



Tìm hiểu và triển khai mạng SDN (Software-Defined Network)

Báo cáo đồ án môn học

Giảng viên hướng dẫn: ThS. Đỗ Hoàng Hiển



Thành viên

Dương Phước Nhật Nam

23520968

Nhóm trưởng

Lương Hoàng Long

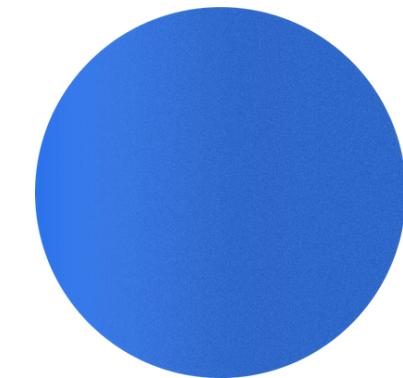
23520879

Nguyễn Gia Luân

23520896

Lê Minh

23520928



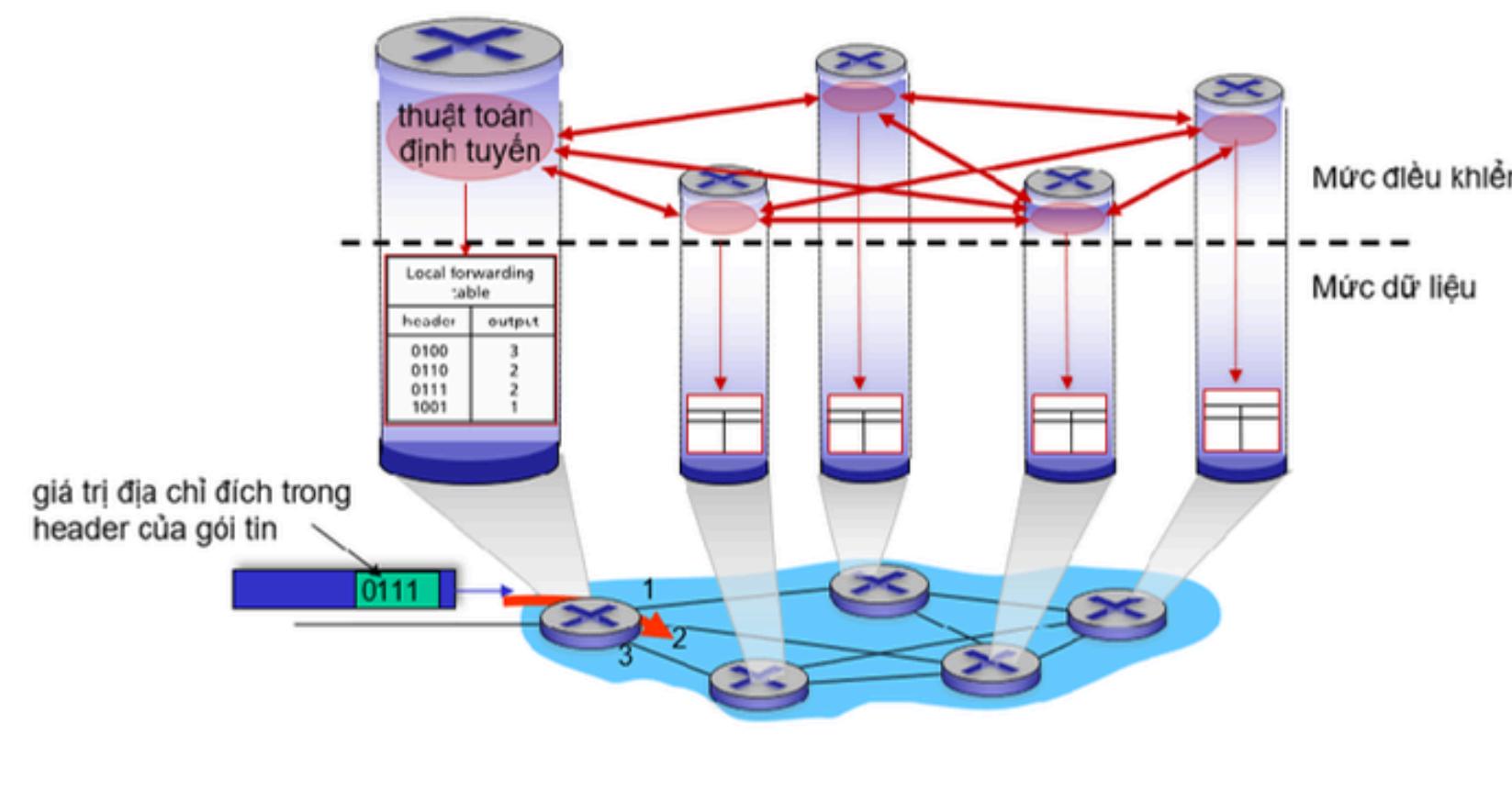
Giới thiệu đề tài

Trước bối cảnh cơ sở hạ tầng mạng ngày càng phức tạp, những đòi hỏi về khả năng quản lý mạng linh hoạt và tự động hóa đang trở nên quan trọng hơn bao giờ hết.

Software-Defined Network (SDN) chính là một cách tiếp cận khác với mạng truyền thống, cung cấp các giải pháp cho những hạn chế đang có của mạng truyền thống.

