Giao thức trong mạng SDN: Overflow

Status	Idea
Date Added	@September 19, 2023 4:32 PM
Live Date	@March 20, 2023
Sponsor	
Tags	Information

🏆 **GOAL** | Khái niệm tổng quan

SCRIPT

Summary

Main contents

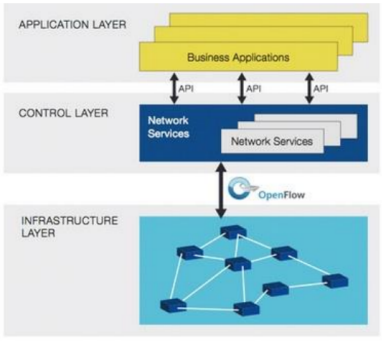
1. Định nghĩa

- **Forwarding Plane:**
    - vận chuyển dữ liệu từ nguồn đến đích trong mạng (chuyển tiếp gói dữ liệu, định tuyến, và chuyển đổi thông tin địa chỉ)
    - xử lý các gói dữ liệu trên cơ sở của quy tắc và quyết định được cấu hình trước trên thiết bị mạng
    - không có khả năng thay đổi hoặc quản lý cấu hình mạng.
- **Control Plane:**

- quản lý và điều khiển cách mạng hoạt động.
- quyết định cách mà dữ liệu nên được chuyển tiếp trong mạng và cấu hình các thông tin địa chỉ cần thiết cho forwarding plane.
- quản lý bảng định tuyến, cấu hình giao thức mạng, quảng bá thông tin địa chỉ, và thực hiện các chức năng điều khiển khác để đảm bảo hoạt động hiệu quả của mạng.

- **Mạng truyền thống:** control plane và forwarding plane thường làm việc cùng nhau trên cùng một thiết bị mạng, ví dụ như router.
- **Mạng SDN:** tách riêng các control plane phân tán từ các forwarding plane và đưa các chức năng của control plane vào trong control plane tập trung.

1.1 Cấu hình: 3 lớp riêng biệt



- **Lớp ứng dụng:**
  - ứng dụng kinh doanh được triển khai trên mạng
  - kết nối tới lớp controller thông qua các API
  - cho phép lập trình lại (cấu hình lại) mạng (điều chỉnh các tham số trễ, băng thông, định tuyến, …) thông qua lớp controller
- **Controller:**
  - thống kế tất cả thông tin từ các flow switch ở trong mạng, cung cấp cái nhìn tổng thể bên trong của network
  - cấu hình mạng theo các yêu cầu từ lớp ứng dụng và khả năng của mạng
    - sử dụng các cơ chế như OpenFlow, ONOS, ForCES, PCEP, NETCONF, SNMP hoặc thông qua các cơ chế riêng biệt

Lợi ích:

- Về kiến trúc:
  - Lớp controller:
    - lập trình trực tiếp
    - điều chỉnh, thay đổi một cách nhanh chóng
    - quản lý tập trung do phần controller

được tập trung

    - Cấu hình lớp cơ sở hạ tầng có thể được lập trình trên lớp ứng dụng và truyền đạt xuống các lớp dưới.

- Về tính năng:
    - Giảm CapEx:
        - giảm thiểu các yêu cầu mua phần cứng
        - hỗ trợ mô hình pay-as-yougrow
    - Giảm OpEx:
        - dễ dàng thiết kế, triển khai, quản lý và mở rộng mạng
        - giảm thời gian quản lý tổng thể, giảm xắc suất lỗi
    - Truyền tải nhanh chóng và linh hoạt:
        - tạo mới các kiểu ứng dụng, dịch vụ và mô hình kinh doanh

- **OpenFlow**: cho phép các thiết bị mạng như switch và router thực hiện chuyển tiếp dựa trên hướng dẫn từ một controller trung tâm. ⇒ thiết lập luồng dữ liệu (flows) và thay đổi bảng định tuyến
- **ONOS (Open Network Operating System)**: một controller SDN được phát triển để quản lý và điều khiển các mạng SDN phức tạp ⇒ quản lý mạng, cân bằng tải, và tự động hóa trong môi trường SDN.
- **ForCES (Forwarding and Control Element Separation)**: tách riêng control plane và forwarding plane
    - thành phần của mạng hoạt động độc lập với nhau và cung cấp một cách để quản lý chúng bằng cách sử dụng giao thức ForCES
    - Giao thức ForCES được sử dụng để truyền thông giữa CE và FE, cho phép CE điều khiển FE một cách tập trung. CE có thể cấu hình FE để thực hiện các nhiệm vụ cụ thể như định tuyến, lọc gói dữ liệu, hay chuyển tiếp dựa trên các hướng dẫn từ CE. ⇒ tách biệt logic điều khiển từ logic chuyển tiếp dữ liệu, giúp tăng tính linh hoạt và quản lý trong mạng
- **PCEP (Path Computation Element Protocol)**: truyền thông giữa các

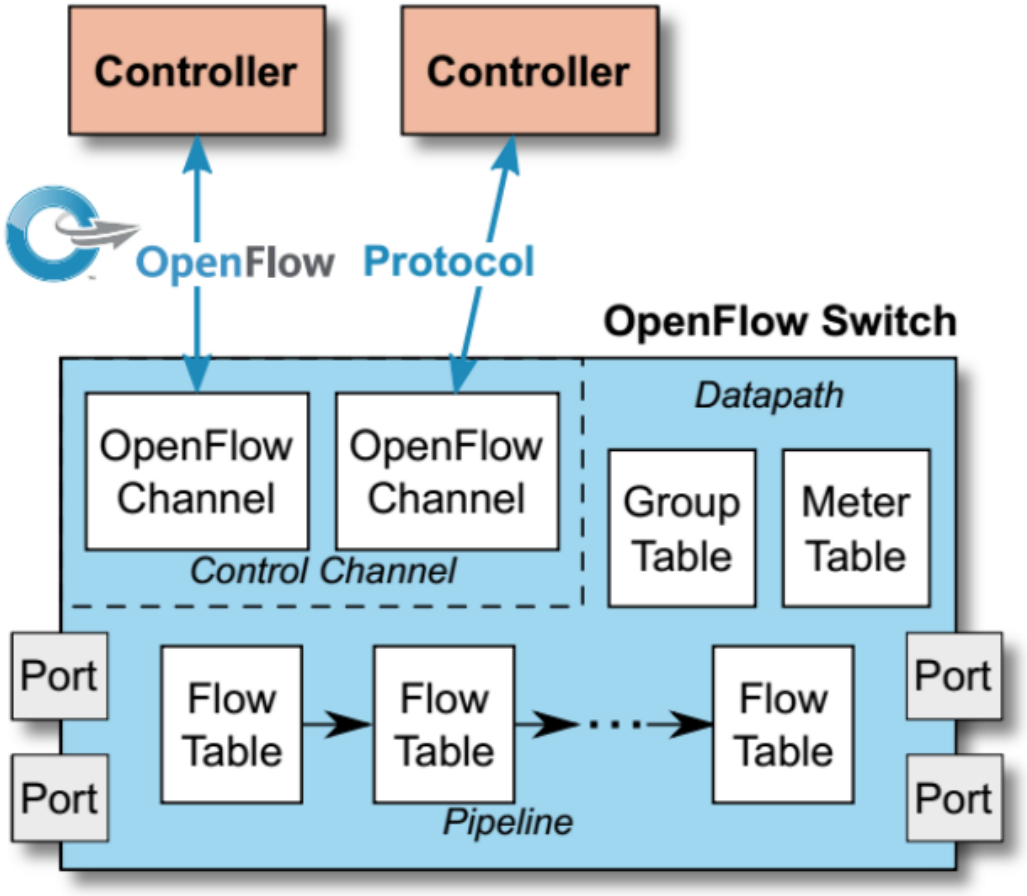
Path Computation Elements (PCEs)

