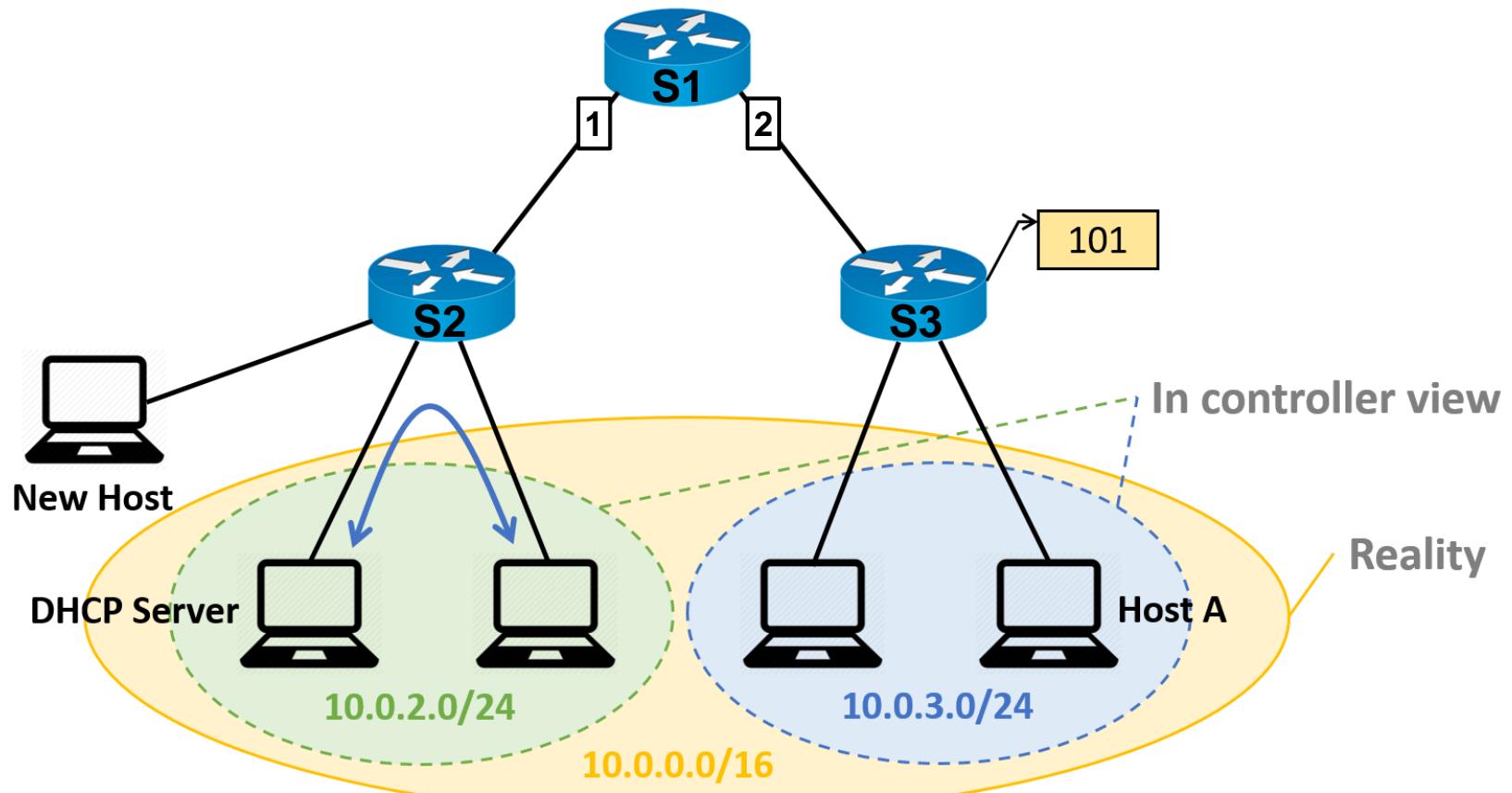
- Forward packets by matching Layer 2 or 3 address

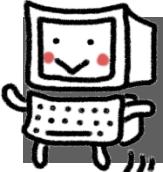




## Workflow (VIII) – Intra-device Forwarding

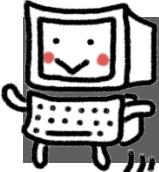
- Intra-device packet could be forwarded by matching Layer 2 or 3 address





