ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SDN- Software Defined Network

Có sử dụng một số nội dung trong bài giảng: Software Defined Networking

Mohammad Alizadeh, MIT

https://people.csail.mit.edu/alizadeh/courses/6.888/slides/lecture14.pdf

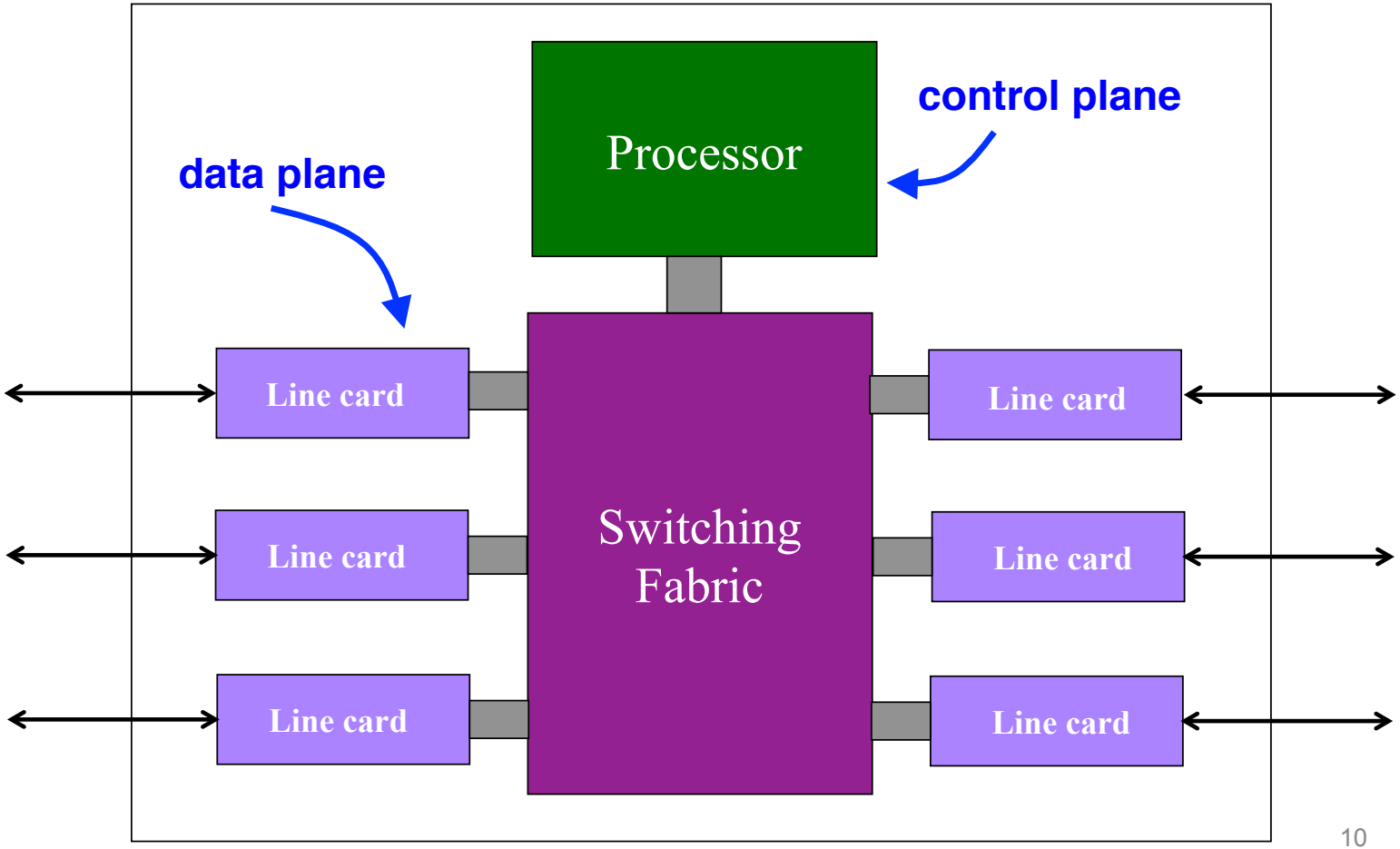
Và bài giảng của Raj Jain, Washington University in Saint Louis

http://www.cse.wustl.edu/~jain/cse570-13/

Data plane-Control plane-Management plan

- **Kiến trúc của router truyền thống bao gồm 3 mặt phẳng chức năng trên cùng một thiết bị:**
  - **Data plane**: xử lý và phân phối các gói tin căn cứ theo trạng thái chuyển tiếp cục bộ trên chuyển mạch (bảng chuyển tiếp)
    - Trạng thái chuyển tiếp + header gói tin → quyết định chuyển tiếp
    - Lọc, lưu trữ và lập lịch
  - **Control plane**: tính toán duy trì, cập nhật bảng chuyển tiếp cho chuyển mạch để data plan hoạt động được hiệu quả
    - Xác định cách chuyển tiếp các gói tin
    - Định tuyến, traffic engineering, phát hiện và phục hồi sự cố/lỗi ...
  - **Management plane**: cung cấp giao diện cho quản trị cấu hình và tinh chỉnh thiết bị/mạng
    - Traffic engineering, ACL config, cấu hình thiết bị, ...

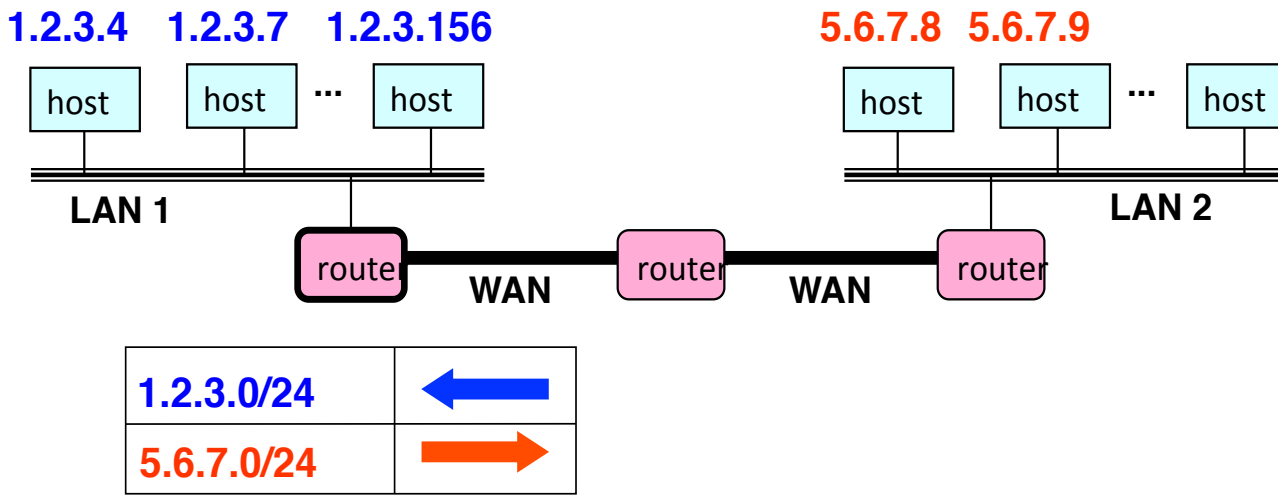
Data plan và Control plan trên router



Data plan

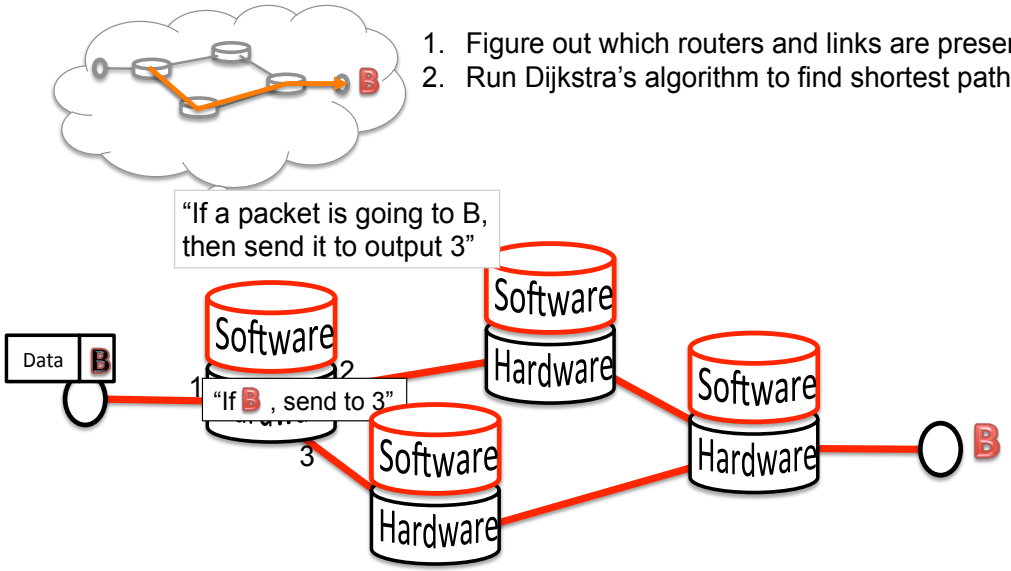
- Đối sánh tiêu đề gói tin với các luật quy định trong bảng chuyển tiếp
- Thực hiện chuyển tiếp gói tin.