- quản lý tài nguyên và tính toán đường dẫn trong mạng SDN.
- PCEs hợp tác để tính toán và thiết lập đường dẫn tối ưu cho dịch vụ mạng.

- **NETCONF (Network Configuration Protocol)**: quản lý mạng dựa trên XML được sử dụng để cấu hình và quản lý các thiết bị mạng
  - controller hoặc quản trị viên mạng cấu hình và thay đổi các thiết lập trên thiết bị mạng từ xa một cách tự động và đáng tin cậy.
- **SNMP (Simple Network Management Protocol)**: theo dõi và quản lý các thiết bị mạng.
  - thu thập thông tin về trạng thái và hiệu suất của các thiết bị mạng và thực hiện các thao tác quản lý trên chúng.

- **Lớp cơ sở hạ tầng:** là các thiết bị mạng thực tế (vật lý hay ảo hóa)
  - chuyển tiếp gói tin theo sự controller của lớp điều khiển
  - Một thiết bị mạng có thể hoạt động theo sự controller của nhiều bộ controller khác nhau ⇒ tăng cường khả năng ảo hóa của mạng.

2. Giới thiệu Switch OpenFlow



2.1 Các thành phần Switch

- **Switch Logic OpenFlow:** Switch giao tiếp với bộ controller và bộ quản lý controller qua giao thức OpenFlow Switch Protocol.
  - một hoặc nhiều bảng flow
  - một bảng group: tra cứu và chuyển tiếp gói tin
  - một hoặc nhiều kênh OpenFlow: tới bộ controller bên ngoài.
- **OpenFlow Switch Protocol:**
  - Controller có thể thêm, cập nhật và xóa các mục flow trong các bảng flow

- Mỗi bảng flow trong switch: một tập các mục flow(mỗi mục flow - trường *match fields, counters* và một bộ *instruction* để áp dụng cho các gói tin tương ứng)
- **Gói tin:** tìm mục flow phù hợp theo thứ tự ưu tiên
  - tìm thấy:
    - Instruction: các hướng dẫn sẽ được thực hiện
    - Các action trong các instruction: mô tả việc chuyển tiếp gói, chỉnh sửa gói tin & xử lý bảng group.
  - không tìm thấy: kết quả phụ thuộc vào cấu hình của mục flow *table-miss.*
- Quá trình xử lý pineline:
  - Các *instruction* gửi các gói tin tới các bảng tiếp theo để xử lý tiếp và cho phép trao đổi thông tin giữa các bảng dưới dạng *metadata*.
  - Dừng lại khi bộ instruction của mục Flow phù hợp không chỉ định một bảng tiếp theo (gói thường được chỉnh sửa và chuyển tiếp)
  - Các mục flow có thể chuyển tiếp gói tin đến một cổng (vật lý, logic hoặc reserved)
    - **Cổng reserved:** tác vụ chuyển tiếp như gửi gói tin đến bộ controller, quảng bá gói tin toàn mạng (flooding) hoặc chuyển tiếp bằng các phương pháp khác không phải OpenFlow, chẳng hạn như các switch bình thường
    - **Cổng logic:** chuyển tiếp gói tin tới các nhóm link *aggregation, tunnels* hoặc các *loopback interfaces.*
      - **Aggregation (Liên kết nhiều kết nối)**: kết hợp nhiều liên kết hoặc kết nối mạng vật lý thành một liên kết logic hoặc giao diện.
        - cổng logic "aggregation": kết hợp lưu lượng từ nhiều kết nối để tăng băng thông hoặc cung cấp tính sẵn sàng dự phòng.
      - **Tunnels (Đường hầm)**: truyền tải lưu lượng mạng qua mạng vật lý hoặc mạng có định dạng khác.
        - bảo mật lưu lượng mạng hoặc để tạo ra kết nối ảo giữa hai điểm trong mạng mà có thể nằm xa nhau về mặt vật lý

        - Cổng logic có thể được sử dụng để đại diện cho một đầu mút của một đường hầm và điều hướng lưu lượng qua đó.
    - **Loopback Interfaces (Giao diện Loopback)**: giao diện ảo trong mạng được sử dụng để kiểm tra sự hoạt động của chính thiết bị hoặc dịch vụ mạng.
        - thiết lập để chỉ trỏ về chính thiết bị mà nó thuộc về
        - Cổng logic có thể đại diện cho một giao diện loopback và được sử dụng để kiểm tra và quản lý mạng hoặc dịch vụ.
- Các action của mục flow cũng có thể chuyển gói tin tới một group (action quảng bá toàn mạng)
    - chuyển tiếp phức tạp hơn như *multipath, fast reroute* và *link aggregation*
        - **Multipath:** chia lưu lượng mạng ra nhiều đường dẫn khác nhau giữa hai điểm kết nối ⇒ tận dụng nhiều đường dẫn để cải thiện hiệu suất và sẵn sàng của mạng.
            - **Group action:** chuyển tiếp gói tin đến nhiều đường dẫn khác nhau dựa trên thuật toán cân bằng tải hoặc các quy tắc khác để tối ưu hóa việc sử dụng các đường dẫn.
        - **Fast Reroute (Tái định tuyến nhanh)**: đảm bảo sự liên tục và sẵn sàng của dịch vụ mạng trong trường hợp sự cố xảy ra, chẳng hạn như một kết nối mạng bị hỏng.
            - **Group action:** sử dụng các đường dẫn phụ thay thế để chuyển tiếp gói tin nhanh chóng mà không gây gián đoạn đáng kể cho dịch vụ.
        - **Link Aggregation (Liên kết gom nhóm)**: kết hợp nhiều kết nối mạng vật lý thành một liên kết logic có băng thông lớn hơn và tính sẵn sàng cao hơn.
            - **Group action:** quản lý và điều hướng lưu lượng trên các liên kết trong nhóm. Khi gặp sự cố, lưu lượng có thể được chuyển tiếp tự động sang các liên kết khác trong nhóm để đảm bảo tính sẵn sàng và tăng băng thông.