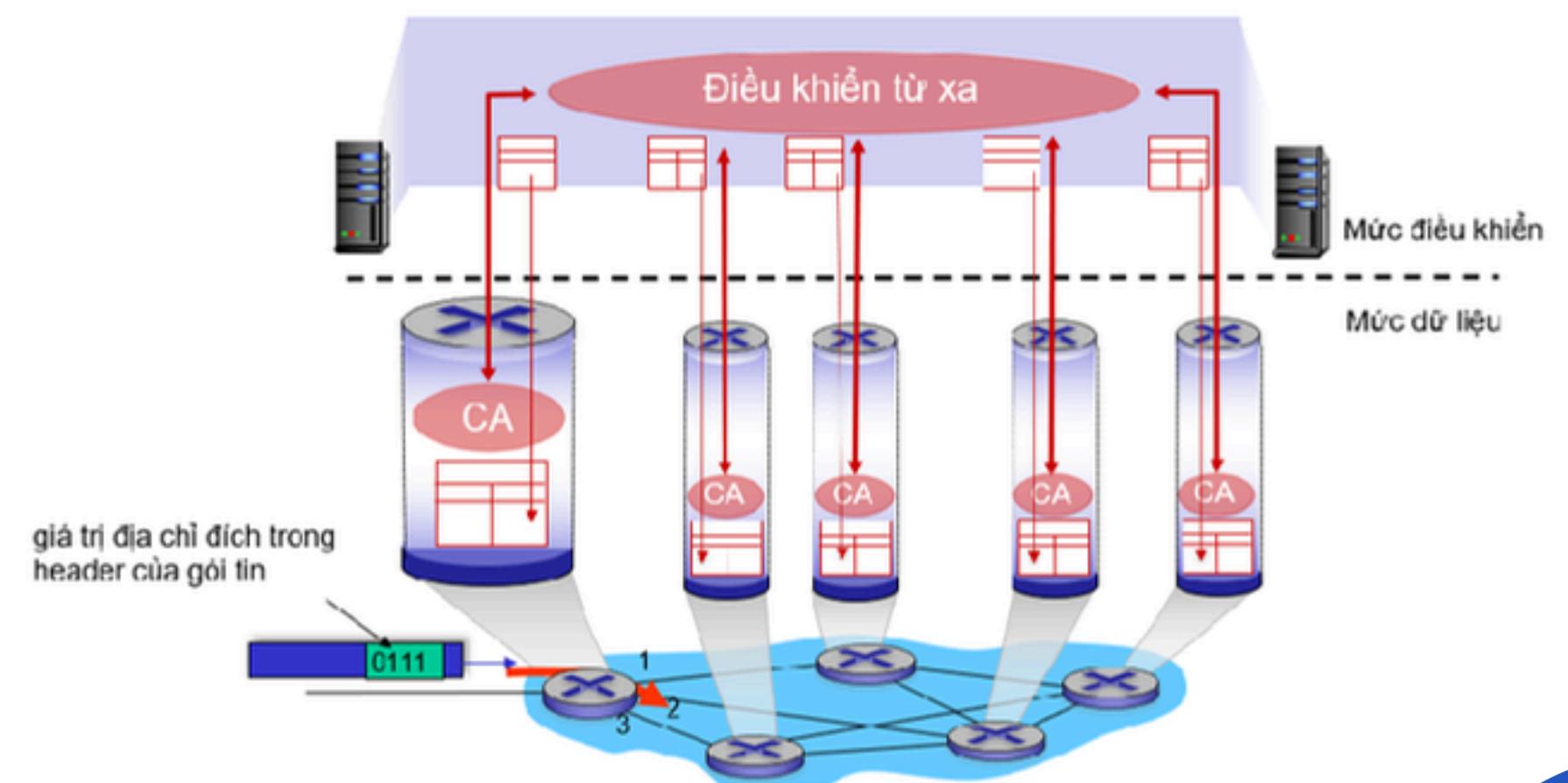


Tổng quan về mạng SDN



Mạng truyền thống

Mạng SDN

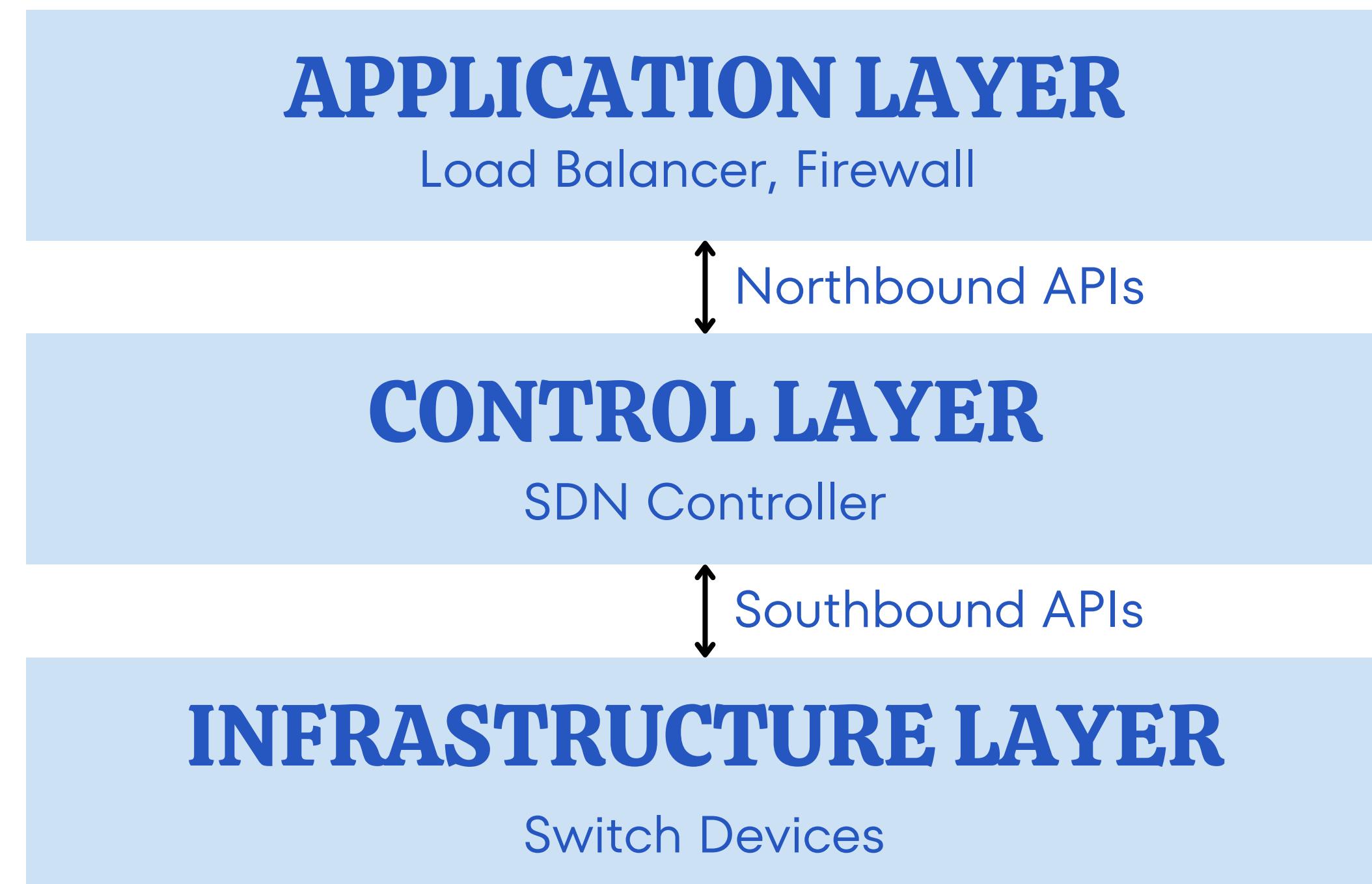


Hình 2: Mạng SDN

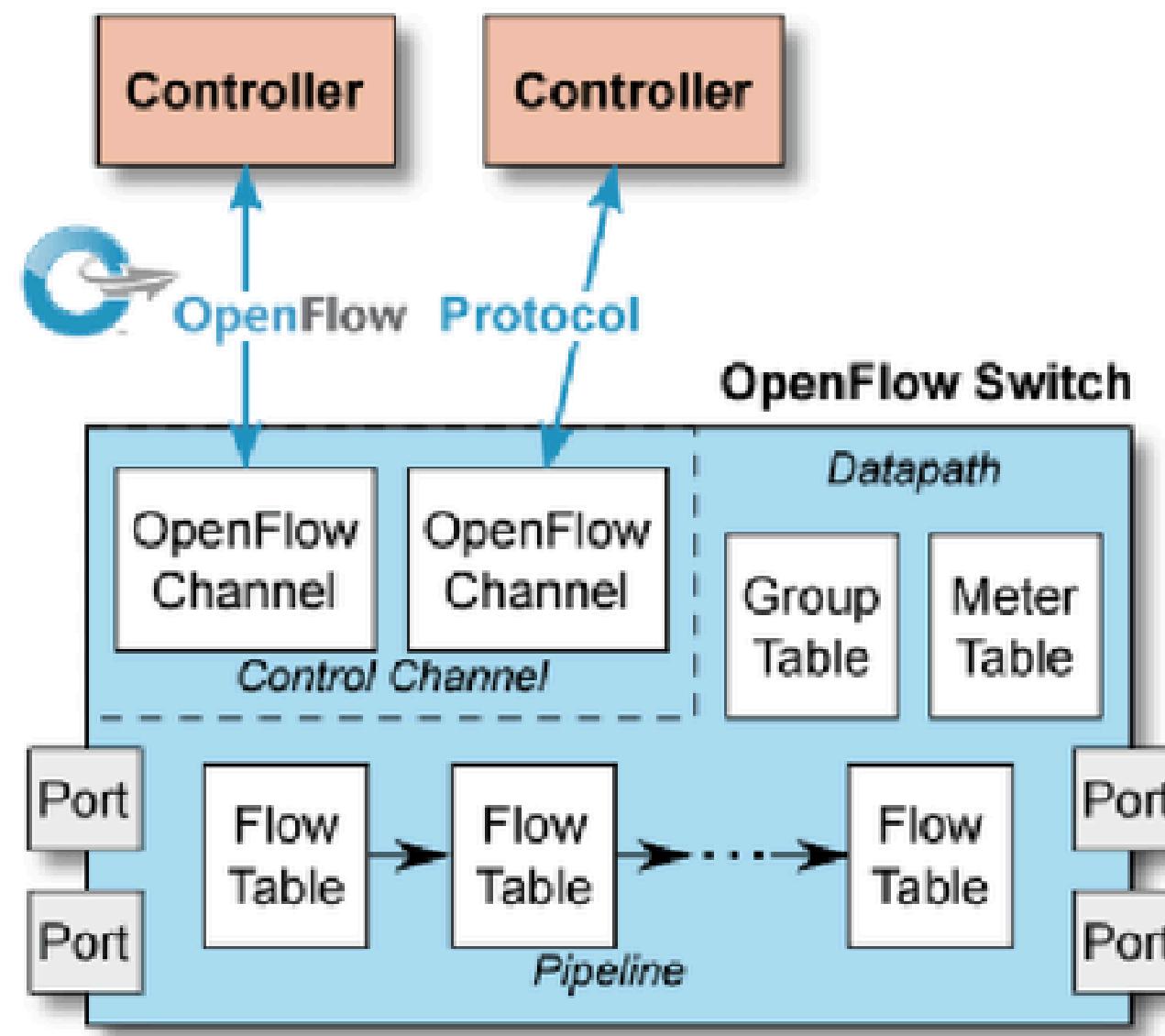
Lý do sử dụng mạng SDN

- Cho phép cấu hình mạng tại bộ điều khiển
- Chi phí vận hành, bảo trì và mở rộng thấp hơn
- Tự động hóa việc cấu hình mạng
- Có thể cài đặt dịch vụ cho bộ điều khiển để phân phối cho hệ thống mạng một cách dễ dàng