Example: IP Forwarding



Control plan

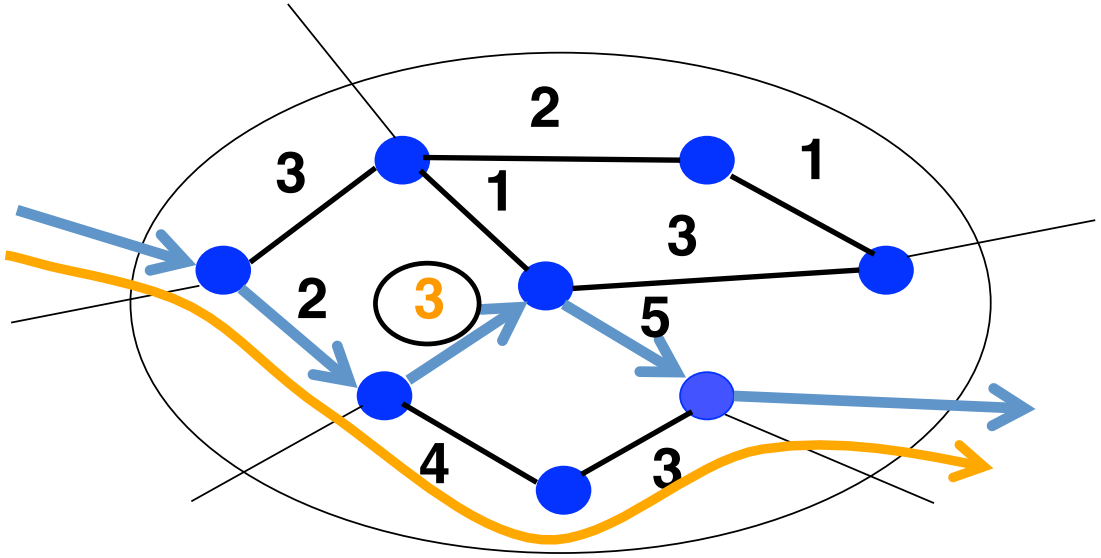
- Tính toán đường chuyển tiếp các gói tin
  - Phân phối thông tin đường đi này đến các bảng chuyển tiếp
  - Thông thường việc tính toán được thực hiện bằng các giao thức phân tán
- Ví dụ: giao thức Link-state routing (OSPF, IS-IS)
  - Quảng bá thông tin topo giữa mọi nút mạng
  - Mỗi nút tính đường đi ngắn nhất
  - Dùng giải thuật Dijkstra.



Management plan

Traffic Engineering: setting the weights

- Inversely proportional to link capacity?
- Proportional to propagation delay?
- Network-wide optimization based on traffic?

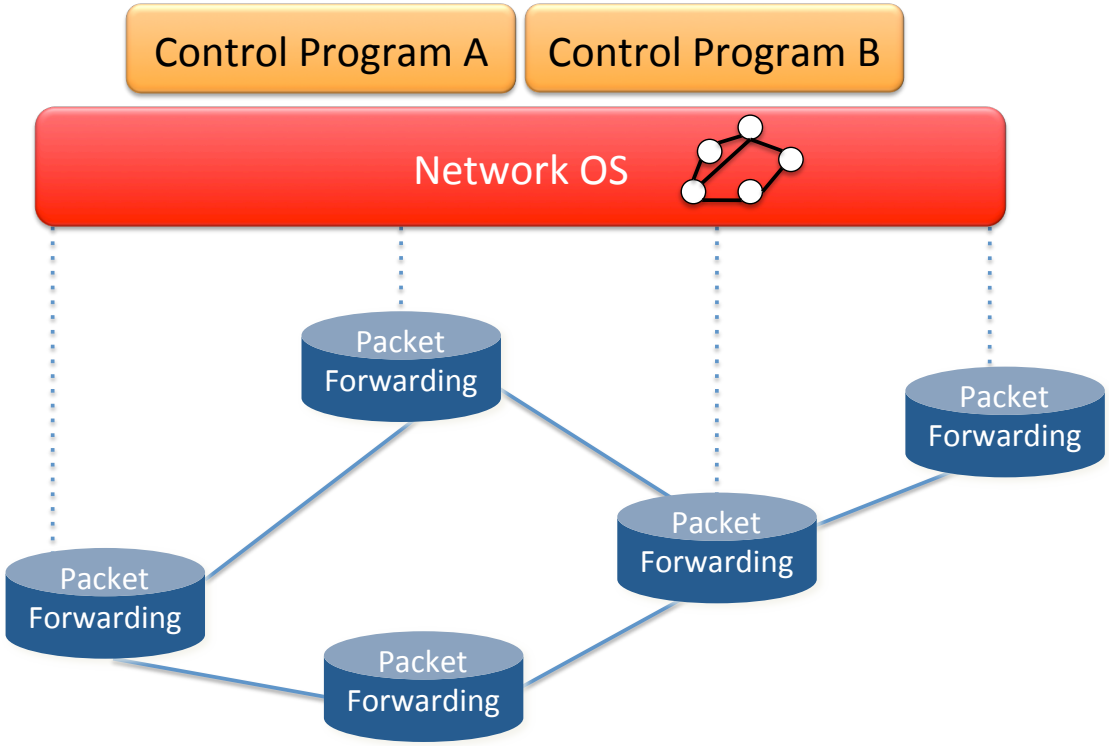


Tốc độ hoạt động của data, control, management plans

	Data	Control	Management
Time-scale	Packet (nsec)	Event (10 msec to sec)	Human (min to hours)
Location	Linecard hardware	Router software	Humans or scripts

Mạng SDN là gì

- Một mạng mà trong đó control plan được tách khỏi data plan về mặt vật lý, i.e. không cùng trên một thiết bị
  - Chức năng control plan của router truyền thống được chuyển đến một chương trình tập trung
  - Chương trình tập trung điều khiển một số thiết bị chuyển tiếp (đóng vai trò switch)



Lý do ra đời của SDN

- Sự phát triển của các thế hệ chuyển mạch và control plan:
  - Năm 90: Chức năng chuyển tiếp tầng 2, định tuyến được thực hiện bằng phần mềm
  - Năm 95: Chuyển tiếp tầng 2 được hardware hóa, định tuyến bằng phần mềm
  - Năm 2000: Chuyển tiếp tầng 2, chuyển tiếp tầng 3 được hardware hóa
- Việc hardware hóa các thành phần tầng thấp làm cho dữ liệu được chuyển tiếp nhanh hơn, phù hợp tốc độ đường truyền
- **➔ hardware cho phần forwarding**
- **➔ software cho phần control**

Lý do ra đời của SDN

- **Control plan phân tán không còn phù hợp với mạng hiện đại**
- Xu hướng thiết kế các thiết bị mạng trước đây
  - Mạng hoạt động phân tán
  - Nguyên tắc plug-and-play được đề cao ➔ Các nút xử lý và ra quyết định chuyển tiếp dữ liệu độc lập
  - Các nút phối hợp để xây dựng các bảng chuyển tiếp
    - Sử dụng Spanning Tree Protocol để xây dựng bảng chuyển tiếp MAC trên switch
    - Sử dụng các routing protocol để phối hợp xây dựng các bảng định tuyến của router.
    - ➔ Tốc độ hội tụ chậm (mức độ hàng chục giây, do quá trình trao đổi thông tin giữa các nút)
    - ➔ Tốc độ hội tụ này khó được chấp nhận trong mô hình mạng hiện đại, ví dụ Data center

Lý do ra đời của SDN

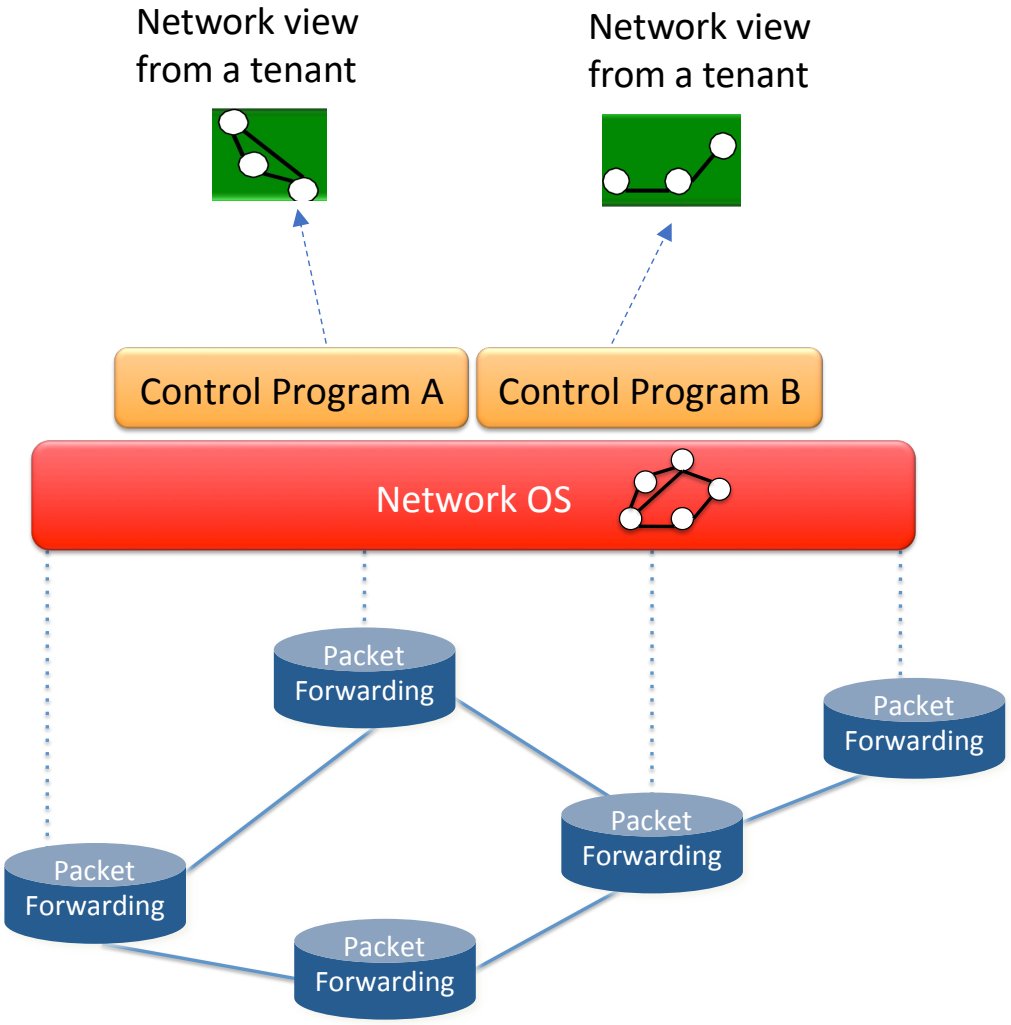
- **Nhu cầu đơn giản hóa các nút mạng**
- Chức năng của nút mạng theo mô hình xử lý độc lập ngày càng trở nên phức tạp
- Cần đơn giản hóa các nút mạng bằng cách tách control plan khỏi thiết bị dẫn đến
  - Dễ dàng tối ưu các tuyến đường nhờ thông tin toàn cục
  - Dễ dàng và nhanh chóng tự động hóa công việc quản lý thiết bị theo chính sách