- **Các group:** nhiều mục flow chuyển tiếp tới một thiết bị duy nhất (ví dụ như chuyển tiếp IP đến một next hop).
    - Next hop: địa chỉ IP hoặc địa chỉ mạng của thiết bị tiếp theo (hoặc điểm kết nối tiếp theo) mà gói tin mạng sẽ được chuyển đến để tiếp tục hành trình của nó.
        - Next hop thường được sử dụng trong quá trình định tuyến để xác định cách một gói tin mạng sẽ được chuyển tiếp từ nguồn đến đích qua các thiết bị mạng trung gian.
        - Quá trình: Nhận gói tin - xác định next hop - Chuyển tiếp gói tin - Lặp lại quá trình.
        - cho phép gói tin đi qua nhiều thiết bị mạng trung gian để đến được đích cuối cùng.
- **Bảng group:** chứa các mục group
    - **Group:** danh sách các *action bucket* với ý nghĩa cụ thể phụ thuộc vào loại group
    - **Action:** một hoặc nhiều *bucket* được áp dụng cho các gói được gửi đến group

2.2 Cổng OpenFlow

- **Cổng OpenFlow:** các *network interface* truyền các gói giữa quá trình xử lý OpenFlow và phần còn lại của mạng.
- Thiết bị switch: kết nối logic với nhau thông qua các cổng OpenFlow
- Các gói tin được tiếp nhận trên một *ingress port,* xử lý bởi OpenFlow pipeline
- **Ingress port:** so khớp với các gói tin.
- **Action output:** chuyển gói tin trên một cổng đầu ra trong quá trình xử lý pipeline, định nghĩa cách gói trở lại mạng.
- Bộ switch OpenFlow hỗ trợ ba loại cổng OpenFlow: cổng vật lý, cổng logic và cổng reserved.

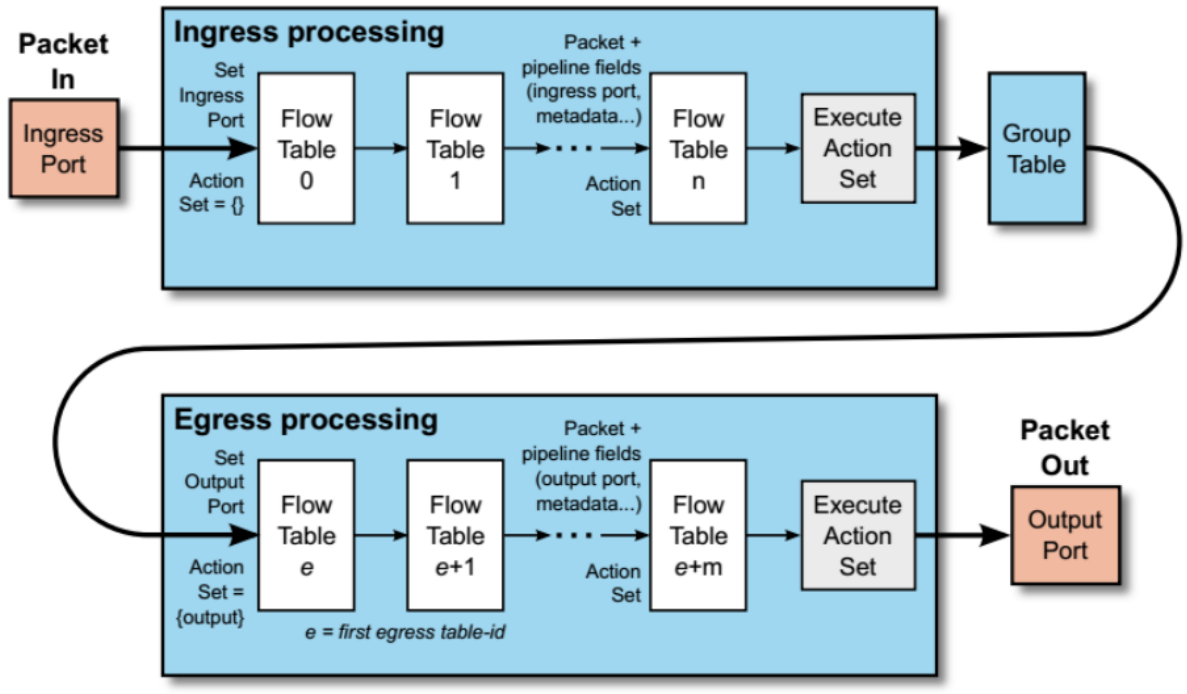
💡 **Cổng reserved OpenFlow:** Switch không yêu cầu hỗ trợ tất cả các cổng reserved, trừ những cổng được đánh dấu "bắt buộc":

