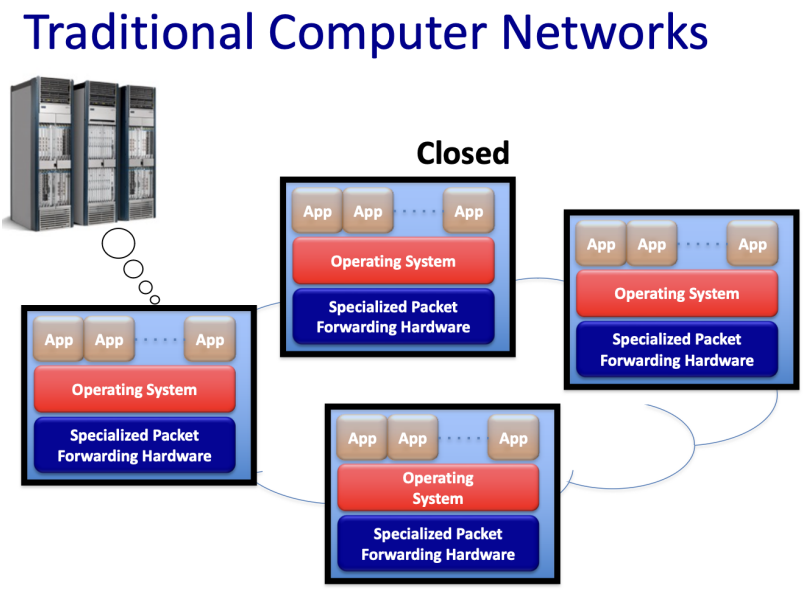
Lecture 5 Soft Defined Network Overview

Traditional Computer Networks

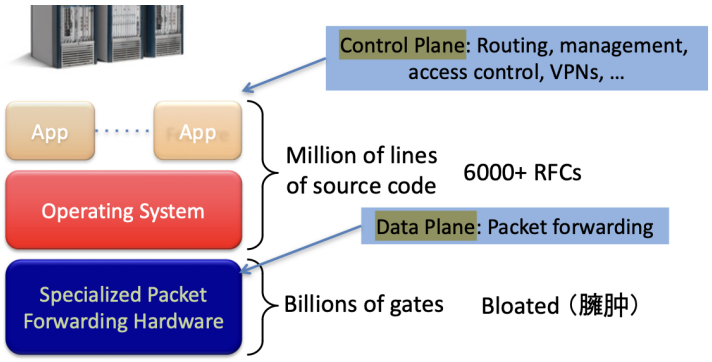
Closed & Distributed

- APP
- Operating System
  - specialized in Server, not just simple OS like Windows and Linux
- **Specialized** Packet Forwarding Hardware
  - encapsulation（封装）



The Ossified(僵化的) Network

- Control Plane
  - Routing / Management / Access Control / VPNs ...
- Plane between APP and OS
  - Million of lines of source code (6000+ RFCs)
- Data Plane
  - Packet forwarding
  - Billion of gates (Bloated, 臃肿的)



The fact is that

- Many complex functions baked into the infrastructure
- An industry with a "mainframe-mentality", reluctant to change

Review

History:

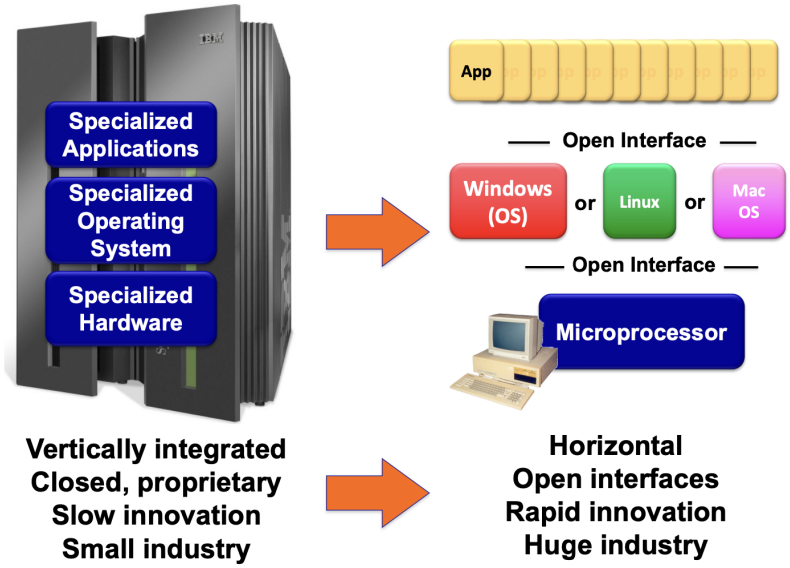
> Vertically Integrated / Closed / proprietary Slow innovation / Small industry