Kiến trúc mạng SDN



OpenFlow



Hình 4: Thành phần chính của OpenFlow switch

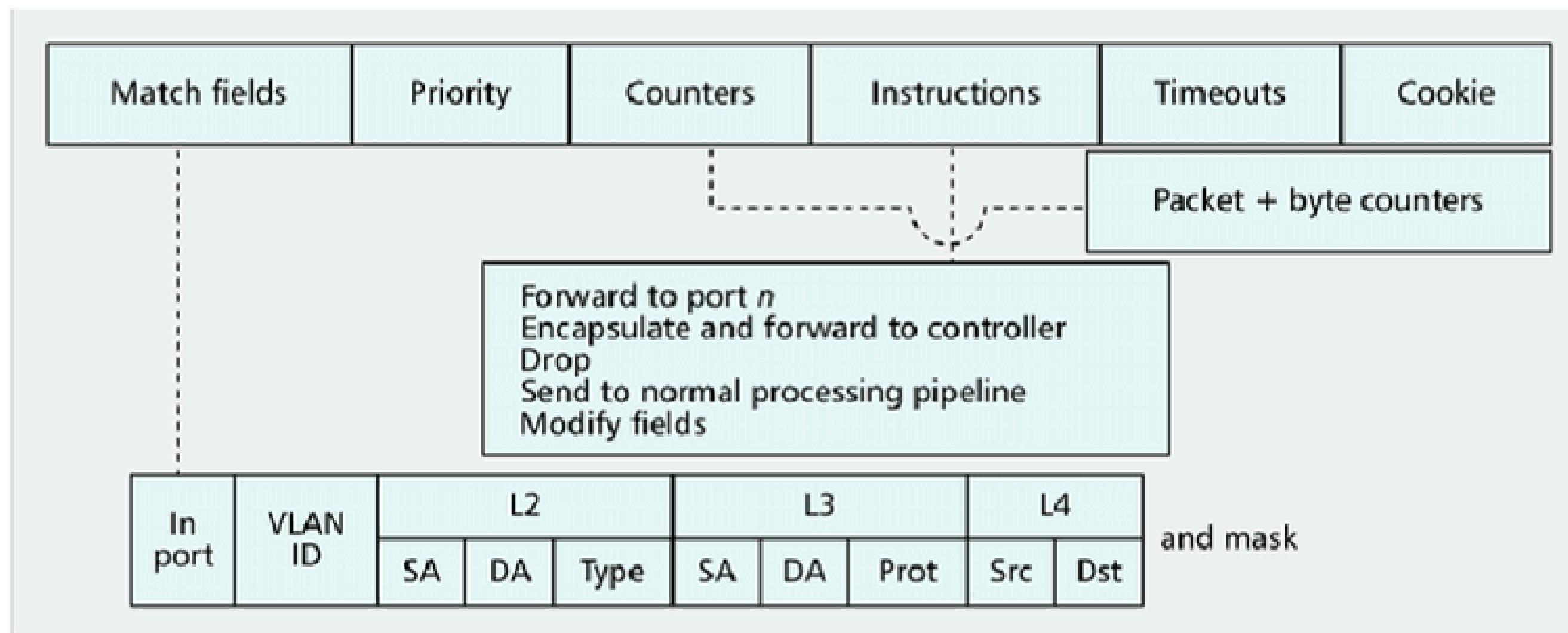
OpenFlow là giao thức để lớp điều khiển và lớp cơ sở hạ tầng giao tiếp với nhau.

Trong một OpenFlow switch, sẽ có các kênh để bộ điều khiển có thể sử dụng để giao tiếp và điều khiển.

Trong một OpenFlow switch có thể có nhiều flow table.