Lý do ra đời của SDN

- Sự ra đời của các trung tâm dữ liệu thuê chung (multi-tenant data center)
  - Một data center với các thiết bị vật lý được thuê bởi nhiều khách hàng (tenant)
- Mỗi khách muốn nhìn/điều khiển phần mạng ảo (bao gồm cả server, storage, phần mềm) như mạng thật
  - Ảo hóa, xây dựng và quản lý mạng ảo được thực hiện dễ dàng hơn khi control plan được tập trung lại như trong mô hình SDN.
  - Cụ thể: các topology ảo của mỗi khách hàng được thiết lập bằng một tập các cấu hinh switch, tính toán và cài đặt bởi control plan tập trung.
  - Khách hang quản lý mạng ảo của mình qua ứng dụng dành riêng

SDN network

- Gồm 3 thành phần:
  - Control program (App): ứng dụng điều khiển sở hữu bởi mỗi tenant để quản lý phần mạng của mình
  - Network OS (Controller): Phần mềm tập trung điều khiển thiết bị chuyển tiếp dữ liệu
  - Thiết bị chuyển tiếp dữ liệu: SDN switch
    - Nằm phân tán.



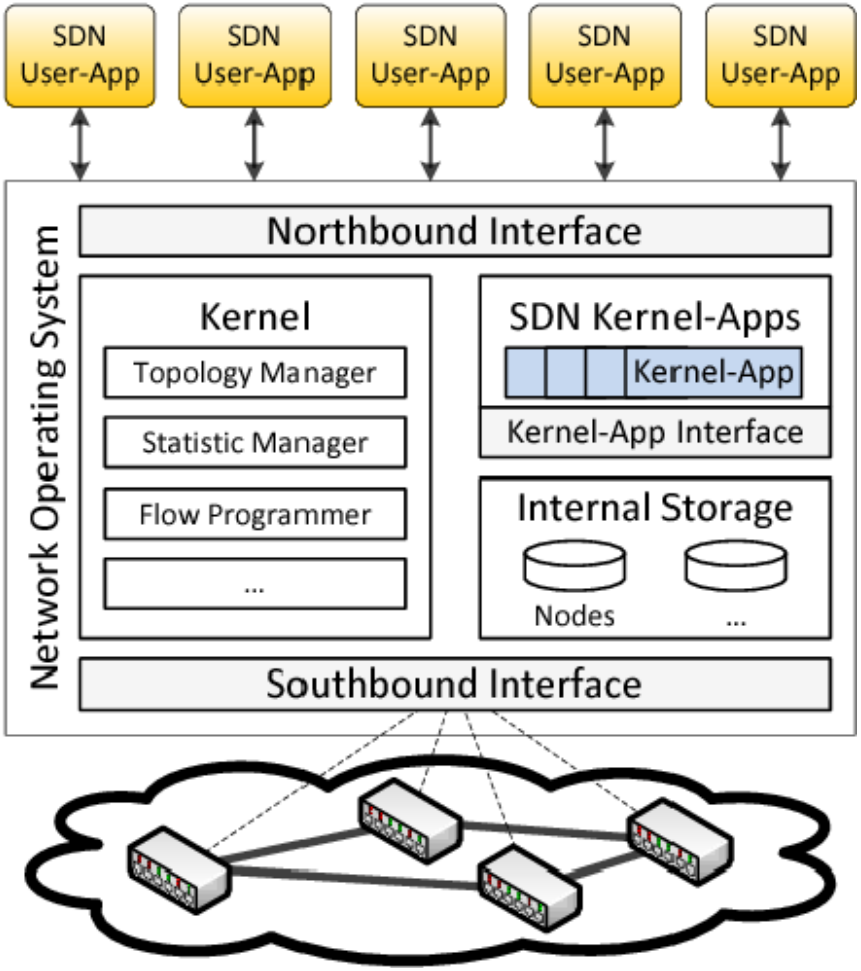
Control program (App)

- Control program (App):
  - Là ứng dụng điều khiển hoạt động của mạng theo cách mong muốn của người dùng (quản trị)
    - VD: Định nghĩa cơ chế chuyển tiếp dữ liệu: đường đi, phân luồng v.v…
- Control program điều khiển tập trung dựa trên khung nhìn toàn cục về mạng
  - **Input**: khung nhìn toàn cục (topo, csdl trạng thái)
  - **Output**: cấu hình cho mỗi thiết bị trong mạng

Network OS (Controller)

- **Network OS:**
  - Biểu diễn các thiết bị mà nó điều khiển (thiết bị chuyển mạch) theo một mô hình trừu tượng cho các App phía trên điều khiển.
    - Cung cấp giao diện cho các App lấy trạng thái của các switch
    - Ra lệnh cho các switch chuyển tiếp dữ liệu
  - Các app thông qua Network OS thiết lập các flow trên các switch.
  - Thường chạy trên một máy chủ
- Có nhiều phần mềm network OS có sẵn:
  - nNOX, ONIX, ONOS, Floodlight, Trema, OpenDaylight, HyperFlow, Kandoo, Beehive, Beacon, Maestro, ... + more

Network OS (Controller)

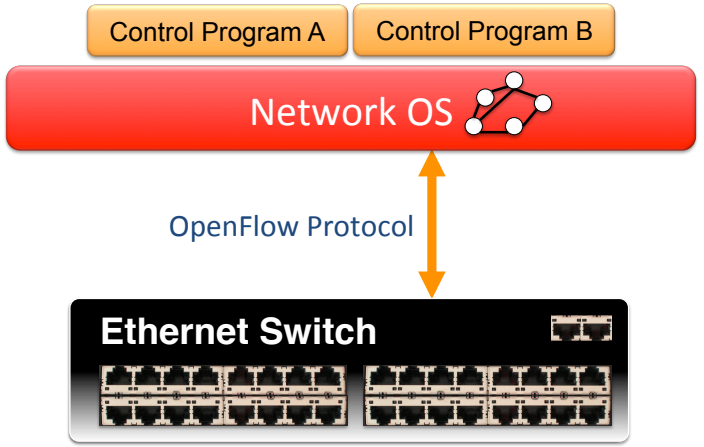


Các module cơ bản của Controller

- End-user Device Discovery:
  - Phát hiện các thiết bị của người dùng cuối: máy tính, máy in, điện thoại di động …
- Network Device Discovery
  - Phát hiện các thiết bị trong hạ tầng mạng bao gồm switch, router, access point …
- Network Device Topology Management
  - Duy trì thông tin về các kết nối giữa các thiết bị trong mạng và thiết bị đầu cuối.
- Flow Management
  - Quản lý một cơ sở dữ liệu các flow đang được áp dụng bởi controller và thực hiện các điều phối cần thiết với các switch để đảm bảo sự đồng bộ giữa các flow entry trên switch và các flow trong cơ sở dữ liệu

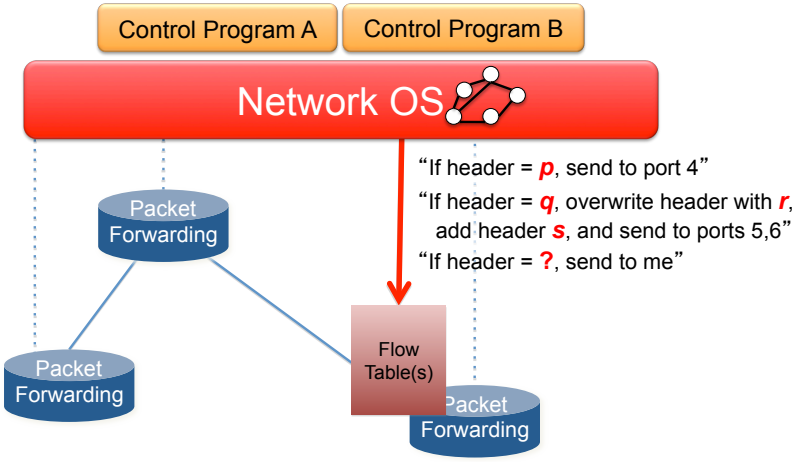
Các giao diện của Controller

- Southbound interface: Giao diện giữa Controller và các chuyển mach vật lý
  - Để Network OS (controller) có thể điều khiển các switch từ xa, cần có một giao thức cho giao diện southbound
  - Đã được chuẩn hóa: OpenFlow
- Northbound interface: giao diện giữa Controller và App
  - Không có chuẩn
  - Mỗi Controller cung cấp một giao diện riêng cho App:
    - Rest API
    - Python API
    - Java API