- Specialized APPs
- Specialized Operating System
- Specialized Hardware

Now:

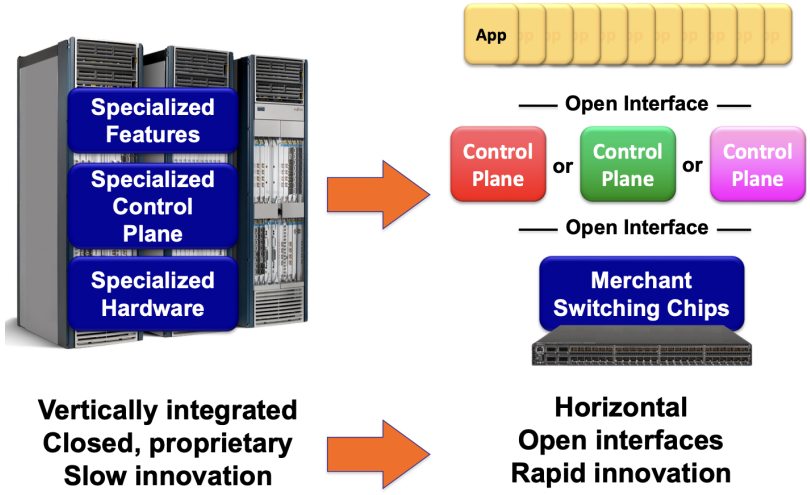
> Horizontal / Open Interfaces / Rapid Innovation / Huge Industry

- APP
  - Open Interface
- Windows(OS) / Linux / Mac OS
  - Open Interface
- Microprocessor



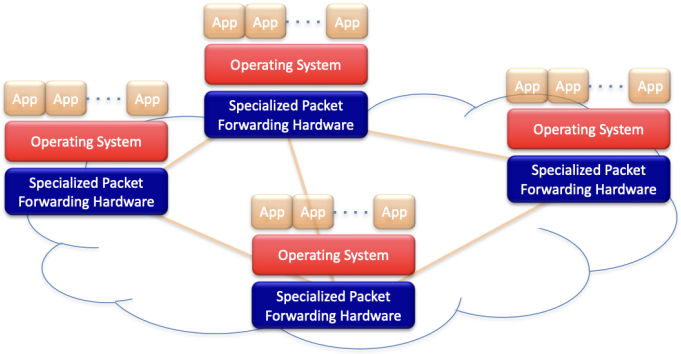
Future:

- APP
  - Open Interface
- Control Plane-1 / Control Plane-2 / ...
  - Open Interface
- Merchant Switching Chips

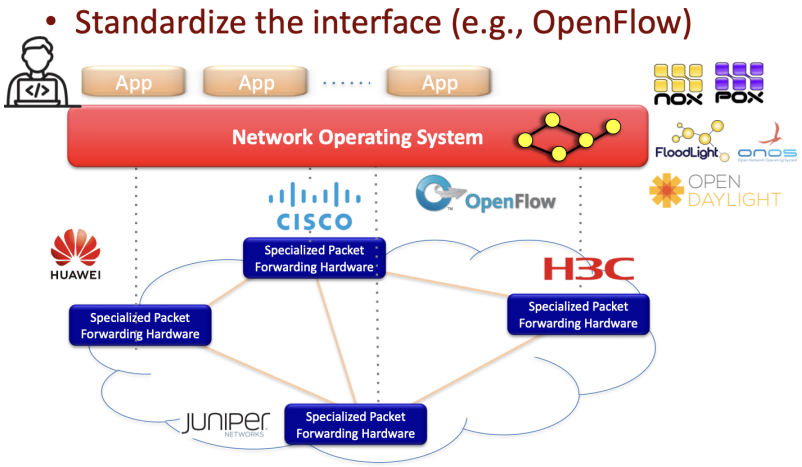


Software Defined Network

origin:



SDN:

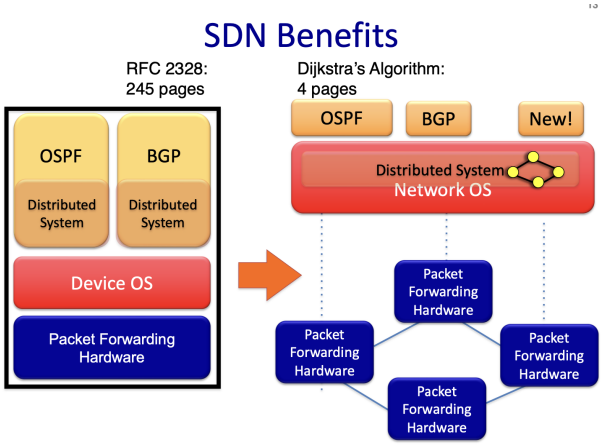


Cogitation (设计思想)