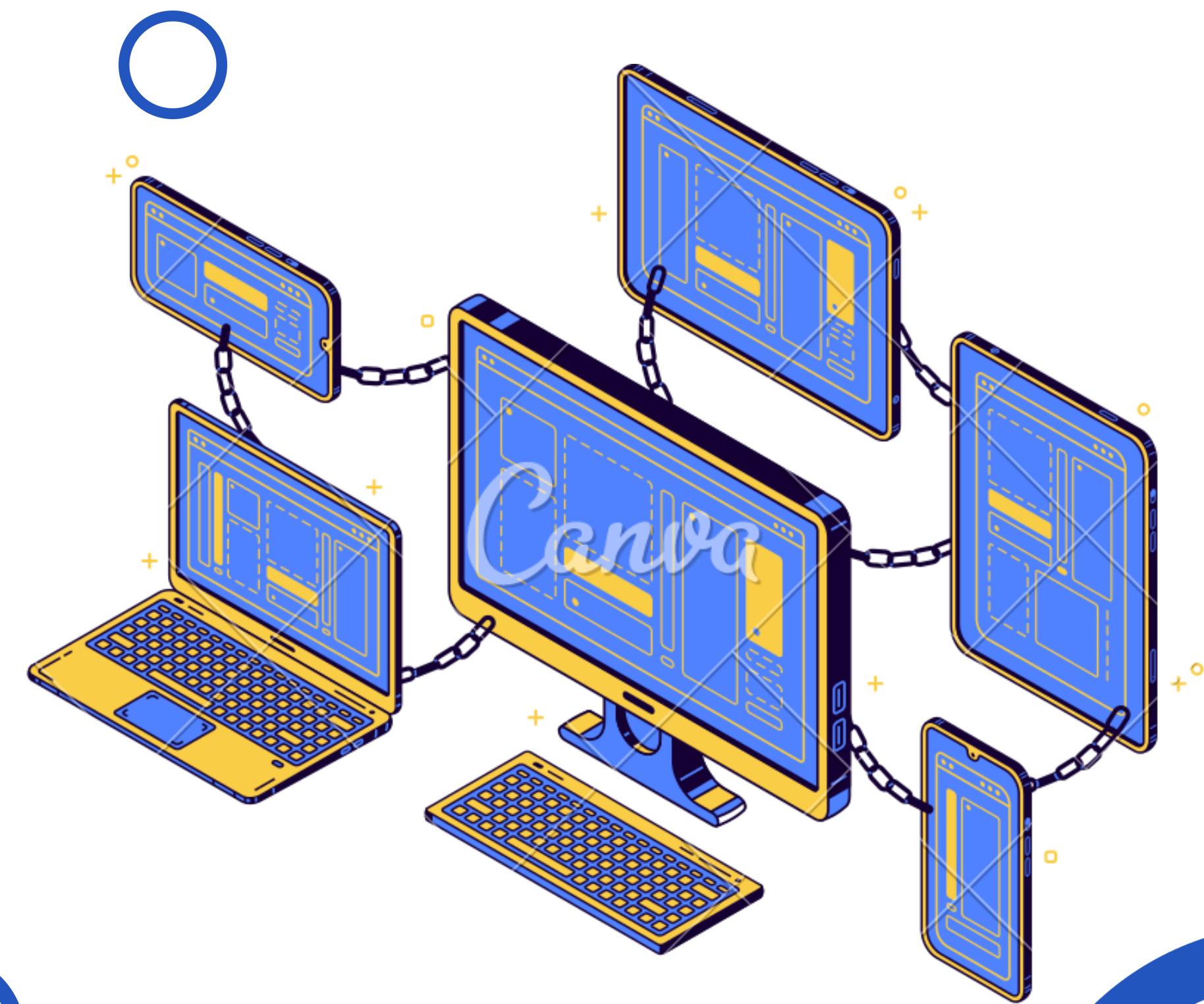
OpenFlow

Một flow table có thể chứa một hoặc nhiều flow entry.



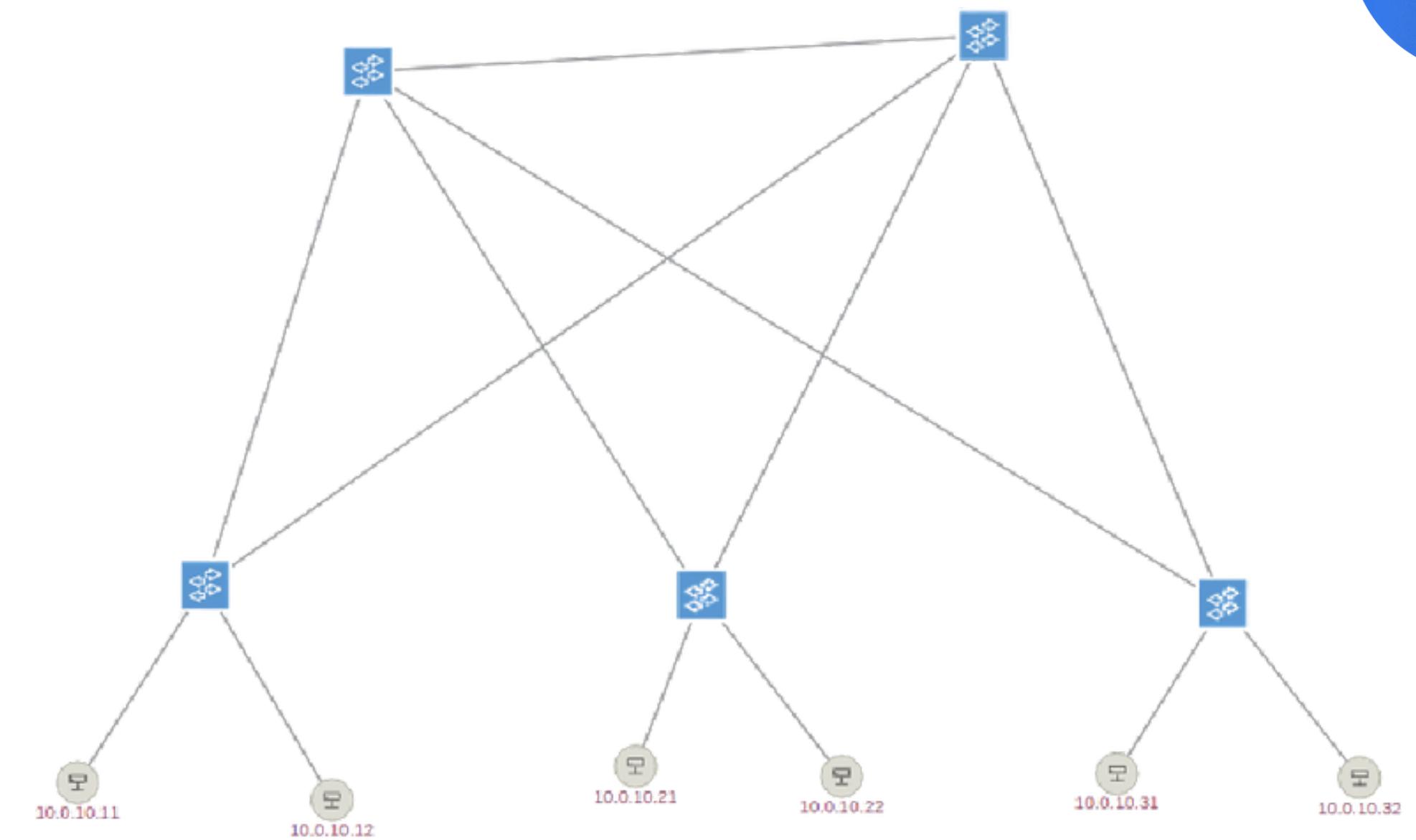
Hình 5: Thành phần chính của flow entry trong một flow table