- **ALL (bắt buộc):** Đại diện cho tất cả các cổng mà switch có thể sử dụng để chuyển tiếp một gói cụ thể ⇒ Sử dụng như một cổng đầu ra.
  - một bản sao của gói bắt đầu quá trình xử lý egress trên tất cả các cổng, ngoại trừ
    cổng ingress (nơi gói tin ban đầu vào switch) và các cổng được cấu hình OFPPC_NO_FWD.
  - **Xử lý egress:**
    - xử lý gói tin khi nó rời khỏi một thiết bị mạng hoặc switch và được chuyển đi đến điểm đích cuối.
    - egress processing: định tuyến (next hop), chuyển tiếp, cấu hình định tuyến lại (re-routing), quá trình cuối cùng trước khi đến đích (kiểm tra bảo mật, ACL, xử lý dịch vụ mạng cuối cùng)
    - OFPPC_NO_FWD (No Forwarding): cổng mà switch không chuyển tiếp gói tin ra ngoài mạng. ⇒ sử dụng cho các mục đích đặc biệt như quản lý switch hoặc kết nối với controller, không tham gia quá trình chuyển tiếp thông thường của gói tin mạng.
- **CONTROLLER (bắt buộc):** Biểu diễn kênh controller với bộ controller OpenFlow.
  - Sử dụng như cổng vào/ ra.
  - **Cổng ra:** đóng gói gói tin trong bản tin *packet-in* và gửi nó bằng cách sử dụng giao thức OpenFlow
- **TABLE (bắt buộc):** Biểu thị sự bắt đầu của quá trình pipeline.
  - Chỉ có giá trị trong action output trong danh sách các action của một thông báo packet-out và gửi gói tin tới bảng Flow đầu tiên để gói có thể được xử lý thông qua pipeline
    OpenFlow.
- **IN PORT (bắt buộc):** Biểu hiện cổng vào của gói tin. (cổng đầu ra)

- **ANY (bắt buộc):** sử dụng trong một số yêu cầu OpenFlow khi không có cổng nào được chỉ định ⇒ Không thể được sử dụng như một cổng vào hay cổng đầu ra.
- **UNSET (bắt buộc):** Giá trị đặc biệt để xác định rằng cổng đầu ra chưa được đặt trong Action-Set.
    - Action-Set: tập hợp các hành động mà switch hoặc controller có thể thực hiện trên một gói tin mạng khi nó đi qua switch.
        - Hành động: định tuyến, chuyển tiếp đến cổng cụ thể, chuyển tiếp đến group, hoặc thậm chí gửi đến controller.
    - Kết hợp cổng đầu ra trong bộ action sử dụng trường so khớp OXM_OF_ACTSET_OUTPUT.
        - **OXM_OF_ACTSET_OUTPUT**: cho phép controller xác định cổng đầu ra trong Action-Set để chỉ định gói tin nên được chuyển tiếp đến cổng nào sau khi xử lý.
    - Không thể được sử dụng như một cổng vào hay cổng đầu ra.
- **LOCAL (không bắt buộc):** Biểu diễn ngăn xếp mạng cục bộ và ngăn xếp quản lý.
    - Sử dụng như cổng vào/ ra
    - Cho phép các thực thể từ xa tương tác với switch và các dịch vụ mạng của nó thông qua mạng OpenFlow, chứ không phải qua một mạng controller riêng biệt.
    - **Bộ flow thích hợp:** thực hiện kết nối bộ controller trong băng
- **NORMAL (không bắt buộc):** Đại diện cho việc chuyển tiếp sử dụng pipeline không phải là OpenFlow truyền thống của switch.
    - Sử dụng như cổng đầu ra và xử lý các gói tin bằng cách sử dụng pipeline bình thường.
    - Nối cầu hoặc định tuyến gói tin
    - Switch không thể chuyển tiếp các gói tin từ pipeline dẫn OpenFlow đến pipeline bình thường: chỉ ra rằng nó không hỗ trợ action này

- **FLOOD (không bắt buộc):** Đại diện cho quảng bá gói tin không sử dụng pipeline, sẽ gửi gói ra tất cả các cổng, trừ cổng gói tin vào và các cổng ở trạng thái OFPPS_BLOCKED.
    - **OFPPS_BLOCKED (Blocked Port Status)**: trạng thái không thể chuyển tiếp gói tin mạng, ngăn chặn việc lặp gói tin khi có các vòng lặp trong mạng (loop prevention).
    - Có thể sử dụng VLAN ID hoặc các tiêu chí khác để chọn cổng sử dụng cho quảng bá.
        - **VLAN ID (Virtual LAN Identifier):** một số nguyên dương đại diện cho một mạng ảo cụ thể trong mạng vật lý.
            - tách mạng vật lý thành nhiều mạng ảo riêng biệt để tăng tính riêng tư, bảo mật và quản lý.
            - xác định cổng nào nên được sử dụng cho việc quảng bá gói tin trong mạng VLAN cụ thể.

2.3 Bảng OpenFlow



7

2.3.1 Xử lý pipeline

- Thiết bị switch hỗ trợ OpenFlow gồm hai loại:
  - **OpenFlow-only:** chỉ hỗ trợ OpenFlow, tất cả các gói tin được xử lý bằng pipeline OpenFlow.
  - **OpenFlow-hybrid:** hỗ trợ cả OpenFlow và switch Ethernet bình thường
- **Pipeline OpenFlow (Logic OpenFlow):** chứa một hoặc nhiều bảng flow, mỗi bảng flow chứa nhiều mục flow.
  - xác định cách các gói tin tương tác với các bảng flow này
  - Switch OpenFlow có ít nhất một bảng *flow ingress* (bảng định tuyến vào).
    - **Flow ingress:** gói tin mạng được xử lý khi chúng nhập vào switch từ một cổng vào (ingress port).
    - **Bảng flow:** đánh số theo thứ tự bắt đầu từ 0
    - **Xử lý pipeline:** hai giai đoạn, xử lý ingress và xử lý egress.

- Phân chia hai giai đoạn: chỉ ra bởi bảng flow ra đầu tiên (bảng có số thứ tự thấp hơn so với bảng egress đầu tiên ⇒ sử dụng như các bảng ingress)
- **Flow ingress:** sử dụng tùy thuộc vào kết quả của việc so khớp trong bảng đầu tiên.
- **Xử lý ingress:** chuyển gói tin tới một cổng đầu ra → switch có thể thực hiện việc xử lý egress trong cổng đầu ra đó
- **Xử lý exgress:** không bắt buộc, switch có thể không có hoặc không được cấu hình để sử dụng.
    - **Không có bảng egress hợp lệ** (cấu hình như bảng egress đầu tiên): gói tin được xử lý bởi cổng đầu ra. ⇒ hầu hết chuyển tiếp ra khỏi switch.
    - **Có bảng egress hợp lệ:** gói tin phải được so khớp với các mục flow của bảng flow đó
        - **Tìm thấy mục flow:** tập instruction trong mục flow đó được thực hiện
            - chỉ có thể trỏ một gói tin đến bảng Flow có số thứ tự lớn hơn số thứ tự của bảng đó
            - **Instruction:** có thể trỏ trực tiếp gói tin tới một bảng flow khác
            - **Không trỏ đến một bảng flow khác:** giai đoạn hiện tại của quá trình xử lý pipeline dừng lại ở bảng này (gói tin xử lý với các action liên quan)
        - **Không tìm thấy mục flow:** table-miss.
            - **Action:** phụ thuộc vào cấu hình.
            - **Instruction:** xác định cách xử lý gói tin chưa được so khớp (bỏ qua gói tin, chuyển chúng đến một bảng khác hoặc gửi chúng tới bộ controller qua kênh controller thông qua bản tin packet-in)