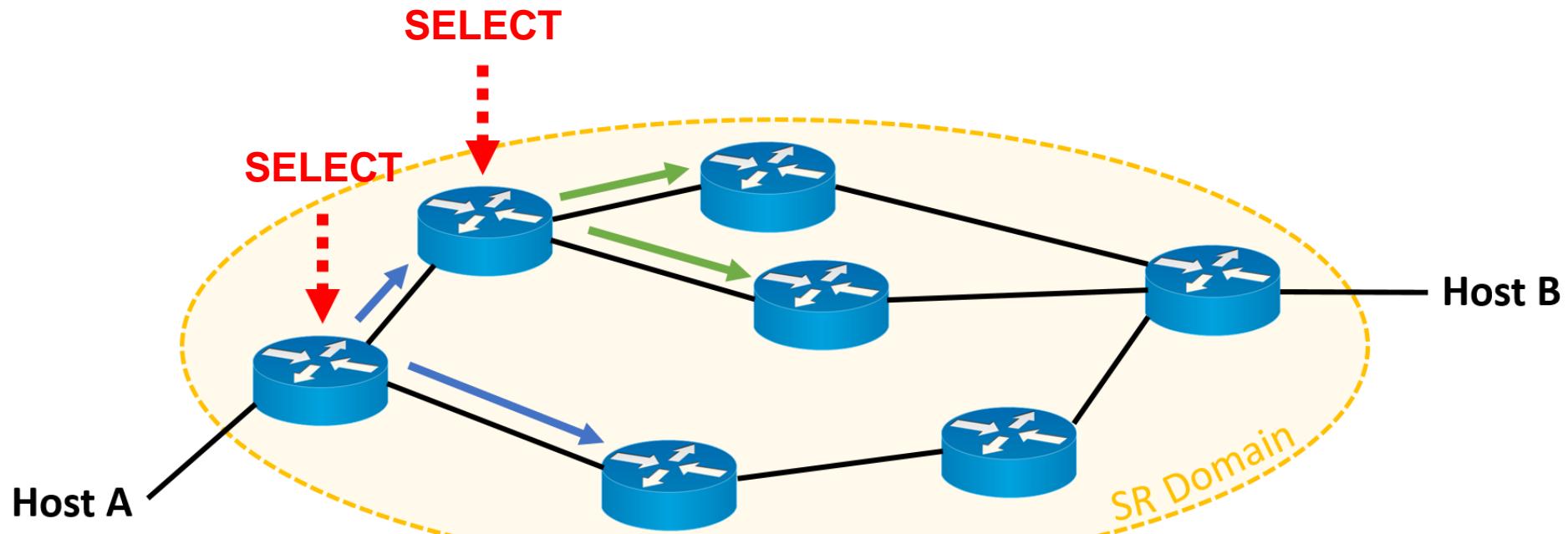
# Requirements (I)

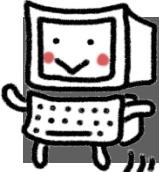
- In final project, the following three applications should be activated
  - DHCP unicast
  - Proxy ARP
  - Path service/VLAN-based Segment Routing
- You should not activate any other applications except OpenFlow-based applications for ONOS users



## Requirements (II) – VLAN-based Segment Routing

- ❑ Forward packets with label switching and source routing mechanism
- ❑ If there are multiple paths with same hop count, use SELECT group to achieve load balancing
- ❑ All flow/group rules should be installed once controller receives configuration





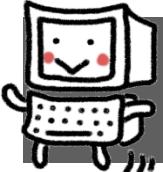
# Submit to e3

## □ Files

- All files of your application
- A report: **final\_studentID.pdf**
  - Write down details of your implementation
    - E.g. fields of installed flow rules

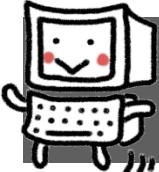
## □ Submit

- Upload “.zip” file to e3
  - Named: **final\_studentID.zip**
- Wrong file name or format would not be scored



# Outline

- ❑ Review of Projects
- ❑ Segment Routing
- ❑ Implementation
- ❑ Final Project Requirements
- ❑ Grading
- ❑ Hints



# Grading

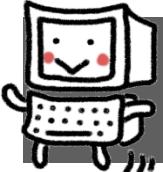
## □ Previous Projects Part

- DHCP unicast (20%)
- Proxy ARP (20%)
- Routing service (20%)

*You could get 60 points as you successfully run all previous projects*

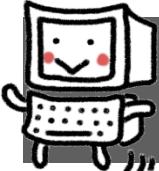
## □ Final Project Part

- Segment routing (20%)
- SELECT Group rules (10%)
- Report (10%)



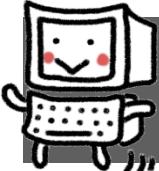
# Outline

- Review of Projects
- Segment Routing
- Implementation
- Final Project Requirements
- Grading
- Hints