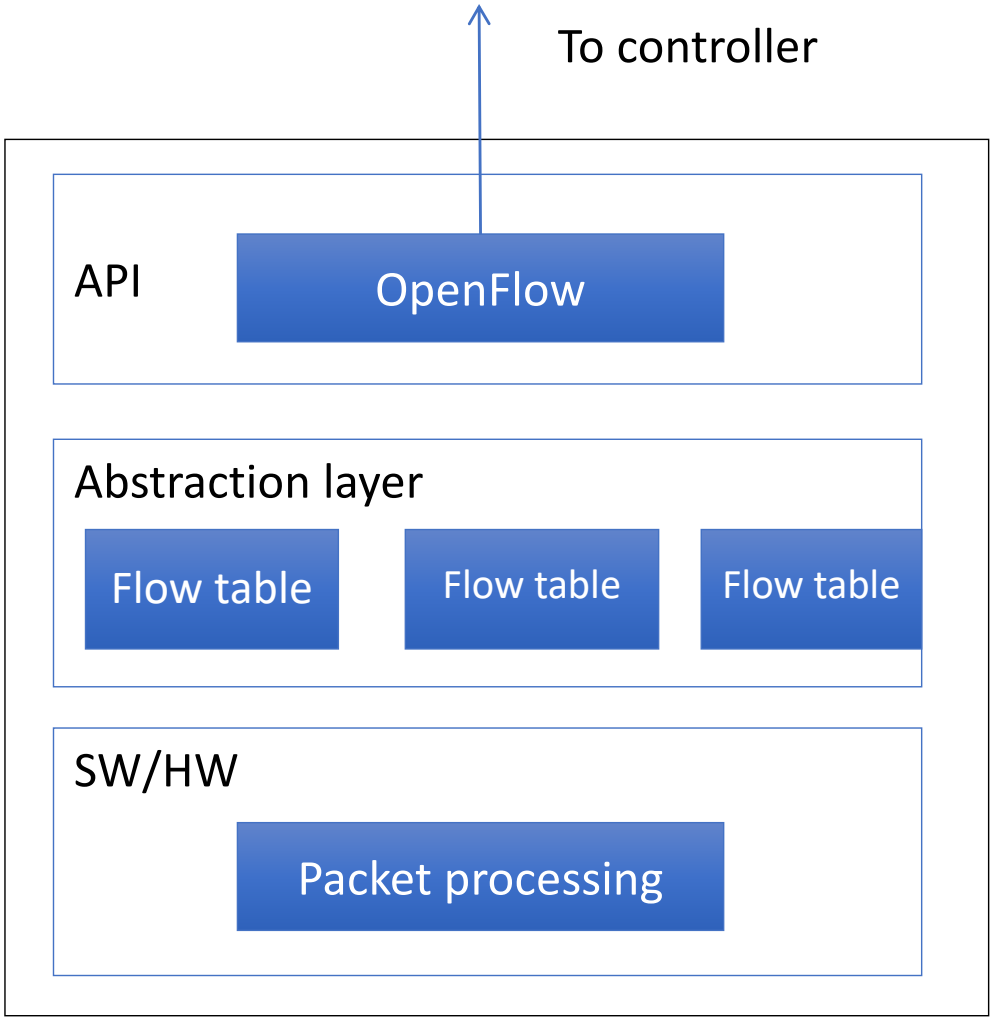
Southbound interface: Openflow

- Openflow: Giao thức giao tiếp giữa controller và switch
- OpenFlow cho phép Controller chỉ định đường đi các gói tin qua mạng các chuyển mạch
- OpenFlow cho phép controller quản trị các bảng chuyển tiếp của các switch ở xa bằng các lệnh thêm, sửa hoặc loại bỏ các luật xử lý gói tin.
  - Ví dụ một luật: Nếu header chứa "q" thì chuyển gói tin sang cổng 4
- Openflow sử dụng TCP để đóng gói các gói tin Openflow.
  - Trao đổi giữa switch và controller được vận chuyển bởi TCP → switch và controller không cần nối trực tiếp với nhau.
  - Controller thường dùng cổng TCP 6653 hoặc 6633 để chờ các yêu cầu kết nối từ switch.
  - Switch chủ động kết nối với controller để được quản lý và điều khiển.



Chuyển mạch SDN

- Flow table: định nghĩa cách xử lý đối với mỗi nhóm gói tin bằng luật:
  - Gói tin thỏa mãn điều kiện X thì được xử lý theo cách Y.
- Packet processing:
  - Đối chiếu mỗi gói tin đến switch lần lượt với các Flow table và thực hiện hành động được định nghĩa trong flow table tương ứng
  - Nếu không tìm được flow phù hợp gói tin được chuyển lên cho Controller xử lý tiếp.
- Switch giao tiếp với controller nhờ tầng API theo giao thức Openflow.



Flow table

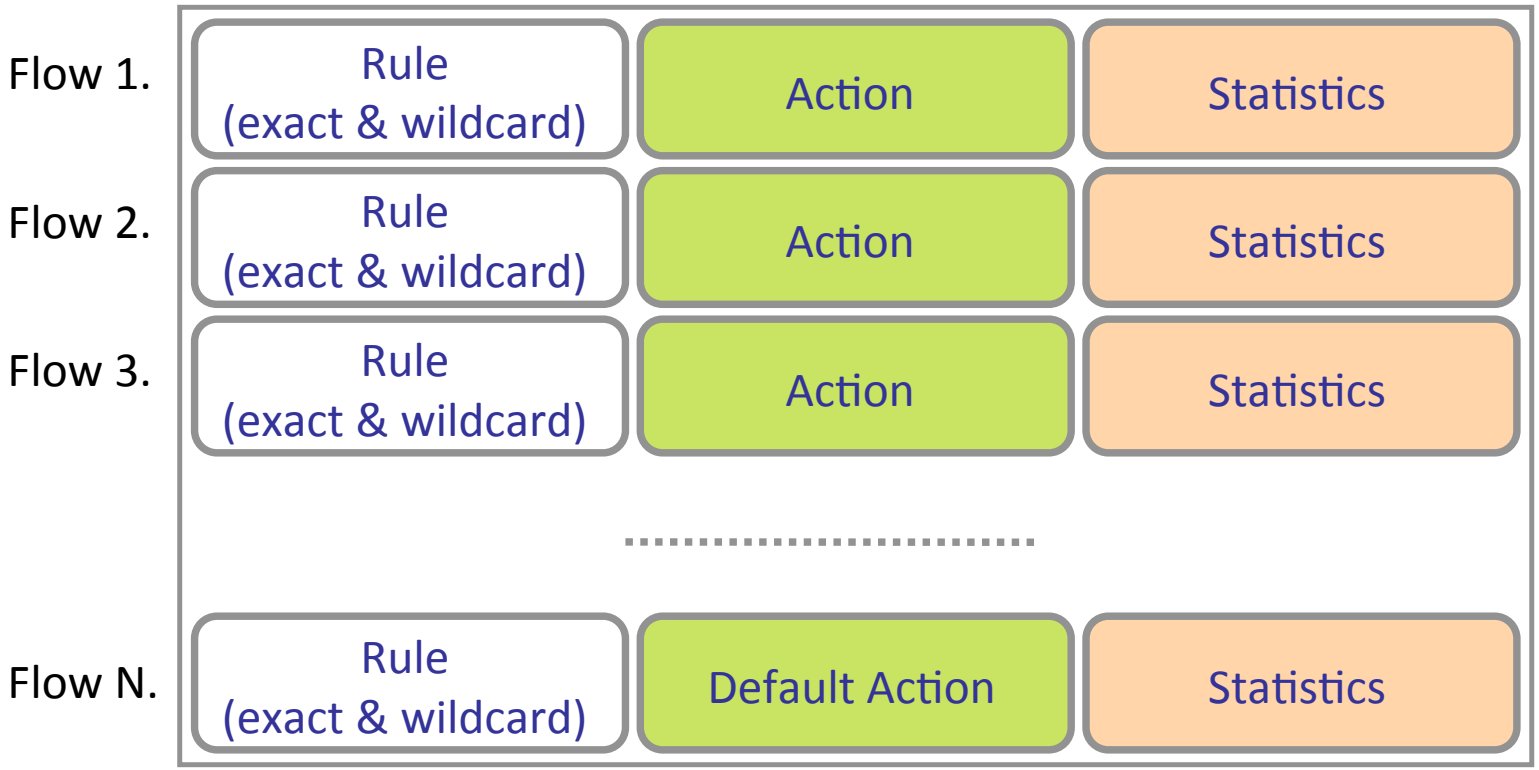
- Mỗi flow table gồm nhiều flow entry
- Mỗi flow entry định nghĩa một luật xử lý gồm <Match, Action>:
  - Match fields: định nghĩa các điều kiện (có thứ tự ưu tiên) mà header của gói tin cần có để gói tin được xử lý bằng hành động định nghĩa trong Action.
  - Action:
    - chuyển đến một cổng, hủy gói tin, gửi ra mọi cổng v.v…
    - Sửa header …

Match: 1000x01xx0101001x

- Dòng các gói tin phù hợp với 1 flow entry tạo thành một flow.