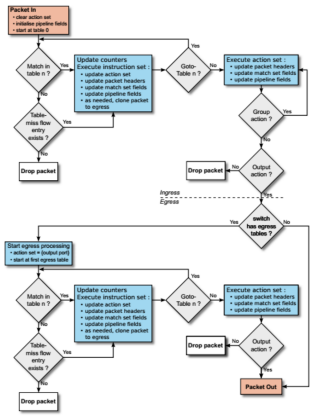
2.3.2 Bảng flow và mục flow

- Các mục flow:

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags

  - **Match Fields:** trường so khớp với các gói tin. Bao gồm cổng vào, tiêu đề gói, và các trường pipeline khác tùy chọn như metadata được chỉ định bởi bảng trước đó.
  - **Priority:** thứ tự ưu tiên của mục flow.
  - **Counters:** bộ đếm được cập nhật khi các gói tin được so khớp.
  - **Instructions:** các hướng dẫn chỉnh sửa bộ action hoặc xử lý pipeline.
  - **Timeouts:** thời gian tối đa hoặc thời gian chờ trước khi luồng hết hạn.
  - **Cookie:** giá trị dữ liệu được chọn bởi bộ controller. Có thể được sử dụng bởi bộ controller để lọc các mục flow bị ảnh hưởng bởi số liệu thống kê, chỉnh sửa và yêu cầu xóa flow. Trường này không được sử dụng khi xử lý các gói tin.
  - **Flags:** cờ thay đổi cách quản lý flow.
- **Mục flow:** xác định bởi các trường *match fields* và *priority*
  - **Table miss:** flow mà wildcards của tất cả các trường (tất cả các trường bị bỏ qua) và piority = 0

2.3.3 So khớp



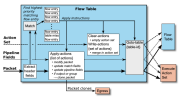
- Trường tiêu đề được tách ra từ gói tin và trường pipeline được lấy lại.
  - **Trường tiêu đề:** tra cứu bảng phụ thuộc vào loại gói tin, VD địa chỉ nguồn Ethernet hoặc địa chỉ đích IPv4.
- So khớp cũng có thể được thực hiện đối với cổng nhập, trường siêu dữ liệu (metadata) và các trường pipeline khác
  - **Siêu dữ liệu:** truyền thông tin giữa các bảng trong một switch

- **Lệnh Apply-Actions:** action được áp dụng trong một phần trước đó ⇒ thay đổi này được phản ánh trong các trường tiêu đề gói tin và trường pipeline.
- **Gói tin khớp với một mục flow:** tất cả các trường match fields của nó khớp với các trường tiêu đề và trường pipeline tương ứng của gói tin.
    - **Match fields bị bỏ qua trong mục flow (nghĩa là giá trị ANY):** nó khớp với tất cả các giá trị có thể có trong trường tiêu đề hoặc trường pipeline của gói tin
    - **Match fields có mặt và không bao gồm mặt nạ:** khớp với trường tiêu đề hoặc trường pipeline tương ứng từ gói có cùng giá trị.
    - **Switch hỗ trợ bitmask tùy ý trên các trường so khớp cụ thể:** mặt nạ này có thể chỉ rõ chính xác hơn các so khớp, trường match fields được khớp nếu nó có cùng giá trị cho các bit được đặt trong mặt nạ.

> 💡 **Quá trình:** So khớp với các mục flow → Chỉ mục có priority cao nhất được chọn → Cập nhật các bộ đếm liên quan → Thực hiện tập các instruction trong mục flow.

2.3.4 Table-miss

- Mọi bảng flow phải hỗ trợ một mục table-miss
- **Mục table-miss được xác định bởi các trường match fields và priority:** sử dụng wildcard cho tất cả các trường match fields (tất cả các trường bị bỏ qua) và có mức ưu tiên thấp nhất (0).
- Table-miss không tồn tại, gói tin không khớp sẽ bị bỏ qua.



Mục table-miss trong bảng thực hiện: không tồn tại mặc định trong một bảng flow, bộ controller có thể thêm/xóa bất cứ lúc nào, có thể hết hạn/bị trục xuất/được đọc bởi bộ controller.

2.3.5 Instruction

- **Mỗi mục flow chứa một tập hợp các instruction:** thay đổi gói tin, thiết lập action và/hoặc xử lý pipeline
- Switch không bắt buộc phải hỗ trợ tất cả các loại instruction, trừ loại được đánh dấu là "instruction bắt buộc":
  - **Apply-Actions action** (instruction không bắt buộc): Áp dụng các action cụ thể ngay lập tức mà không có bất kỳ thay đổi nào đối với bộ action. ⇒ chỉnh sửa gói giữa hai bảng hoặc để thực hiện nhiều action cùng loại.
  - **Clear-Action (instruction bắt buộc):** Xóa tất cả các action trong bộ action ngay lập tức
  - **Write-Actions action (instruction bắt buộc):** Hợp nhất (các) action cụ thể vào bộ action hiện tại. (Tồn tại: ghi đè/ chưa tồn tại: thêm)
  - **Write-Metadata metadata / mask (instruction không bắt buộc):** Viết giá trị siêu dữ liệu được che dấu vào trường metadata.
  - **Stat-Trigger stat thresholds (instruction không bắt buộc):** Tạo một sự kiện cho bộ controller nếu một số thống kê flow vượt qua một trong các giá trị ngưỡng stat.
  - **Goto-Table next-table-id (instruction bắt buộc):** Cho biết bảng kế tiếp trong quá trinh xử lý pipeline.
- Tập instruction liên kết với mục flow chứa tối đa một instruction cho từng loại.

    - **Ràng buộc:** *Clear-Actions* được thực thi trước lệnh *Write-Actions*, các *action-Apply* được thực hiện trước khi *Write-Metadata* và *Goto-Table* được thực hiện lần cuối.
- **Trả lại thông báo lỗi:** switch từ chối một mục flow do không thể thực hiện các instruction hoặc một phần của các instruction liên quan.