Hiện thực hóa đề tài



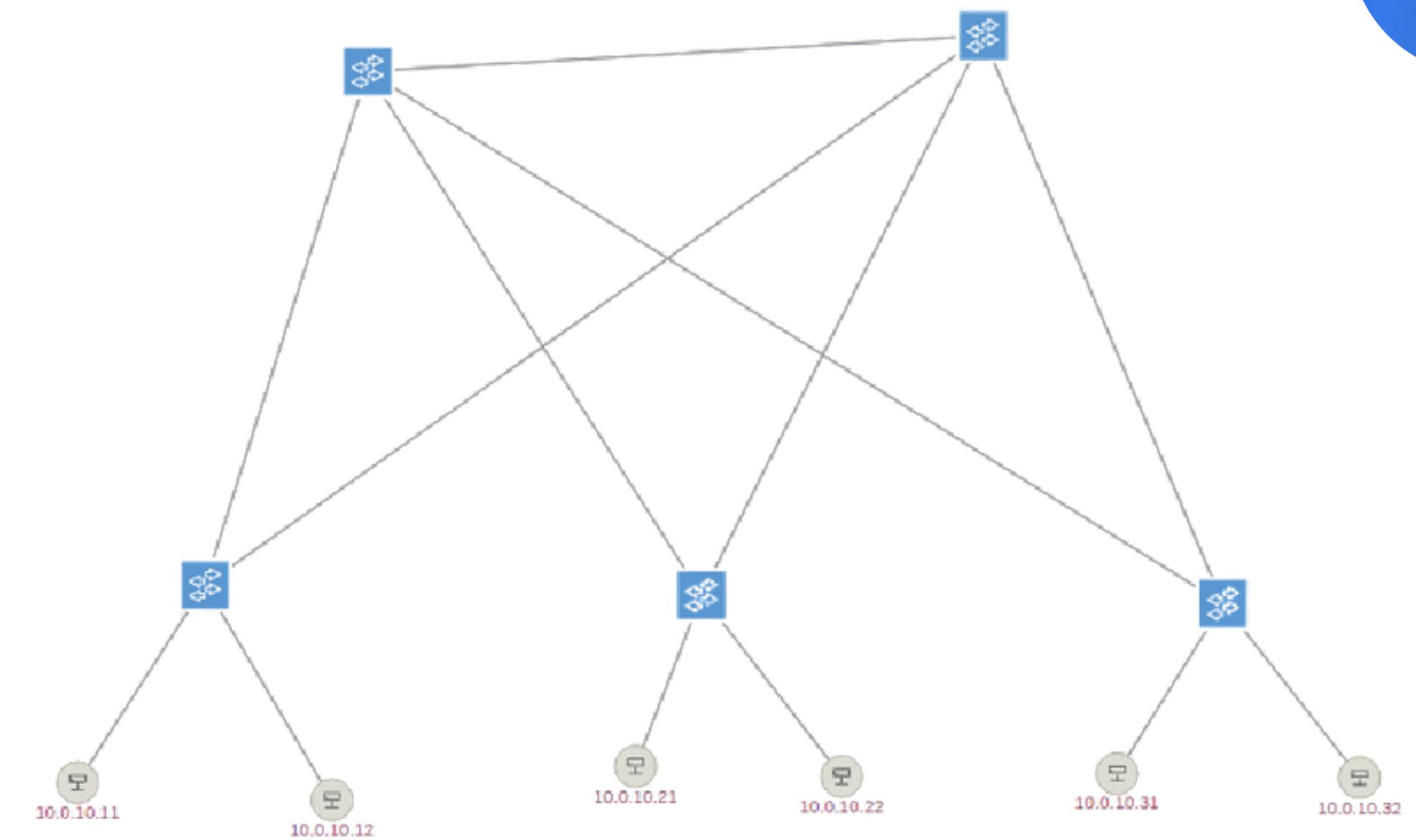
Mạng SDN đầu

- Một switch trung tâm và một switch dự phòng để nối vào ba switch “lá”.
- Mỗi switch lá sẽ được nối vào hai máy khách.
- Tất cả các máy khách sẽ đều thuộc cùng một mạng $10.0.10.0/24$



Mạng SDN đầu

- Chuyển tiếp giữa các gói tin trong mạng.
- Chặn một máy khách không được phép chủ động liên lạc tới máy khách khác.
- Chỉ cho phép một máy khách được liên lạc tới máy khách chỉ định.
- Kiểm tra việc chuyển tiếp gói tin bằng switch dự phòng.



Mạng SDN đầu

Kịch bản 1: Chuyển tiếp giữa các gói tin trong mạng

- Bộ điều khiển ONOS đã cung cấp sẵn ứng dụng để tiến hành điều khiển các thiết bị mạng chuyển tiếp gói tin đi khi có gói tin được chuyển tiếp lên bộ điều khiển. Lệnh kích hoạt:

```
onos> app activate org.onosproject.fwd
```

- Trên giao diện câu lệnh của Mininet, kiểm tra kết nối bằng lệnh pingall:

```
mininet> pingall
*** Ping: testing ping reachability
ha1 -> ha2 hb1 hb2 hc1 hc2
ha2 -> ha1 hb1 hb2 hc1 hc2
hb1 -> ha1 ha2 hb2 hc1 hc2
hb2 -> ha1 ha2 hb1 hc1 hc2
hc1 -> ha1 ha2 hb1 hb2 hc2
hc2 -> ha1 ha2 hb1 hb2 hc1
*** Results: 0% dropped (30/30 received)
```

Hình 8: Kết quả kịch bản đầu

Mạng SDN đầu

Kịch bản 2: Chặn một máy khách không được phép chủ động liên lạc tới máy khách khác

Kịch bản này sẽ sử dụng hai máy khách là ha1 và hb1, các gói tin từ máy khách ha1 chủ động gửi đi tới máy khách hb1 sẽ bị bỏ đi.

Để thực hiện kịch bản, tiến hành cài đặt một flow rule lên switch “lá”:

- Chỉ xử lý khi máy khách ha1 muốn liên lạc với máy khách hb1.
- Trong bối cảnh kịch bản, tiến hành lọc thêm là chỉ xử lý khi máy khách ha1 sử dụng giao thức ICMP Request.
- Không xử lý gói tin khi xác nhận đúng đối tượng, lúc đó gói tin sẽ bị bỏ đi.

Sau khi đã xây dựng flow rule xong, tiến hành đăng tải:

```
curl -u onos:rocks -X POST -H "Content-Type: application/json" -d @drop-ha1-to-hb1.json  
http://172.17.0.5:8181/onos/v1/flows/of:0000000000000003
```

Mạng SDN đầu

Kịch bản 2: Chặn một máy khách không được phép chủ động liên lạc tới máy khách khác

```
{  
    "priority": 50000,  
    "timeout": 0,  
    "isPermanent": true,  
    "deviceId": "of:0000000000000003",  
    "treatment": {  
        "instructions": []  
    },  
    "selector": {  
        "criteria": [  
            { "type": "ETH_TYPE", "ethType": "0x800" },  
            { "type": "IPV4_SRC", "ip": "10.0.10.11/32" },  
            { "type": "IPV4_DST", "ip": "10.0.10.21/32" },  
            { "type": "IP_PROTO", "protocol": "1" },  
            { "type": "ICMPV4_TYPE", "icmpType": "8" }  
        ]  
    }  
}
```

Mạng SDN đầu

Kịch bản 2: Chặn một máy khách không được phép chủ động liên lạc tới máy khách khác

Sau khi cài đặt thành công, kiểm tra kết quả bằng cách sử dụng lệnh pingall trên giao diện dòng lệnh của Mininet. Có thể thấy, liên lạc từ máy khách ha1 đến máy khách hb1 sẽ thất bại.

```
mininet> pingall
*** Ping: testing ping reachability
ha1 -> ha2 X hb2 hc1 hc2
ha2 -> ha1 hb1 hb2 hc1 hc2
hb1 -> hal ha2 hb2 hc1 hc2
hb2 -> ha1 ha2 hb1 hc1 hc2
hc1 -> ha1 ha2 hb1 hb2 hc2
hc2 -> ha1 ha2 hb1 hb2 hc1
*** Results: 3% dropped (29/30 received)
```

Hình 9: Kết quả kịch bản thứ hai

Mạng SDN đầu

Kịch bản 3: Chỉ cho phép một máy đích được liên lạc tới một máy đích chỉ định

Kịch bản này sẽ sử dụng hai máy khách chính là ha1 và hb2, các gói tin từ máy khách khác ha1 chủ động gửi tới máy khách hb2 sẽ bị bỏ đi.

Để thực hiện kịch bản, cần thiết phải cài đặt hai flow rule:

- Bỏ đi các gói tin khác chủ động gửi tới máy khách hb2.
- Chấp nhận gói tin đến từ máy khách ha1.