- Open the black box: decouple the control plane and data plane
- Make the control plane centralized
- (Form a centralized Network Operating System)
  - Examples
    - nox / pox
    - floodlight
    - onos
    - open daylight
- Standardize the Interface (eg: OpenFlow)
  - Switch send its state-message to Network OS
  - Network OS send its controlling message to Switch
  - Examples
    - CISCO
    - HUAWEI
    - Juniper Networks
    - H3C

Benefits

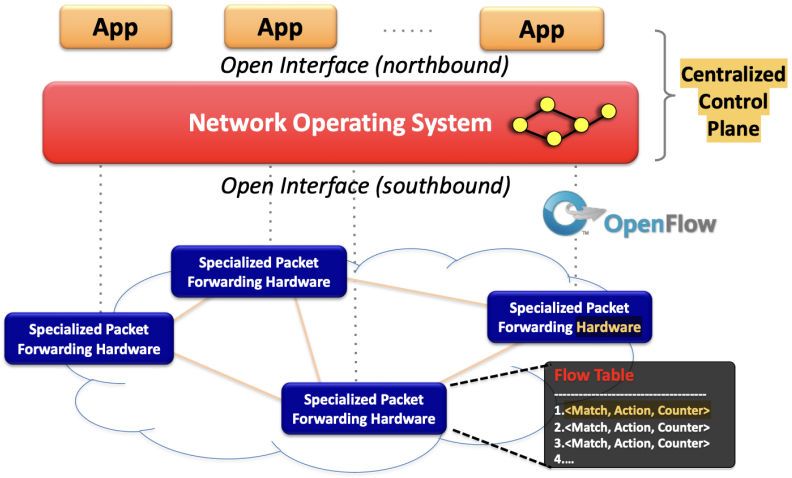
- Distributed ⇒ Centralized

- eg:
  - 

OpenFlow

Overview

- APP
  - Open Interface (Northbound)
- Network Operating System
  - Open Interface (Southbound)
- Specialized Packet Forwarding Hardware



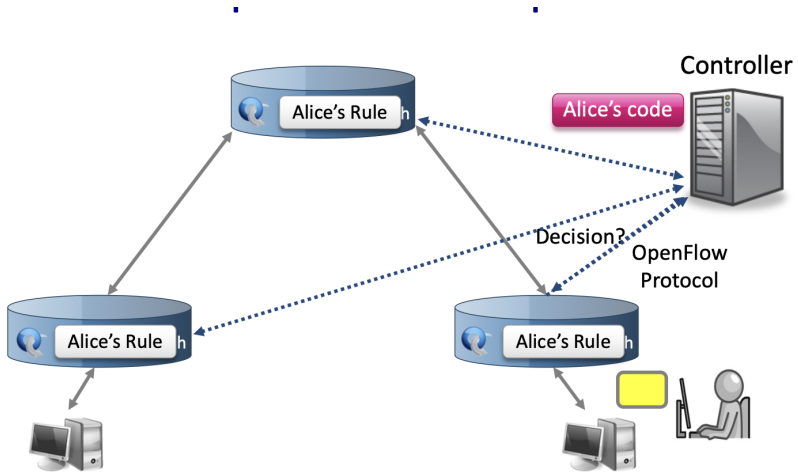
OpenFlow

- *Offer the Protocol* between Network OS and Switches
- *Define the actions* of Switches

OpenFlow

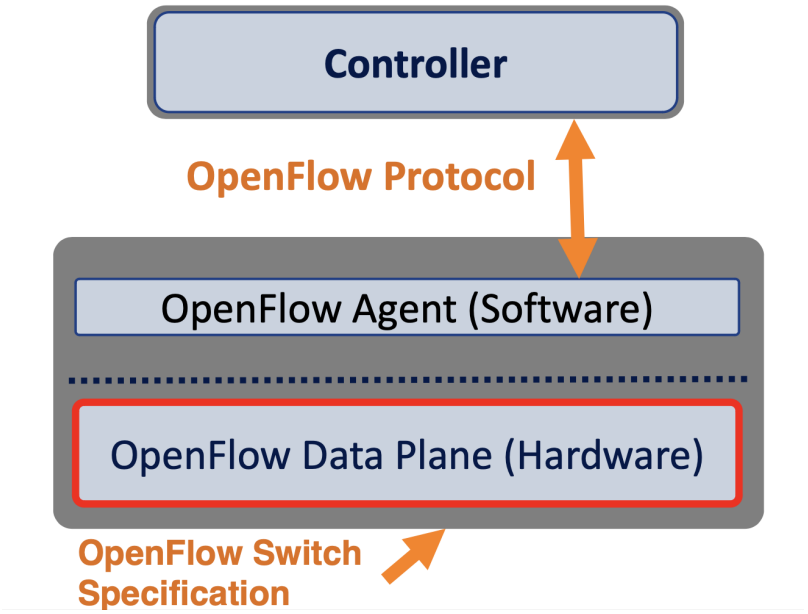
- Original Purpose: make it easy for experiments
  - Not practical to experiment with new network protocol