Các flow entry trong flow table

Exploit the flow table in switches, routers, and chipsets



Testbed

- Network emulation tools
  - Mininet: giả lập các chuyển mạch + các liên kết giữa các chuyển mạch → giả lập mạng vật lý.
    - Dùng script để tạo lập topo: thêm switch, định nghĩa kết nối giữa các switch.
    - http://mininet.org/
  - Openvswitch: giả lập từng chuyển mạch riêng
- SDN Controller (Network OS)
  - Opendaylight (6653):
    - https://docs.opendaylight.org/en/latest/getting-started-guide/index.html
  - Floodlight, POX (cổng 6633),
  - ONOS (cổng 6653)
    - https://wiki.onosproject.org/display/ONOS/Basic+ONOS+Tutorial
- Cấu hình các chuyển mạch SDN để kết nối đến controller.
  - Cung cấp cho các chuyển mạch IP của máy cài Controller và cổng của controller.
  - Mininet cung cấp cơ chế để đưa thông tin này cho các chuyển mạch.

Bài tập 1

- Cài đặt thử nghiệm mạng SDN
  - Sử dụng Mininet giả lập mạng các switch
  - Sử dụng 1 controller: Opendaylight, ONOS
  - Yêu cầu:
    - Nhìn được topo các switch trên giao diện đồ họa của Opendaylight và ONOS
    - Ping thử giữa các máy nối với các switch.

Tìm hiểu xa hơn

- Openflow được cài đặt trong MiniNet (mininet.org)
- Nguồn tài liệu thêm
  - Open Networking Foundation: https://www.opennetworking.org/
- Nhiều phần của slide được lấy từ

  https://www.clear.rice.edu/comp529/www/papers/tutorial_4.pdf

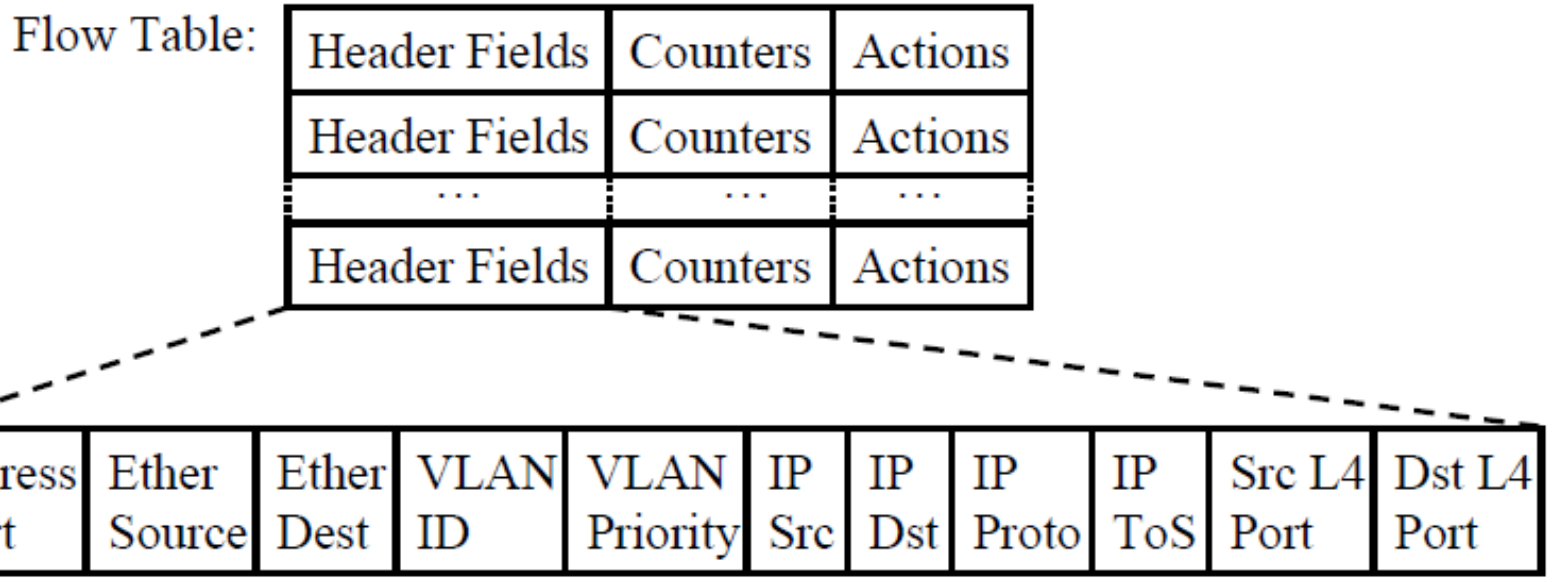
  Và sách “Software Defined Networks A Comprehensive Approach” Paul Goransson, Chuck Black

Openflow 1.0

- Ports, Port queues
- Flow table
- Packet matching
- Actions và chuyển tiếp gói tin
- Các bản tin trao đổi giữa controller và switch

Đặc tả Openflow 1.0

- Khi một packet đến switch, các trường header được đối chiếu với các flow entry trong flow table.
  - Nếu có một entry phù hợp, giá trị các bộ đếm trong mục counter được cập nhật và các hành động trong trường action được thực hiện.



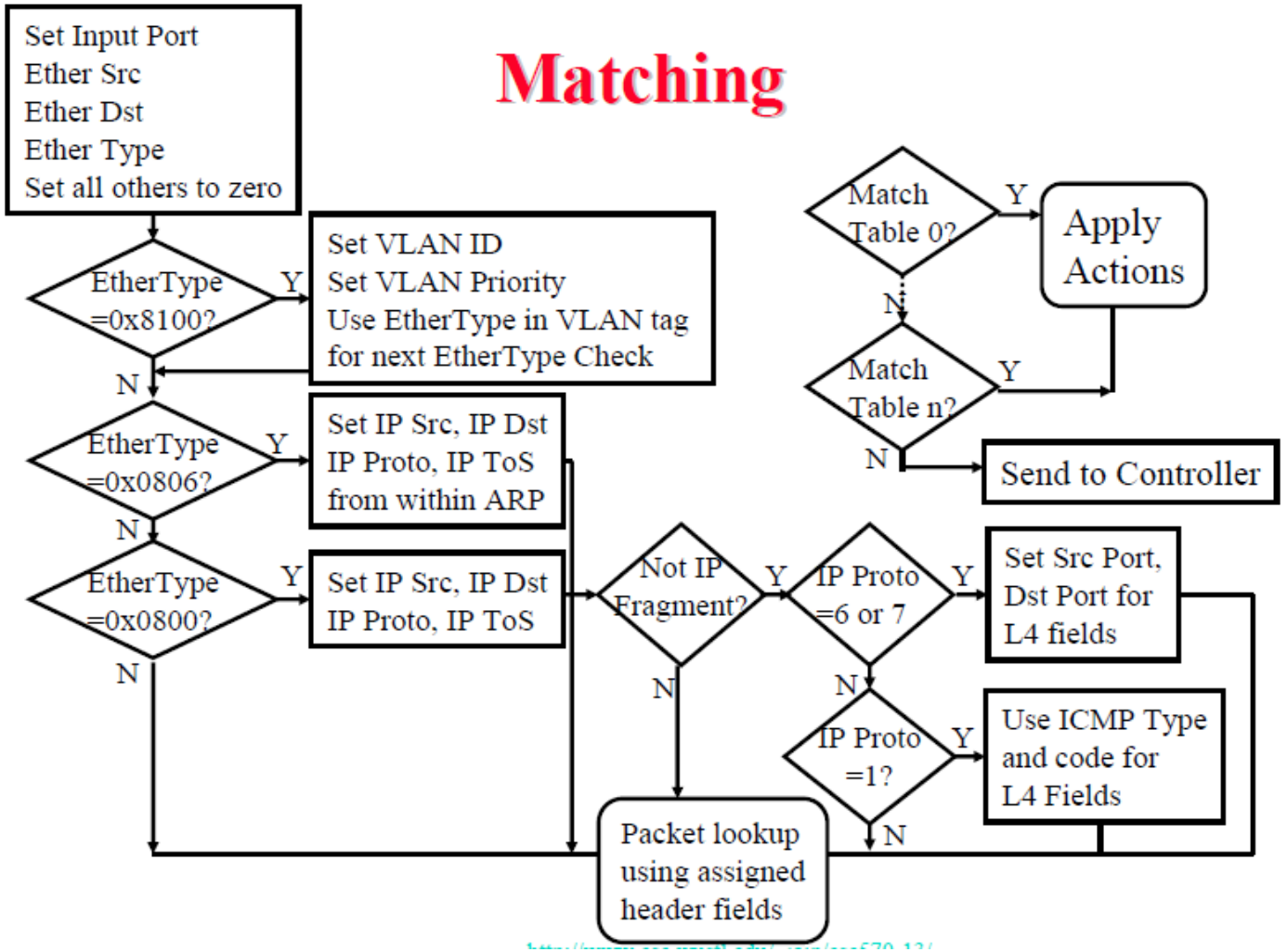
Ví dụ một flow table

Port	Src MAC	Dst MAC	VLAN ID	Priority	EtherType	Src IP	Dst IP	IP Proto	IP ToS	Src L4 Port ICMP Type	Dst L4 Port ICMP Code	Action	Counter
*	*	0A:C8:*	*	*	*	*	*	*	*	*	*	Port 1	102
*	*	*	*	*	*	*	192.168.*.*	*	*	*	*	Port 2	202
*	*	*	*	*	*	*	*	*	*	21	21	Drop	420
*	*	*	*	*	*	*	*	0x806	*	*	*	Local	444
*	*	*	*	*	*	*	*	0x1*	*	*	*	Controller	1

- Idle timeout: Remove entry if no packets received for this time
- Hard timeout: Remove entry after this time
- If both are set, the entry is removed if either one expires.