Tiến hành đăng tải hai flow rule đã xây dựng và kiểm tra bằng lệnh pingall trên Mininet:

```
mininet> pingall
*** Ping: testing ping reachability
ha1 -> ha2 hb1 hb2 hc1 hc2
ha2 -> ha1 hb1 X hc1 hc2
hb1 -> ha1 ha2 X hc1 hc2
hb2 -> ha1 ha2 hb1 hc1 hc2
hc1 -> ha1 ha2 hb1 X hc2
hc2 -> ha1 ha2 hb1 X hc1
*** Results: 13% dropped (26/30 received)
```

Hình 10: Kết quả kịch bản thứ ba

Mạng SDN đầu

Kịch bản 3: Chỉ cho phép một máy đích được liên lạc tới một máy đích chỉ định

Flow rule bỏ gói tin được gửi chủ động tới máy khách hb2

```
{  
    "priority": 50000,  
    "timeout": 0,  
    "isPermanent": true,  
    "deviceId": "of:0000000000000004",  
    "treatment": {  
        "instructions": []  
    },  
    "selector": {  
        "criteria": [  
            { "type": "ETH_TYPE", "ethType": "0x800" },  
            { "type": "IPV4_DST", "ip": "10.0.10.22/32" },  
            { "type": "IP_PROTO", "protocol": "1" },  
            { "type": "ICMPV4_TYPE", "icmpType": "8" }  
        ]  
    }  
}
```

Mạng SDN đầu

Kịch bản 3: Chỉ cho phép một máy đích được liên lạc tới một máy đích chỉ định

Flow rule chấp nhận gói tin do máy khách hal gửi

```
{  
    "priority": 51000,  
    "timeout": 0,  
    "isPermanent": true,  
    "deviceId": "of:0000000000000004",  
    "treatment": {  
        "instructions": [{ "type": "OUTPUT", "port": "3" }]  
    },  
    "selector": {  
        "criteria": [  
            { "type": "ETH_TYPE", "ethType": "0x800" },  
            { "type": "IPV4_SRC", "ip": "10.0.10.11/32" },  
            { "type": "IPV4_DST", "ip": "10.0.10.22/32" },  
            { "type": "IP_PROTO", "protocol": "1" },  
            { "type": "ICMPV4_TYPE", "icmpType": "8" }  
        ]  
    }  
}
```

Mạng SDN đầu

Kịch bản 4: Kiểm tra việc chuyển tiếp giữa các gói tin bằng switch dự phòng

Đầu tiên, tiến hành ngắt tất cả kết nối giữa các switch “lá” tới switch trung tâm:

```
mininet> link sa1 s0a down  
mininet> link sb1 s0a down  
mininet> link sc1 s0a down
```

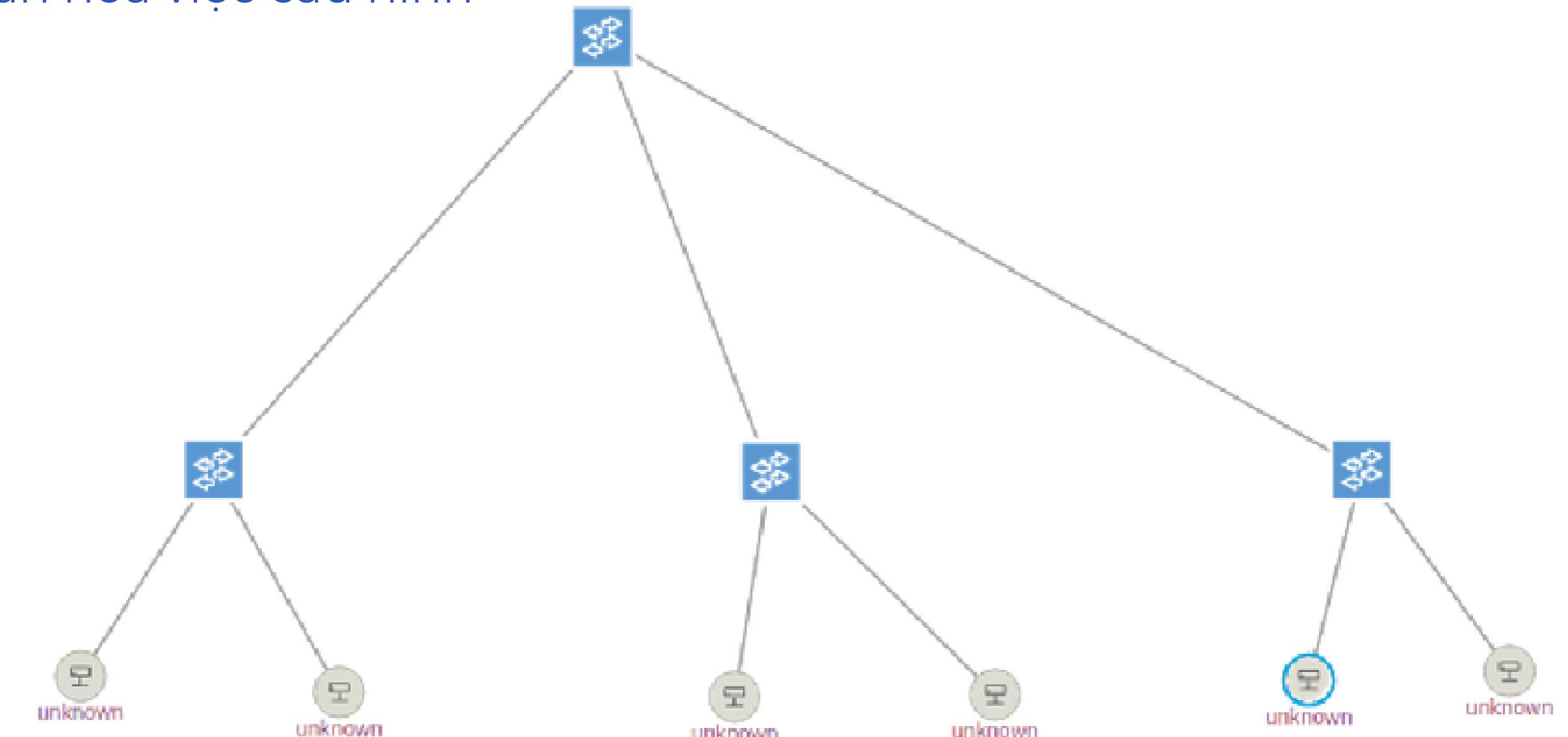
Kiểm tra kết nối bằng lệnh pingall trên Mininet:

```
mininet> link sa1 s0a down  
mininet> link sb1 s0a down  
mininet> link sc1 s0a down  
mininet> pingall  
*** Ping: testing ping reachability  
ha1 -> ha2 hb1 hb2 hc1 hc2  
ha2 -> ha1 hb1 hb2 hc1 hc2  
hb1 -> ha1 ha2 hb2 hc1 hc2  
hb2 -> ha1 ha2 hb1 hc1 hc2  
hc1 -> ha1 ha2 hb1 hb2 hc2  
hc2 -> ha1 ha2 hb1 hb2 hc1  
*** Results: 0% dropped (30/30 received)
```