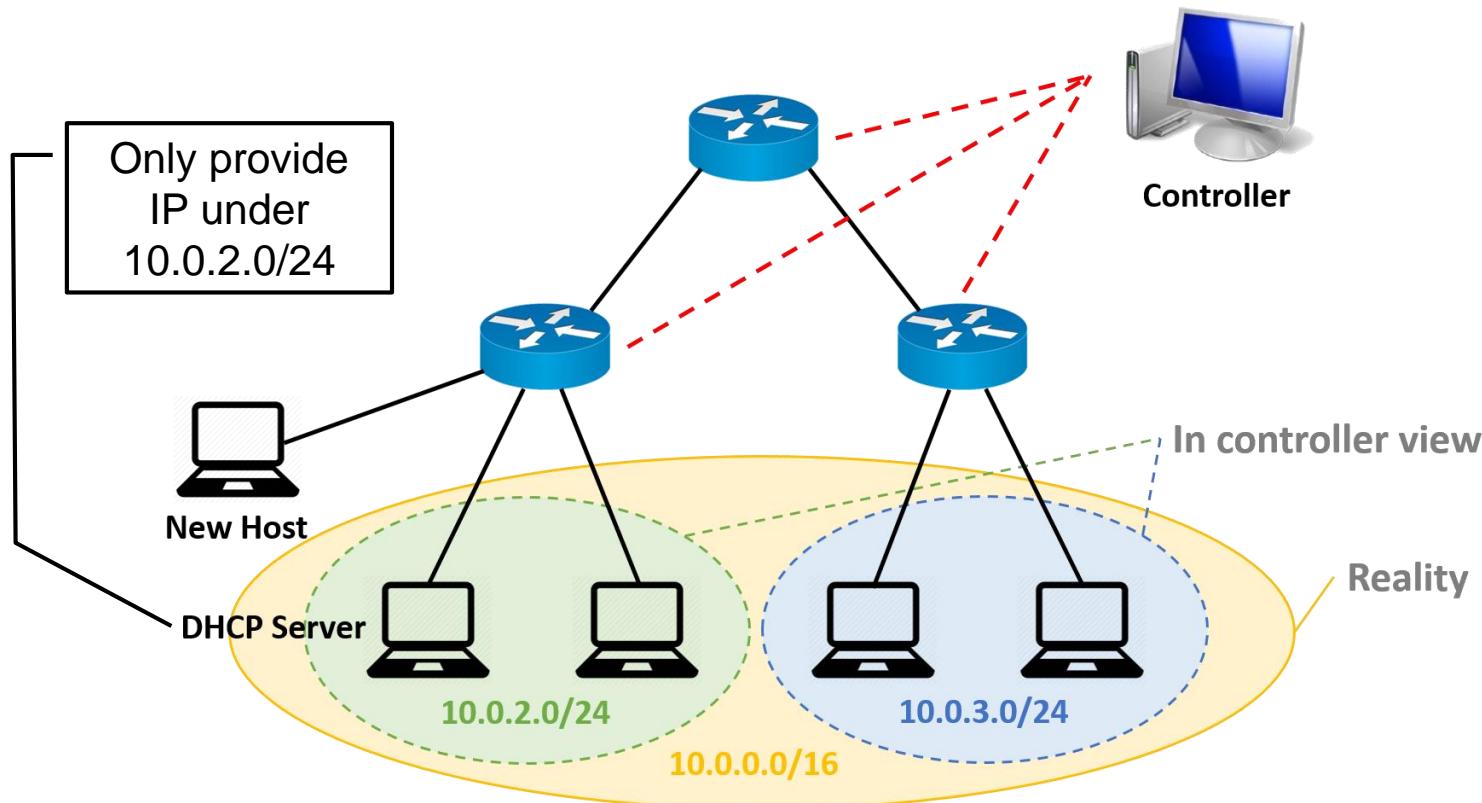
## Hints

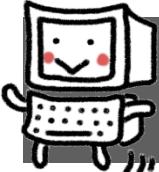
- You should design your configuration and installed flow/group rules
- At least following configuration should be included
  - DHCP server location
  - Segment ID for each switch
  - Subnet attached to edge switch
- Flow rule interference should be considered
  - Assign proper priority to each flow rule
- Packet-In would be received by all activated applications
  - Make sure you drop irrelevant packet type in each application
- In VLAN-based Segment Routing
  - For inter-device packets, using INDIRECT/SELECT group rules to output is recommended
  - For intra-device packets, you can simply use flow rules to match and output



# Issues (I)

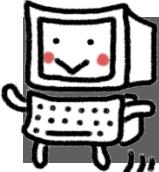
- Since we do not implement Layer 3 network functions
  - DHCP Server would be discovered by hosts in all subnets
- In our topology, host with dynamic IP would be under same subnet of DHCP server





## Issues (II)

- Since controller do not know host at the first time
  - May not be able to install correct flow rules when receiving configuration
- First solution
  - Send packets before you push configuration to let controller get host information
  - When you test host connectivity, use host with static IP
- Second solution
  - Add host information in configuration
- You can figure out other solution on your own



# Reference

## □ Installing group rules

### ■ Ryu

- [https://github.com/osrg/ryu/blob/704dcc786aa9f975a9d486d9a1699fd7995ecd4b/ryu/ofproto/ofproto\\_v1\\_3\\_parser.py#L3654](https://github.com/osrg/ryu/blob/704dcc786aa9f975a9d486d9a1699fd7995ecd4b/ryu/ofproto/ofproto_v1_3_parser.py#L3654)

### ■ ONOS

- <https://github.com/opennetworkinglab/onos/blob/master/apps/segmentrouting/app/src/main/java/org/onosproject/segmentrouting/grouphandler/PolicyGroupHandler.java#L214>