Ref: S. Azodolmolky, "Software Defined Networking with OpenFlow," Packt Publishing, October 2013, 152 pp., ISBN:978-1-84969-872-6 (Safari Book)

Quy tắc matching



Các bộ đếm

Per Table	Per Flow	Per Port	Per Queue
Active Entries	Received Packets	Received Packets	Transmit Packets
Packet Lookups	Received Bytes	Transmitted Packets	Transmit Bytes
Packet Matches	Duration (Secs)	Received Bytes	Transmit overrun errors
	Duration (nanosecs)	Transmitted Bytes	
		Receive Drops	
		Transmit Drops	
		Receive Errors	
		Transmit Errors	
		Receive Frame Alignment Errors	
		Receive Overrun erorrs	
		Receive CRC Errors	
		Collisions	

Các action

- Chuyển tiếp gói tin nhận được đến một cổng vật lý hoặc cổng ảo (virtual port). Các loại cổng ảo có thể:
  - All: tất cả các cổng ra trừ cổng gói tin đến
  - Controller: đóng gói và gửi gói tin đến controller
  - Local: áp dụng cho trường hợp gói tin Openflow nhận được từ controller (control plan) nhưng đến trên các cổng data plan, gửi gói đến bộ xử lý openflow của chính switch
  - Table: áp dụng cho các gói controller gửi cho switch có kèm theo danh sách action, thực hiện action trong danh sách

Các action

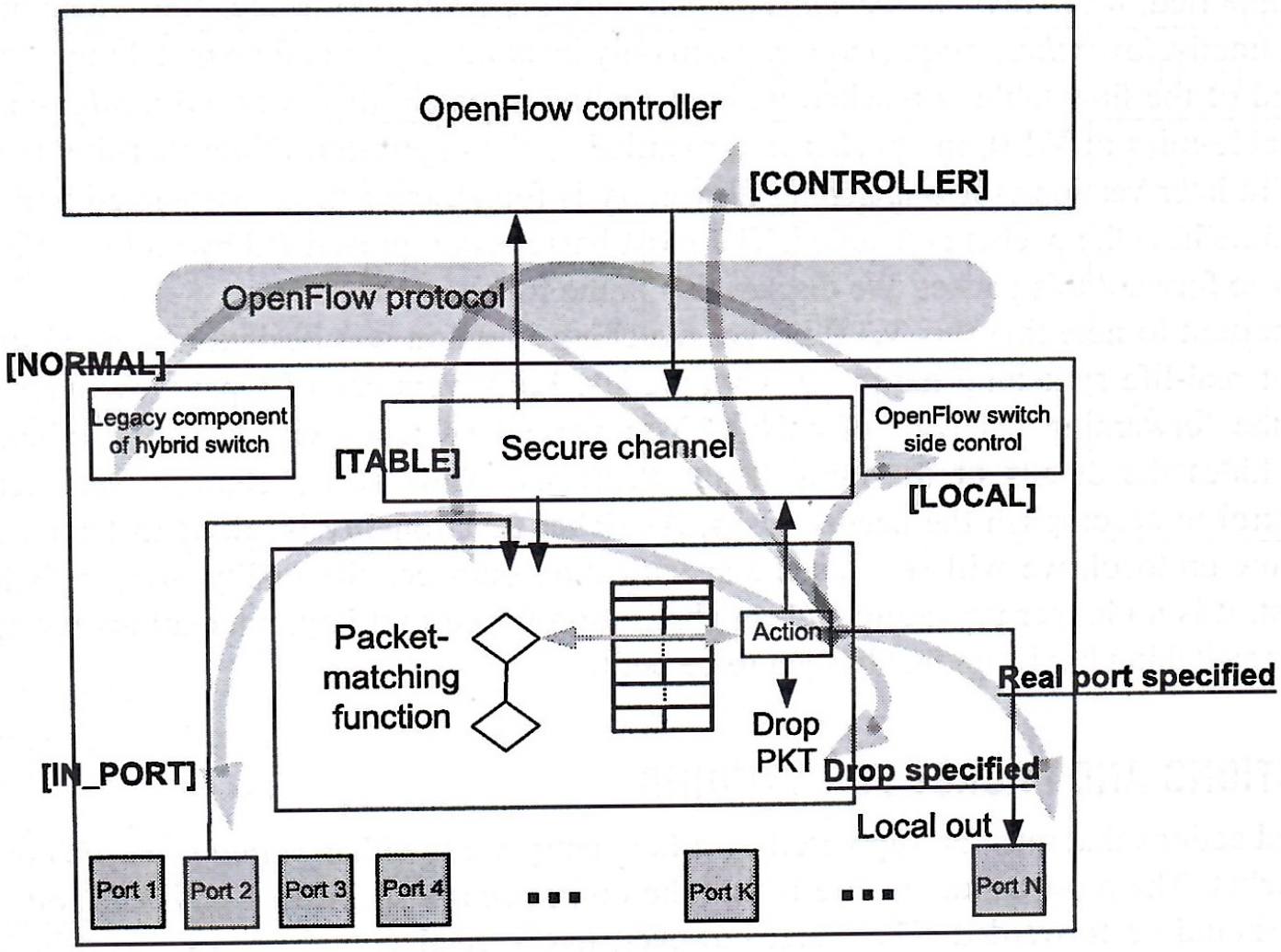
- In-port: Gửi ngược lại cổng vào
- Normal: Gửi gói tin ra cổng Ethernet theo cơ chế chuyển tiếp của switch thường (cơ chế tự học)
- Flood: Copy gói tin đến tất cả các cổng theo cây khung trừ cổng gói tin đến

- Xếp gói tin vào hàng đợi: xếp vào một hàng đợi của một cổng để được xử lý theo mức QoS tương ứng.
- Hủy gói
- Thay đổi trường: ví dụ thêm/bớt VLAN tags, đổi TTL, thay đổi bit ToS

Các action

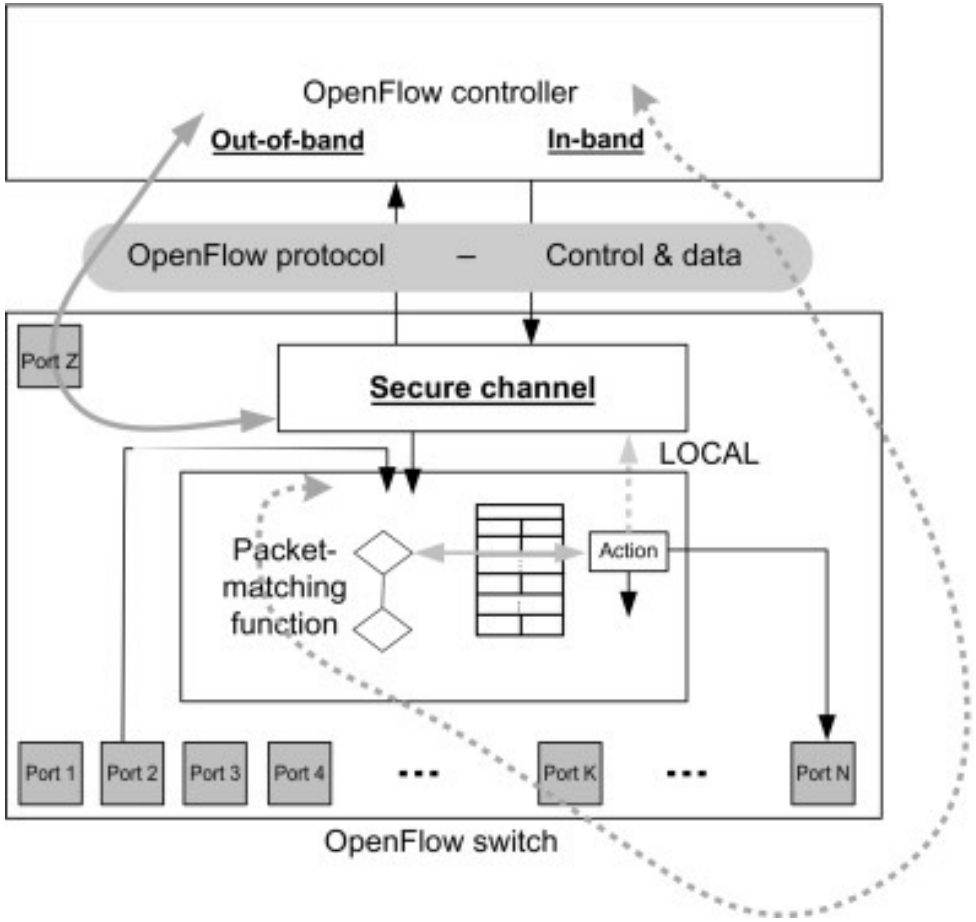
- Nếu header của gói tin không phù hợp với bất kỳ flow entry nào, gói tin được xếp vào hàng đợi và header được gửi cho controller, controller gửi lại switch một luật mới để xử lý các gói tin tương tự sau này.
- Secure chanel: kênh kết nối giữa controller và switch sử dụng TLS

Action: minh họa các luồng xử lý



Kênh liên lạc Controller-switch

- Gọi là secure chanel
- Thông thường được bảo mật bằng mã hóa không đối sứng dựa trên TLS
- Cũng có thể dùng kênh không mã hóa TCP
- Chế độ in-band
  - Dùng cổng dữ liệu nên gói tin điều khiển cũng được xử lý bằng openflow
- Chế độ out-of-band
  - Dùng cổng riêng, không xử lý chuyển mạch bằng openflow



Các bản tin giữa Controller và switch

- Controller-to-switch: các bản tin kiểm tra tình trạng switch
  - Features, config, thay đổi trạng thái, đọc trạng thái, packet-out, etc
- Loại bản tin Asynchronous: gửi từ switch đến controller mà không cần được yêu cầu
  - Packet-in, flow removed/expired, port status, error, etc
- Loại bản tin Symmetric: bản tin có thể gửi từ switch hoặc controller mà không cần đối phương yêu cầu.
  - Hello, Echo, etc.