2.3.6 Bộ Action

- Kết hợp với mỗi gói
- Mặc định trống, chuyển giữa các bảng flow
- **Mục flow:** chỉnh sửa bộ action (lệnh Write-Action/Clear-Action) khi so khớp
    - Instruction: không chứa lệnh Goto-Table, quá trình xử lý pipeline dừng lại và các action trong bộ action của gói được thực hiện
- Bộ action chứa tối đa một action của mỗi loại.
    - Action *set-field*: xác định bởi trường đích ⇒ mỗi bộ chứa tối đa một action *set-field.* (ghi đè nếu cùng loại tồn tại)
    - Cần nhiều action cùng loại: sử dụng lệnh Apply-Actions.
- Thứ tự thực hiện bộ action (bất kể thứ tự được thêm vào tập hợp):
    1. **copy TTL inwards:** áp dụng các action sao chép TTL nội bộ vào gói tin
        a. **TTL (Time-to-Live)**: trường trong header của gói tin mạng (thường là IP) và nó được sử dụng để đếm số lượng bước chuyển tiếp (hops) mà gói tin đã thực hiện trong mạng. Mỗi khi gói tin đi qua một bước chuyển tiếp, giá trị TTL sẽ giảm đi 1. Khi giá trị TTL đạt 0, gói tin sẽ bị loại bỏ khỏi mạng.
    2. **pop:** áp dụng tất cả các hoạt động của thẻ vào gói tin
    3. **push-MPLS:** áp dụng thao tác đẩy thẻ MPLS đến gói tin
        a. **MPLS (Multiprotocol Label Switching)**: công nghệ định tuyến và chuyển tiếp dữ liệu trong mạng. Nó sử dụng thẻ (label) để định tuyến và chuyển tiếp dữ liệu thay vì sử dụng địa chỉ IP truyền thống.
    4. **push-PBB:** áp dụng thao tác push thẻ PBB vào gói tin

a. **PBB (Provider Backbone Bridges)**: tiêu chuẩn kỹ thuật cho việc định tuyến và chuyển tiếp gói tin trong các mạng truyền thông.

5. **push-VLAN:** áp dụng các thao tác đẩy thẻ VLAN vào gói tin
6. **copy TTL outwards:** áp dụng bản sao TTL ra bên ngoài action vào gói tin
7. **decrement TTL:** áp dụng giảm TTL đến gói tin
8. **set:** áp dụng tất cả các action thiết lập trường vào gói tin
9. **qos:** áp dụng tất cả các action QoS, chẳng hạn như meter (đo lường lưu lượng) và set-queue (thiết lập hàng đợi)cho gói tin
    a. **QoS (Quality of Service)**: tập hợp các kỹ thuật và chính sách được sử dụng để quản lý và ưu tiên lưu lượng trong mạng để đảm bảo chất lượng dịch vụ tốt.
10. **group:** nếu một action nhóm được chỉ định, áp dụng các action của (các) nhóm nhóm có liên quan theo thứ tự được chỉ định bởi danh sách này
11. **output:** nếu không có action group, chuyển tiếp gói tin trên cổng được xác định bởi action output

> 💡 **Chú ý:**
>
> - Action output trong tập action được thực hiện cuối cùng.
> - Nếu cả action output và action group được chỉ định trong tập hợp action, action output sẽ bị bỏ qua và action group được ưu tiên.
> - Gói tin bị bỏ qua:
>   - Không có action group & action output
>   - Không có action group & action output tham chiếu đến một cổng không tồn tại

- Xử lý action output về ingress & exgress:
    - action ingress chứa một action output/ action group chuyển tiếp gói tin tới một cổng: bắt đầu xử lý exgress trên cổng đó.

- action output tham chiếu đến cổng reserved ALL, bản sao (clone) của gói tin bắt đầu xử lý egress
- exgress chứa action output, gói tin rời khỏi quá trình egress và được xử lý bởi cổng (hầu hết chuyển tiếp ra khỏi switch)

2.3.7 Danh sách các action

- Instruction Apply-Actions và thông báo Packet-out bao gồm một danh sách các action.
- Thực hiện theo thứ tự
- **Hiệu quả:** tích lũy, nếu có chứa hai action Push VLAN, hai tiêu đề VLAN được thêm vào gói tin.
- **Chứa action output:** một bản sao (clone) của gói tin được chuyển tiếp trong trạng thái hiện tại đến cổng mong muốn (xử lý egress).
- Tham chiếu đến cổng reserved ALL, bản sao của gói bắt đầu xử lý egress cho mỗi cổng có liên quan.
- Action output tham chiếu đến 1 cổng không tồn tại ⇒ bản sao bị bỏ qua.
- Danh sách các action có chứa một action group: một bản sao của gói trong trạng thái hiện tại được xử lý bởi các group có liên quan.
- Thực hiện danh sách các action → tiếp tục thực hiện pipeline trên các gói đã chỉnh sửa

2.3.8 Action

Switch không bắt buộc để hỗ trợ tất cả các loại action, trừ "action bắt buộc”:

- **Output port_no (action bắt buộc):** Xuất chuyển gói tin tới một cổng OpenFlow đã chỉ định, nơi nó bắt đầu xử lý egress.
- **Group group_id (action bắt buộc):** Xử lý gói thông qua các group được chỉ định.
- **Drop (action bắt buộc):** Không có action đại diện rõ ràng. Gói tin bỏ qua khi tập action không có action đầu ra và không có action nhóm (do tập lệnh rỗng/ giỏ action rỗng khi xử lý pipeline/lệnh Clear-Actions).

- **Set-Queue queue_id (action không bắt buộc):** Action set-queue đặt id hàng đợi cho một gói tin.
- **Meter meter_id (action không bắt buộc):** Chuyển trực tiếp gói tới meter được chỉ định.
- **Push-Tag/Pop-Tag ethertype (action không bắt buộc):** Các switch có thể hỗ trợ khả năng push / pop tags (dạng thẻ ngoài cùng bên phải, xếp theo thứ tự MPLS → PBB → VLAN)
- **Set-Field field_type value (action không bắt buộc):** Các action Set-Field khác nhau được xác định theo loại trường và chỉnh sửa các giá trị của các trường tiêu đề tương ứng trong gói tin.
- **Change-TTL ttl (action không bắt buộc):** Các action Change-TTL khác nhau làm thay đổi giá trị của IPv4 TTL, IPv6 Hop Limit hoặc MPLS TTL trong gói tin.
- **Copy-Field src_field_type dst_field_type (action không bắt buộc):** Tác vụ Copy-Field có thể sao chép dữ liệu giữa bất kỳ trường tiêu đề gói tin hoặc pipeline nào.