Hình 12: Kết quả kịch bản thứ tư

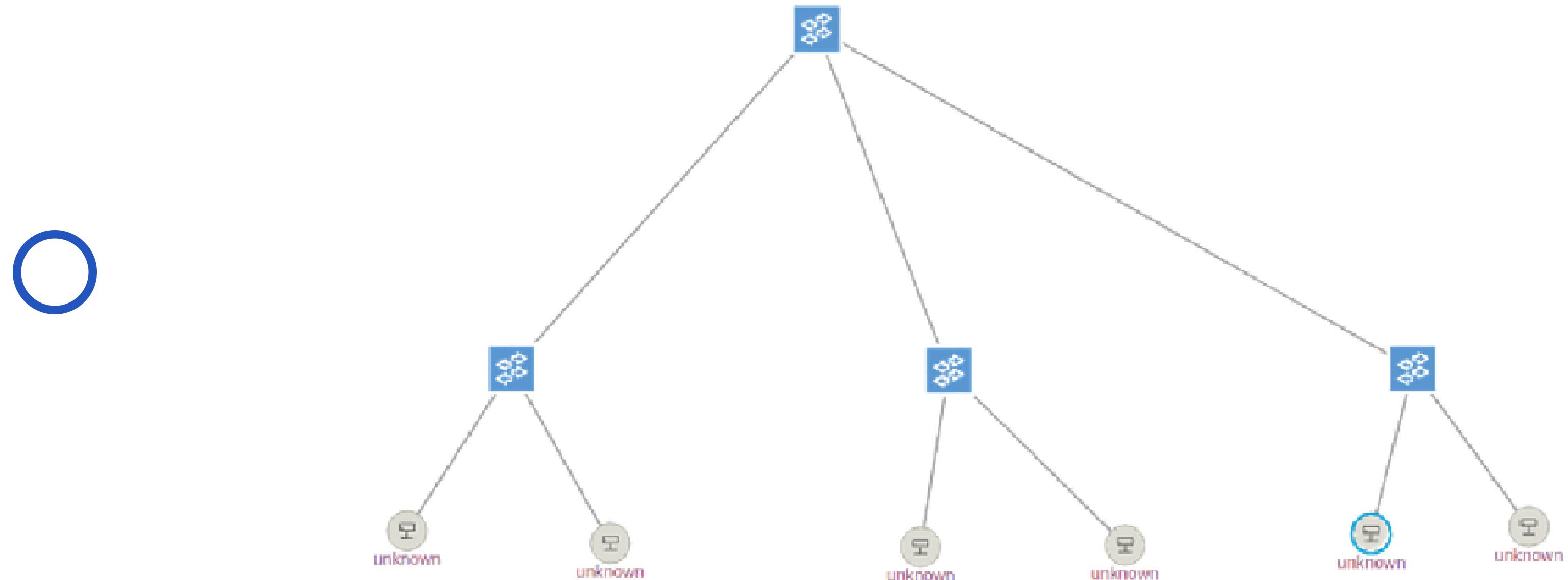
Mạng SDN thứ hai

- Mạng SDN thứ hai: Các switch “lá” bây giờ sẽ nối vào hai máy khách thuộc mạng khác nhau là 10.0.10.0/24 và 10.0.20.0/24 với VLAN lần lượt là 10 và 20. Mạng này sẽ không còn switch dự phòng để đơn giản hóa việc cấu hình



Mạng SDN thứ hai

- Định tuyến và chuyển tiếp các gói tin trong mạng.



Mạng SDN thứ hai

Vì không thể dựa vào ứng dụng chuyển tiếp như những kịch bản trước, nên với kịch bản này, cần thiết phải cấu hình thủ công các flow rule để các gói tin có thể đến đích. Việc cài đặt có thể chia làm các công việc chính như sau:

- Cấu hình cổng mặc định cho các máy khách tại switch “lá”. Cần thiết để định tuyến giữa các máy khách thuộc mạng khác nhau.
- Định nghĩa các hàm xử lý gói tin và flow rule đã xây dựng.
- Định tuyến giữa các máy khách khác mạng, nhưng liên kết trực tiếp cùng một switch “lá”.
- Định tuyến và chuyển tiếp giữa các máy khách liên kết trực tiếp khác switch “lá”.

Mạng SDN thứ hai

Cấu hình cổng mặc định cho các máy khách tại switch “lá”

Để cấu hình cổng mặc định, có thể khai báo cổng mặc định của máy khách khi khởi tạo các máy khách như sau

```
#Hosts (ha1, ha2, hb1, hb2, hc1, hc2)
ha1 = self.addHost('ha1', ip = '10.0.10.11/24', mac='00:00:00:00:01:01', defaultRoute='via 10.0.10.1')
ha2 = self.addHost('ha2', ip = '10.0.20.11/24', mac='00:00:00:00:02:01', defaultRoute='via 10.0.20.1')
...
...
```

Tuy nhiên, nếu chỉ vậy thì chưa đủ. Các máy khách cần phải biết địa chỉ MAC đích tương ứng với IP cổng mặc định là gì để có thể chuyển tiếp gói tin đi. Ở đây, có thể sử dụng địa chỉ MAC ảo bất kỳ cho các switch “lá”:

```
ha1.cmd('arp -s 10.0.10.1 00:00:00:00:00:99')
ha2.cmd('arp -s 10.0.20.1 00:00:00:00:00:99')
...
...
```

Mạng SDN thứ hai

Định nghĩa một số hàm xử lý gói tin và đăng tải flow rule đã xây dựng

Đầu tiên, cần định nghĩa một hàm xây dựng flow rule để xử lý gói tin. Nhiệm vụ của flow rule này chủ yếu là gán vào thẻ VLAN phù hợp, và chuyển tiếp cho flow table tiếp theo xử lý. Ở đây, quy ước là mạng 10.0.10.0/24 sẽ có VLAN 10, mạng 10.0.20.0/24 sẽ có VLAN 20.

```
def provision_ingress_rule(device_id, host_port, vlan_id):
    print(f"Configuring Ingress for Port {host_port} on VLAN {vlan_id}...")
    rule = {
        "priority": 40000, "isPermanent": True, "deviceId": device_id, "tableId": 0,
        "selector": {
            "criteria": [ {"type": "IN_PORT", "port": host_port} ]
        },
        "treatment": { "instructions": [
            {"type": "L2MODIFICATION", "subtype": "VLAN_PUSH"},
            {"type": "L2MODIFICATION", "subtype": "VLAN_ID", "vlanId": vlan_id},
            {"type": "TABLE", "tableId": 1}
        ]}
    }
    send_flow(device_id, rule)
```

Mạng SDN thứ hai

Định nghĩa một số hàm xử lý gói tin và đăng tải flow rule đã xây dựng

Ở cuối hàm này có gọi một hàm là send_flow(), hàm này sẽ đóng vai trò đăng tải flow rule đã xây dựng lên switch phù hợp. Hàm send_flow() cũng sẽ được sử dụng cho các hàm xây dựng flow rule khác:

```
def send_flow(device_id, flow_data):
    url = f'http://{{ONOS_IP}}: {{ONOS_PORT}}/onos/v1/flows/{{device_id}}'
    response = requests.post(url, auth=AUTH, data=json.dumps(flow_data), headers={'Content-Type': 'application/json'})
    if response.status_code not in [200, 201]:
        print(f" [FAIL] {device_id} Error: {response.text}")
```

Mạng SDN thứ hai

Định tuyến giữa các máy khách khác mạng, cùng switch “lá”

Khi đích đến là một máy khách cùng switch “lá”, hướng tiếp cận sẽ trở nên đơn giản hơn nhiều, khi mà ta chỉ cần bỏ hẳn thẻ VLAN và chuyển tiếp gói tin sang máy khách đó.

Ngoài ra, để cho đúng quy tắc chuyển tiếp, có thể điều chỉnh địa chỉ MAC nguồn thành địa chỉ MAC cổng mặc định đã định nghĩa trước đó.