Bản tin Openflow

Table 5.1 OFPT Message Types in OpenFlow 1.0

Message Type	Category	Subcategory
HELLO	Symmetric	Immutable
ECHO_REQUEST	Symmetric	Immutable
ECHO_REPLY	Symmetric	Immutable
VENDOR	Symmetric	Immutable
FEATURES_REQUEST	Controller-Switch	Switch Configuration
FEATURES_REPLY	Controller-Switch	Switch Configuration
GET_CONFIG_REQUEST	Controller-Switch	Switch Configuration
GET_CONFIG_REPLY	Controller-Switch	Switch Configuration
SET_CONFIG	Controller-Switch	Switch Configuration
PACKET_IN	Async	NA
FLOW_REMOVED	Async	NA
PORT_STATUS	Async	NA
ERROR	Async	NA
PACKET_OUT	Controller-Switch	Cmd from controller
FLOW_MOD	Controller-Switch	Cmd from controller
PORT_MOD	Controller-Switch	Cmd from controller
STATS_REQUEST	Controller-Switch	Statistics
STATS_REPLY	Controller-Switch	Statistics
BARRIER_REQUEST	Controller-Switch	Barrier
BARRIER_REPLY	Controller-Switch	Barrier
QUEUE_GET_CONFIG_REQUEST	Controller-Switch	Queue configuration
QUEUE_GET_CONFIG_REPLY	Controller-Switch	Queue configuration

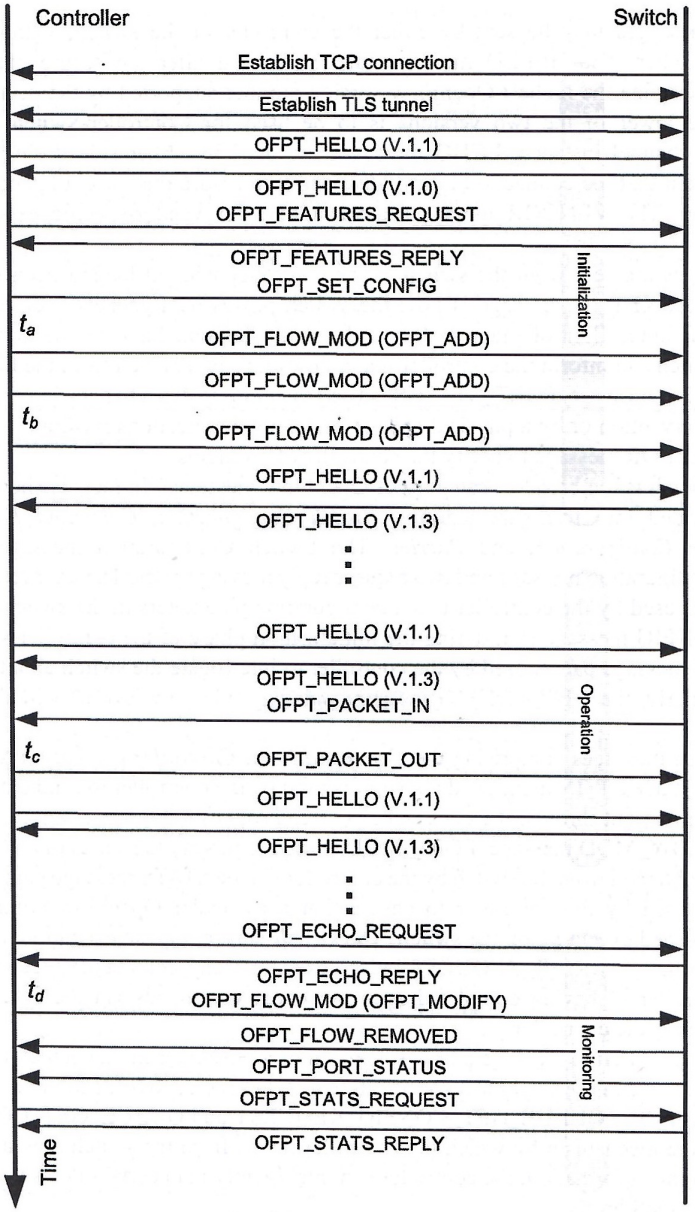
Các bản tin Openflow

- HELLO: trao đổi sau khi secure channel được thiết lập. Xác định phiên bản OF cao nhất được cả controller và switch hỗ trợ để sử dụng
- ECHO: Sử dụng bởi một trong hai bên kiểm tra nếu kênh vẫn còn tồn tại
- PACKET-IN: switch chuyển lại một gói tin cho controller xử lý ngoại lệ
- FLOW_REMOVE: switch báo cho controller biết 1 flow đã bị loại bỏ trên switch
- PORT_STATUS: switch thông báo cho controller trạng thái một cổng đã thay đổi.

Các bản tin Openflow

- Các bản tin cấu hình switch:
  - SET_CONFIG, GET_CONFIG, FEATURE..
- Các bản tin BARRIER
  - Đảm bảo switch thực hiện xong các lệnh đã nhận được trước bản tin BARRIER_REQUEST
- Các bản tin QUEUE
  - Dùng để controller đọc từ switch một queue đã được cấu hình như thế nào.

Bản tin Openflow



Hardware Open flow switch

- Arista 7050
- Brocade MLXe, Brocade CER, Brocade CES
- Extreme Summit x440, x460, x670
- Huawei openflow-capable router platforms
- HP 3500, 3500yl, 5400zl, 6200yl, 6600, and 8200zl (the oldstyle L3 hardware match platform)
- HP V2 line cards in the 5400zl and 8200zl (the newer L2 hardware match platform)
- IBM 8264
- Juniper (MX, EX)
- NEC IP8800, NEC PF5240, NEC PF5820
- NetGear 7328SO, NetGear 7352SO
- Pronto (3290, 3295, 3780) - runs the shipping pica8 software
- Switch Light platform

Các switch mềm hiểu OpenFlow

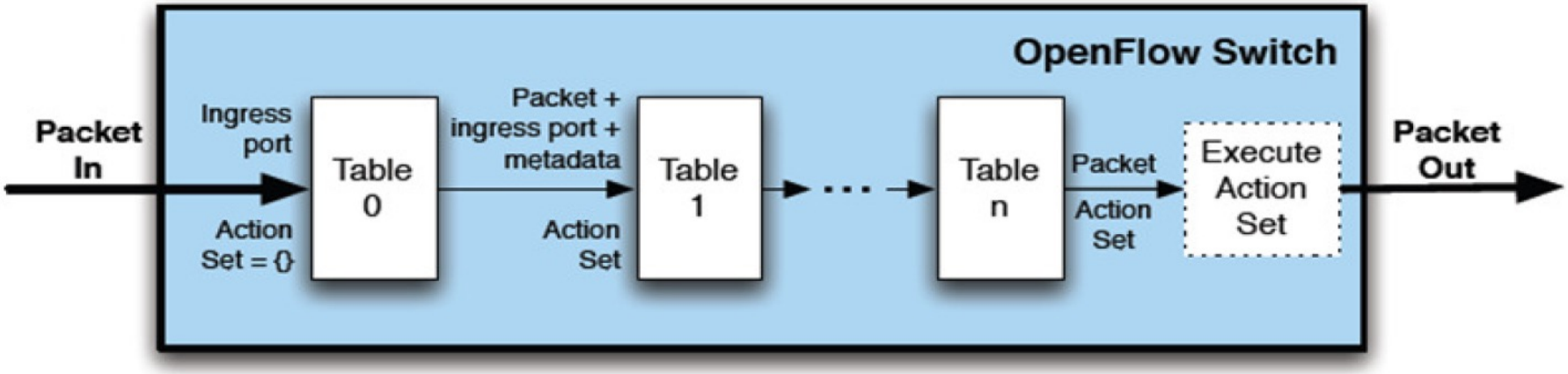
- **Open vSwitch**
- **Indigo**: Open source implementation that runs on physical switches and uses features of the ASICs to run OpenFlow
- **LINC:** Open source implementation that runs on Linux, Solaris,Windows, MacOS, and FreeBSD
- **Pantou:** Turns a commercial wireless router/access point to an OpenFlow enabled switch. OpenFlow runs on OpenWRT. Supports generic Broadcom and some models of LinkSys and TP-Link access points with Broadcom and Atheros chipsets.
- **Of13softswitch:** User-space software switch based on Ericsson TrafficLab 1.1 softswitch
- **XORPlus:** Open source switching software to drive high-performance ASICs. Supports STP/RSTP/MSTP, LCAP, QoS, VLAN, LLDP, ACL,OSPF/ECMP, RIP, IGMP, IPv6, PIM-SM

Openflow 1.1 concepts

- Nhiều flow tables
- Có Groups
- MPLS and VLAN tag support
- Virtual ports
- Controller connection failure