2.3.9 Bảng group

- Group table: gồm các mục group.
- Mỗi mục của group được xác định bởi group identifter của nó và chứa:

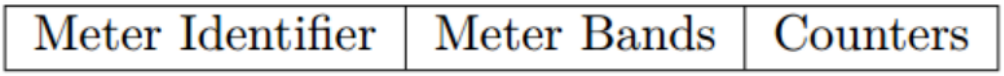
Group Identifier	Group Type	Counters	Action Buckets

  - **group identifter:** một số nguyên 32 bit không dấu duy nhất xác định group trên switch OpenFlow.
  - **group type:** để xác định loại nhóm.
  - **counters:** bộ đếm cập nhật khi các gói được xử lý bởi một nhóm.
  - **action buckets:** danh sách các giỏ action theo thứ tự, trong đó mỗi giỏ action chứa một tập hợp các action để thực hiện và các tham số liên quan. Các action trong một cái giỏ luôn được áp dụng như một bộ action

> 💡 Mục group có thể không có hoặc có nhiều giỏ action, ngoại trừ group indirect, là loại luôn luôn có một giỏ action.

2.3.10 Bảng meter

- Mỗi luồng flow cho phép OpenFlow thực hiện việc hạn chế tốc độ, một thao tác QoS đơn giản sẽ hạn chế một tập hợp các luồng đến một băng thông đã chọn
- Các meter hoàn toàn độc lập với các hàng đợi per-port
  - Có thể kết hợp để thực hiện các công việc phức tạp để bảo tồn các khuôn khổ QoS, chẳng hạn như DiffServ.
  - **DiffServ (Differentiated Services):** đánh dấu và quản lý lưu lượng mạng để đảm bảo các dịch vụ khác nhau trên mạng nhận được chất lượng phục vụ phù hợp.
    - **Hoạt động:** gắn thẻ (marking) các gói tin mạng với giá trị DSCP (Differentiated Services Code Point) trong header của chúng. DSCP là một trường 6 bit trong header IP được sử dụng để chỉ định cách xử lý gói tin trong mạng. Giá trị DSCP được sử dụng để phân loại các gói tin thành các lớp khác nhau dựa trên mức độ ưu tiên và yêu cầu về chất lượng dịch vụ.
- Một meter đo tốc độ các gói được gán cho nó và cho phép controller điều khiển tốc độ của các gói đó
- **Meter được gắn trực tiếp vào các mục flow:** meter đo và kiểm soát tỷ lệ tổng của tất cả các mục Flow mà nó được gắn vào.
- Các thành phần chính của một mục meter:



  - **meter identifter:** một số nguyên 32 bit không dấu xác định duy nhất một meter
  - **meter bands:** một danh sách các dải meter không có thứ tự, trong đó mỗi băng đo được xác định tỷ lệ băng tần và cách xử lý gói tin
  - **counters:** cập nhật khi các gói tin được xử lý bởi một meter

- Các flow khác nhau trong cùng một bảng Flow có thể sử dụng cùng một meter
- Mỗi bảng flow có thể nhập trực tiếp nó vào một meter

3. Tổng quan về OpenFlow Switch Protocol

- Kênh OpenFlow là kênh kết nối mỗi Switch OpenFlow với một bộ controller OpenFlow.
- Bộ controller cấu hình và quản lý switch, nhận các sự kiện từ switch, gửi các gói tin ra khỏi switch.
- Kênh OpenFlow thường được mã hóa bằng TLS, nhưng có thể chạy trực tiếp qua TCP
  - **TLS (Transport Layer Security)**:
- Giao thức OpenFlow Switch Protocol hỗ trợ ba loại bản tin:
  - **controller-to-switch:** bộ controller trực tiếp quản lý hoặc kiểm tra trạng thái của switch.
    - Thông điệp không được đồng bộ sử dụng để cập nhật cho bộ controller về các sự kiện mạng và thay đổi trạng thái switch
    - Thông điệp đối xứng được gửi mà không có sự báo trước
  - asynchronous - không đồng bộ
  - symmetric - đối xứng

3.1 Các loại bản tin

- Bản tin controller to switch khởi tạo bởi bộ controller và có thể yêu cầu phản hồi từ switch:
  - **Features:** Bộ controller có thể yêu cầu định danh switch và các tính năng cơ bản của nó bằng cách gửi một yêu cầu features.
  - **Conftguration:** bộ controller có thể thiết lập và truy vấn các tham số cấu hình trong switch. Switch chỉ trả lời một truy vấn từ bộ controller.

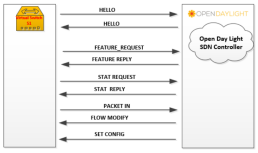
- **Modify-State:** được gửi bởi bộ controller để quản lý trạng thái trên các switch (bổ sung, xóa và sửa đổi các mục flow hoặc group, chèn hoặc xóa giỏ action của group trong các bảng OpenFlow và để thiết lập các thuộc tính cổng switch)
- **Packet-out:** được bộ controller sử dụng để gửi các gói tin ra khỏi một cổng quy định trên switch, và để chuyển tiếp các gói tin nhận được thông qua bản tin *Packetin*. Thông báo gói tin phải chứa một gói đầy đủ hoặc một bộ đệm ID tham chiếu đến một gói được lưu trữ trong switch. Bản tin cũng chứa một danh sách các action được áp dụng theo thứ tự chỉ định, một danh sách các action trống sẽ bỏ qua gói tin.
- **Barrier:** bản tin barrier request / reply được bộ controller sử dụng để đảm bảo gói tin đã được xử lý hoặc nhận thông báo cho các hoạt động đã hoàn thành.
- **Role-Request:** được bộ controller sử dụng để thiết lập vai trò của kênh OpenFlow của nó, thiết lập Controller ID, hoặc truy vấn các thông số này
- **Asynchronous-Conftguration:** bản tin được bộ controller sử dụng để thiết lập một bộ lọc bổ sung cho các bản tin không đồng bộ mà nó muốn nhận được trên kênh OpenFlow, hoặc để truy cập bộ filter