Định tuyến giữa các máy khách khác mạng, cùng switch “lá” (tiếp theo)

```
def provision_intra_switch_route(  
    name,  
    device_id,  
    src_vlan,  
    dst_ip,  
    dst_mac,  
    dst_vlan,  
    out_port  
):  
    print(f"Configuring Local Route: {name}...")  
    rule = {  
        "priority": 40000, "isPermanent": True, "deviceId": device_id, "tableId": 1,  
        "selector": {  
            "criteria": [  
                {"type": "IPV4_DST", "ip": dst_ip},  
                {"type": "ETH_TYPE", "ethType": "0x0800"},  
                {"type": "VLAN_VID", "vlanId": src_vlan}  
            ]  
        },  
        "treatment": { "instructions": [  
            {"type": "L2MODIFICATION", "subtype": "ETH_DST", "mac": dst_mac},  
            {"type": "L2MODIFICATION", "subtype": "ETH_SRC", "mac": ROUTER_MAC},  
            {"type": "L2MODIFICATION", "subtype": "VLAN_ID", "vlanId": dst_vlan},  
            {"type": "L2MODIFICATION", "subtype": "VLAN_POP"},  
            {"type": "OUTPUT", "port": out_port}  
        ]  
    }
```

Mạng SDN thứ hai

Định tuyến và chuyển tiếp giữa các máy khách khác switch

Về cơ bản, việc chuyển tiếp và định tuyến đều đi qua các bước giống nhau:

- Chuyển tiếp gói tin từ switch “lá” lên switch trung tâm.
- Switch trung tâm đọc thông tin gói tin và quyết định nên chuyển tiếp tới switch “lá” nào.
- Switch “lá” đích nhận gói tin, kiểm tra gói tin có đúng không, bỏ thẻ VLAN và chuyển tiếp về máy đích.

Định nghĩa flow rule chuyển tiếp lên switch trung tâm với máy khách cùng mạng:

```
rule_src = {
    "priority": 40000, "isPermanent": True, "deviceId": src_leaf_id, "tableId": 1,
    "selector": {
        "criteria": [
            {"type": "ETH_DST", "mac": dst_mac},
            {"type": "VLAN_VID", "vlanId": vlan_id}
        ]
    },
    "treatment": { "instructions": [
        {"type": "OUTPUT", "port": src_uplink}
    ]}
}
```

Mạng SDN thứ hai

Định tuyến và chuyển tiếp giữa các máy khách khác switch (tiếp theo)

Định nghĩa flow rule chuyển tiếp gói tin lên switch trung tâm với máy khách khác mạng;

```
rule_src = {
    "priority": 41000,
    "isPermanent": True, "deviceId": src_leaf_id, "tableId": 1,
    "selector": {
        "criteria": [
            {"type": "ETH_DST", "mac": ROUTER_MAC},
            {"type": "IPV4_DST", "ip": dst_ip},
            {"type": "ETH_TYPE", "ethType": "0x0800"},
            {"type": "VLAN_VID", "vlanId": src_vlan}
        ]
    },
    "treatment": { "instructions": [
        {"type": "L2MODIFICATION", "subtype": "ETH_SRC", "mac": ROUTER_MAC},
        {"type": "L2MODIFICATION", "subtype": "ETH_DST", "mac": dst_mac},
        {"type": "L2MODIFICATION", "subtype": "VLAN_ID", "vlanId": dst_vlan},
        {"type": "OUTPUT", "port": src_uplink}
    ]}
}
```

Mạng SDN thứ hai

Định tuyến và chuyển tiếp giữa các máy khách khác switch (tiếp theo)

Flow rule sử dụng để chuyển tiếp từ switch trung tâm đến switch “lá” đích, và từ switch “lá” đích tới máy khách đích cho việc chuyển tiếp và định tuyến đều giống nhau.

- Switch trung tâm sẽ lọc switch “lá” đúng và chuyển tiếp tới đó. Định nghĩa flow rule chuyển tiếp tới switch “lá” đích đúng như sau:

```
rule_spine = {
    "priority": 40000, "isPermanent": True, "deviceId": spine_id,
    "tableId": 0,
    "selector": {
        "criteria": [ {"type": "ETH_DST", "mac": dst_mac} ]
    },
    "treatment": { "instructions": [
        {"type": "OUTPUT", "port": spine_downlink}
    ]}
}
```

Mạng SDN thứ hai

Định tuyến và chuyển tiếp giữa các máy khách khác switch (tiếp theo)

- Switch “lá” đích sẽ bỏ thẻ VLAN đi và chuyển tiếp tới máy đích đúng. Định nghĩa flow rule chuyển tiếp tới máy đích đúng như sau:

```
rule_dst = {
    "priority": 40000, "isPermanent": True, "deviceId": dst_leaf_id, "tableId": 0,
    "selector": {
        "criteria": [
            {"type": "ETH_DST", "mac": dst_mac},
            {"type": "VLAN_VID", "vlanId": dst_vlan}
        ]
    },
    "treatment": { "instructions": [
        {"type": "L2MODIFICATION", "subtype": "VLAN_POP"},
        {"type": "OUTPUT", "port": dst_host_port}
    ]}
}
```

Mạng SDN thứ hai

Đăng tải flow rule và kết quả

Để tất cả các máy khách có thể liên lạc được với nhau, trong chương trình được soạn sẵn sẽ thực hiện đăng tải các flow rule theo từng cặp máy khách cụ thể (chiều xuôi và ngược) như sau:

- Định tuyến giữa các máy khách khác mạng, cùng switch “lá”:
 1. Máy khách ha1 và ha2.
 2. Máy khách hb1 và hb2.
 3. Máy khách hc1 và hc2.

```
provision_intra_switch_route(  
    name="ha1 to ha2",  
    device_id=DEV_SA1,  
    src_vlan=10,  
    dst_ip="10.0.20.11/32",  
    dst_mac="00:00:00:00:02:01",  
    dst_vlan=20,  
    out_port=3  
)
```

Mạng SDN thứ hai

Đăng tải flow rule và kết quả

- Chuyển tiếp giữa các máy khách cùng mạng, khác switch “lá”:
 1. Máy khách ha1 và hb1.
 2. Máy khách ha1 và hc1.
 3. Máy khách hb1 và hc1.
 4. Máy khách ha2 và hb2.
 5. Máy khách ha2 và hc2.
 6. Máy khách hb2 và hc2.

```
provision_l2_remote_forwarding(  
    name="L2: ha1->hb1",  
    src_leaf_id=DEV_SA1,  
    dst_leaf_id=DEV_SB1,  
    spine_id=DEV_S0A,  
    dst_mac=DEV_HB1,  
    vlan_id=10,  
    src_uplink=1,  
    spine_downlink=3,  
    dst_host_port=2  
)
```

Mạng SDN thứ hai

Đăng tải flow rule và kết quả

- Định tuyến giữa các máy khách khác mạng, khác switch “lá”:

- Máy khách ha1 và hb2.
- Máy khách ha1 và hc2.
- Máy khách ha2 và hb1.
- Máy khách ha2 và hc1.
- Máy khách hb1 và hc2.
- Máy khách hb2 và hc1.

```
provision_l3_remote_routing(  
    name="L3: ha1->hb2",  
    src_leaf_id=DEV_SA1,  
    dst_leaf_id=DEV_SB1,  
    spine_id=DEV_S0A,  
    src_vlan=10,  
    dst_vlan=20,  
    dst_ip="10.0.20.12/32",  
    dst_mac=DEV_HB2,  
    src_uplink=1,  
    spine_downlink=3,  
    dst_host_port=3  
)
```

Mạng SDN thứ hai

Đăng tải flow rule và kết quả

- Thực hiện lệnh pingall trên Mininet để kiểm tra cấu hình cho thấy đã thành công chuyển tiếp và định tuyến tất cả gói tin giữa các máy khách, cùng mạng hay khác mạng bên trong mạng SDN đã triển khai.

```
mininet> pingall
*** Ping: testing ping reachability
ha1 -> ha2 hb1 hb2 hc1 hc2
ha2 -> ha1 hb1 hb2 hc1 hc2
hb1 -> ha1 ha2 hb2 hc1 hc2
hb2 -> ha1 ha2 hb1 hc1 hc2
hc1 -> ha1 ha2 hb1 hb2 hc2
hc2 -> ha1 ha2 hb1 hb2 hc1
*** Results: 0% dropped (30/30 received)
```

Hình 14: Kết quả kịch bản thứ năm

Demo



Thank you
for
listening