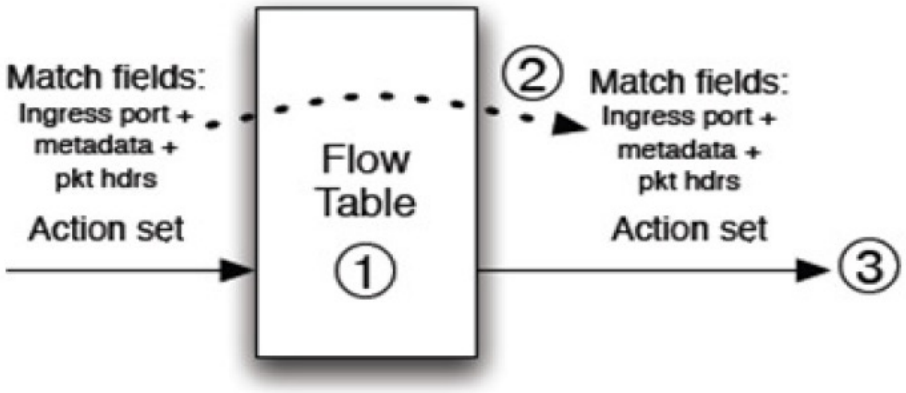
Pipeline processing (mới có từ Openflow 1.1)

- Một switch có thể có nhiều flow table và dữ liệu đến được đối chiếu với các table này theo kiểu pipeline.



(a) Packets are matched against multiple tables in the pipeline

Xử lý gói tin trên từng bảng



① Find highest-priority matching flow entry

② Apply instructions:
- i. Modify packet & update match fields (apply actions instruction)
- ii. Update action set (clear actions and/or write actions instructions)
- iii. Update metadata

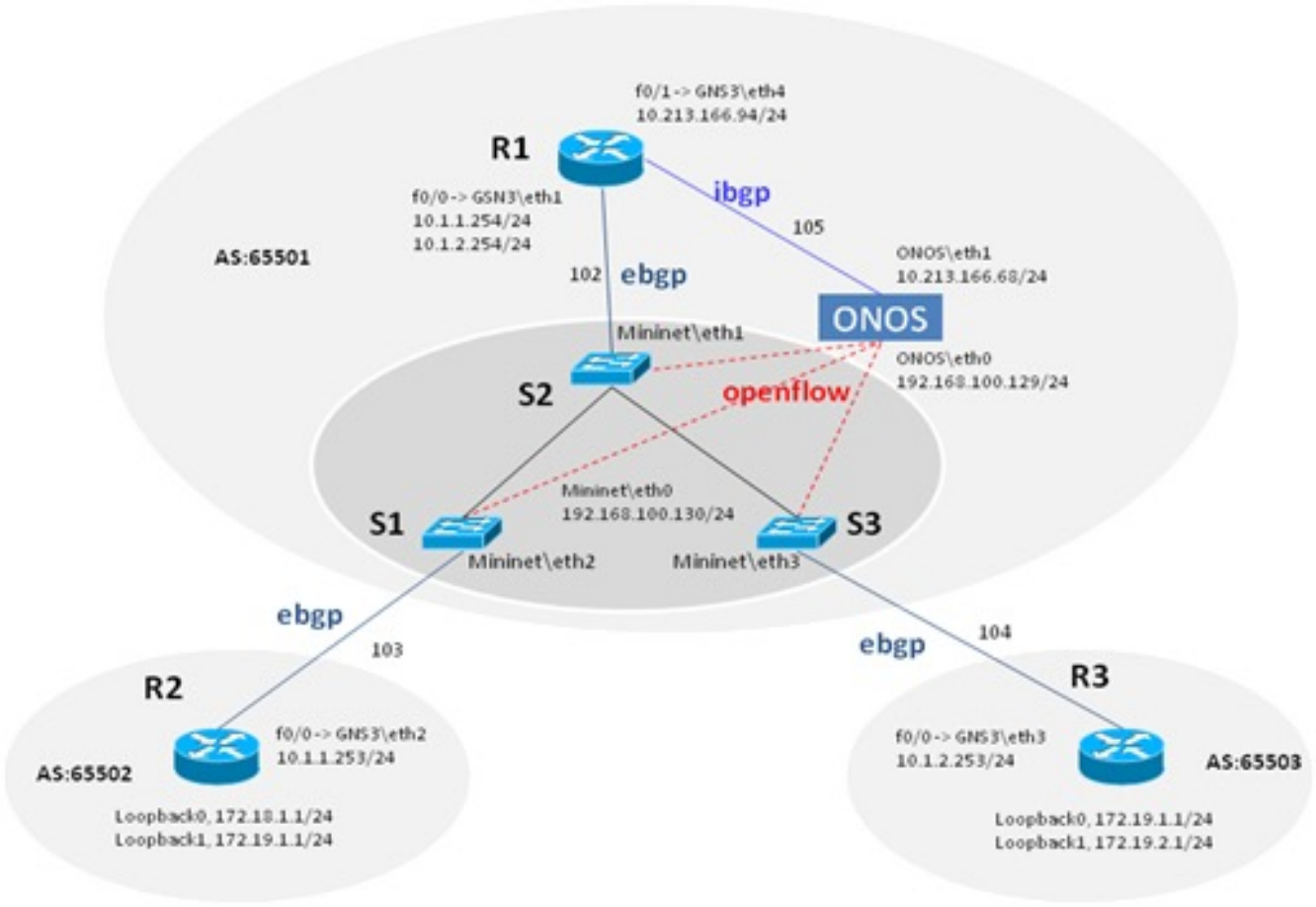
③ Send match data and action set to next table

(b) Per-table packet processing

Ghép nối SDN và IP

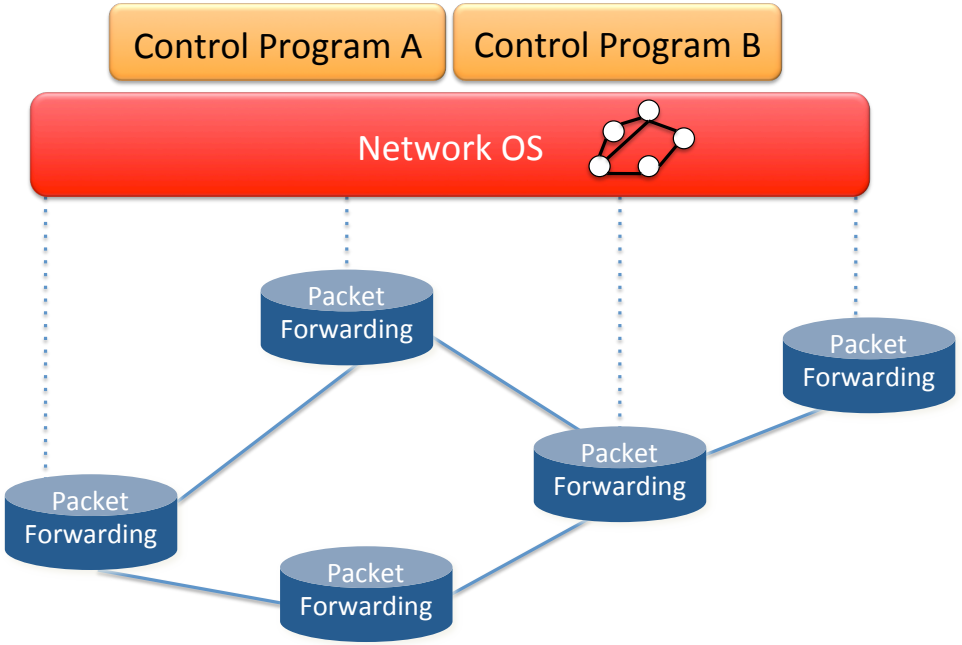
- Làm thế nào để vận chuyển dữ liệu giữa mạng IP và mạng SDN trong khi phương pháp định tuyến hai bên khác nhau.
- Coi phần mạng SDN và phần mạng IP là hai AS và sử dụng BGP để trao đổi thông tin network reachability hai bên
- SDN controller phải có module hiểu BGP (BGP plugin)
  - SDN controller đóng vai trò BGP router của miền SDN
- Trên control plan, SDN controller và các BGP router của các AS IP trao đổi thông tin định tuyến
- Trên data plan, SDN switch và BGP router của các AS IP chuyển tiếp gói tin cho nhau
- Chức năng đã được cài đặt trên Controller ONOS trong gói SDN-IP
- ONOS: wiki.onosproject.org

Ghép nối SDN và IP



Xây dựng SDN app

- Xây dựng topo mạng với switch cứng hoặc switch mềm sử dụng Mininet,
- Cài đặt controller
- Cấu hình các switch (mininet) làm việc với controller
- Xác định phương thức giao tiếp với Controller
- Xây dựng ứng dụng định tuyến theo mong muốn.



Bài tập 2

- Bài tập:
  - Xây dựng một topo thử nghiệm có ít nhất 10 switches trên Mininet
  - Xây dựng ứng dụng (SDN app) cho phép định tuyến dữ liệu giữa 2 điểm nguồn đích bất kỳ theo một đường đi có tiêu chí nhất định do nhóm quy định hoặc theo một tuyến đường do người dùng nhập vào
- Tài liệu:
  - Giao diện RestAPI của ONOS:
    - https://wiki.onosproject.org/display/ONOS/Appendix+B%3A+REST+API
  - Giao diện API của Opendaylight:
    - https://docs.opendaylight.org/en/stable-oxygen/developer-guide/controller.html#