- **Bản tin không đồng bộ:** switch gửi các bản tin không đồng bộ đến các bộ controller để biểu thị một gói tin đến hoặc sự thay đổi trạng thái.
  - **Packet-in:** Chuyển việc xử lý gói tin đến bộ controller. Đối với tất cả các gói được chuyển tiếp tới cổng reserved CONROLROLER sử dụng mục flow hoặc mục tablemiss.
  - **Flow-Removed:** Thông báo cho bộ controller về việc loại bỏ một mục flow từ một bảng flow.
  - **Port-status:** Thông báo cho bộ controller sự thay đổi trạng thái cổng (cấu hình cổng và các sự kiện thay đổi trạng thái cổng)
  - **Role-status:** Thông báo cho bộ controller về sự thay đổi vai trò của nó. Khi một bộ controller mới tự chọn làm chủ, switch sẽ gửi thông báo trạng thái đến bộ controller cũ.
  - **Controller-Status:** Thông báo cho bộ controller khi tình trạng của một kênh OpenFlow thay đổi. Điều này có thể hỗ trợ quá trình thiết lập kênh dự phòng nếu bộ controller mất khả năng giao tiếp với nhau.

- **Flow-monitor:** Thông báo cho bộ controller sự thay đổi trong một bảng flow.

- Bản tin đối xứng được gửi đi theo cả hai hướng:
  - **Hello:** các bản tin được trao đổi giữa switch và controller khi khởi động kết nối.
  - **Echo:** có thể được gửi từ switch hoặc bộ controller, và phải trả về một phản hồi echo. Chúng chủ yếu được sử dụng để xác minh tính liên tục của bộ controller và switch, cũng có thể được sử dụng để đo độ trễ hoặc băng thông của nó.
  - **Error:** Các thông báo lỗi được sử dụng bởi switch hoặc bộ controller để thông báo các sự cố cho phía bên kia của kết nối. Chúng chủ yếu được sử dụng bởi switch để chỉ ra sự thất bại của một yêu cầu được tạo bởi bộ controller.
  - **Experimenter:** Thông điệp Experimenter cung cấp một cách chuẩn cho các switch OpenFlow cung cấp các chức năng bổ sung. Đây là một khu vực dàn cho các tính năng mới của các phiên bản OpenFlow trong tương lai.

3.2 Thiết lập kết nối



- Switch phải có khả năng giao tiếp với một bộ controller qua kết nối URI (cổng mặc định 6653).
  - Khởi tạo một kết nối TLS hoặc TCP tiêu chuẩn tới bộ controller theo các trường của kết nối URI.
- Các bước kết nối:
  - **Thiết lập kết nối lần đầu:** mỗi bên của kết nối phải gửi một thông điệp `OFPT_HELLO` với trường phiên bản OpenFlow cao nhất mà nó hỗ trợ
    - Nếu phiên bản thương lượng được hỗ trợ bởi người nhận, thì kết nối sẽ tiến hành. Nếu không, báo lỗi Hello Failed & kết thúc kết nối.
    - Trao đổi bản tin `OFPT_HELLO` và đàm phán thành công một phiên bản chung: thiết lập kết nối được thực hiện và thông điệp OpenFlow tiêu chuẩn có thể được trao đổi qua kết nối
      - Nên: controller gửi một thông báo `OFPT_FEATURES_REQUEST` để nhận được Datapath

ID của switch.

- **Nhận được thông báo:** tính toán phiên bản giao thức chuyển mạch OpenFlow.
  - Bản tin Hello nhận và gửi chứa hello `OFPHET_VERSIONBITMAP`:
    - Bitmap đó có một số bit thông thường: phiên bản thương lượng phải là phiên bản cao nhất được đặt trong cả hai bitmap.
    - Nếu không: phiên bản thương lượng phải là số phiên bản nhỏ hơn đã được gửi và số phiên bản đã nhận được trong các trường phiên bản

3.3 Bảo trì kết nối

- Thực hiện bởi các cơ chế kết nối TLS hoặc TCP ⇒ Đảm bảo hỗ trợ rộng rãi
- **Phát hiện gián đoạn kết nối & chấm dứt:** TCP timeouts và TLS session timeouts
- Mỗi kết nối phải được duy trì riêng biệt
  - Các kết nối phụ phải được chấm dứt khi kết nối chính tương ứng được chấm dứt hoặc bị hỏng, mặt khác nếu một kết nối phụ bị ngắt hoặc hỏng sẽ không làm ảnh hưởng đến kết nối chính hoặc các kết nối phụ.
- **Kết nối chấm dứt do điều kiện mạng hoặc thời gian chờ:** switch/controller cố gắng kết nối lại cho đến khi kết nối mới được thiết lập/ Connection URI của bên kia được gỡ bỏ khỏi cấu hình của nó.
- **Chấm dứt kết nối khi nhận được thông báo lỗi:** không có phiên bản Overflow chung/bên thứ ba không hỗ trợ tính năng quan trọng.

3.4 Ngắt kết nối

- Một switch bị mất liên lạc với bộ controller: gửi thông báo *controller status* đến tất cả các bộ controller kết nối còn lại (nếu có) cho biết vai trò và tình trạng kênh điều khiển của bộ controller bị ảnh hưởng.

    - Khi kênh OpenFlow được thiết lập lại, switch phải gửi thông báo cập nhật *controller status* đến tất cả bộ controller

- Nếu muốn, bộ controller có tùy chọn đọc tất cả các dòng chảy với một yêu cầu thống kê flow, để đồng bộ lại trạng thái với trạng thái switch.
    - Có thể tùy chọn xóa tất cả các mục dòng chảy với một yêu cầu mod-flow, để bắt đầu từ một trạng thái sạch trên switch.
-

RESEARCH AND NOTES

Tìm hiểu chung SDN:	https://www.geeksforgeeks.org/software-defined-networking/ https://www.techtarget.com/searchnetworking/definition/software-defined-networking-SDN https://www.cloudflare.com/en-gb/learning/network-layer/what-is-sdn/

FEEDBACKS

Gợi ý của thầy

- Tài liệu trên chủ yếu về Overflow, chưa đi chi tiết vào cấu trúc mạng SDN (tham khảo thêm các nguồn khác).

Công việc hoàn thiện

- [ ] Tìm hiểu SDN
- [ ] Đọc bài survey, tìm hiểu