    - Not practical to persuade vendors (卖家) to provide an open, programmable platform on their switches and routers
    - Most modern switches and routers contain flow-tables that run at line-rate to implement firewalls, QoS, etc.
- Example
    - the administrator Alice applies the "Alice's code" on the Network OS (Controller)
    - A client called Bob sends an message to a switch
    - This switch receives the message and asks the controller what to do next via a TCP connection
    - The request is answered by "Alice's code" in controller and the decision is made here
    - The Network OS gives orders to those switches which are along the way
    - Then the Bob's message sent along those switches above



OpenFlow offloads control intelligence to a remote software

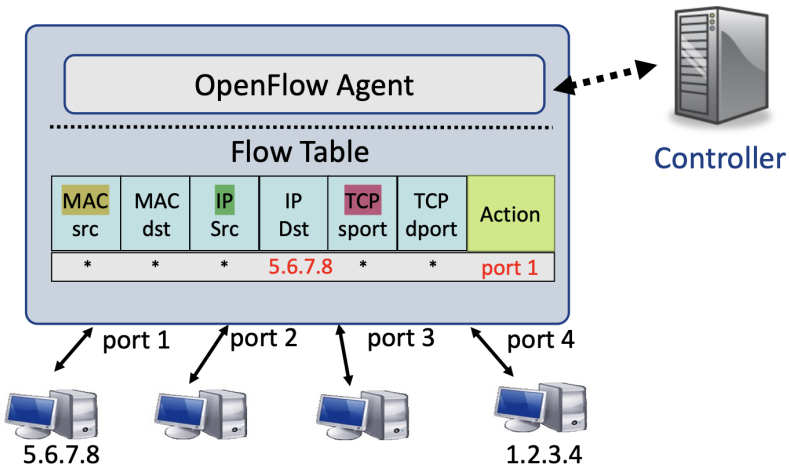
The Role of OpenFlow in SDN



1. OpenFlow Rule = **Match + Action**

2. We use a 2-element tuple to present Rule as (Match, Action)
- Match
  - Match on any header, or new header
  - Allows any flow granularity(粒度)
  - Only match on Header Part, not Data
  - “通配”可以出现在任何位置，没有限制
- Action
  - Forward to port(s), drop, send to controller
  - Overwrite header with, mask, push or pop
  - Forward at specific bit-rate (velocity controlling)

3. Hence, we offer a new concept "Flow Table"



OpenFlow Controller

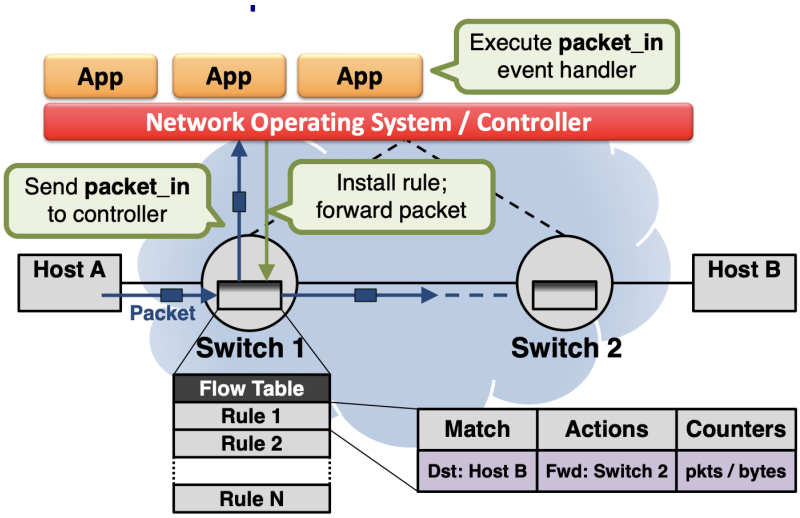
> the controller is just a software

- Receive **the events from the switches**
  - Topology changes
  - Traffic statistics
  - Arriving packets
- Make **decisions** and calculations, and etc.
- Offer **commands to switches**
  - (Un)install rules
  - Query statistics
  - Send packets

Reactive Control (响应式)

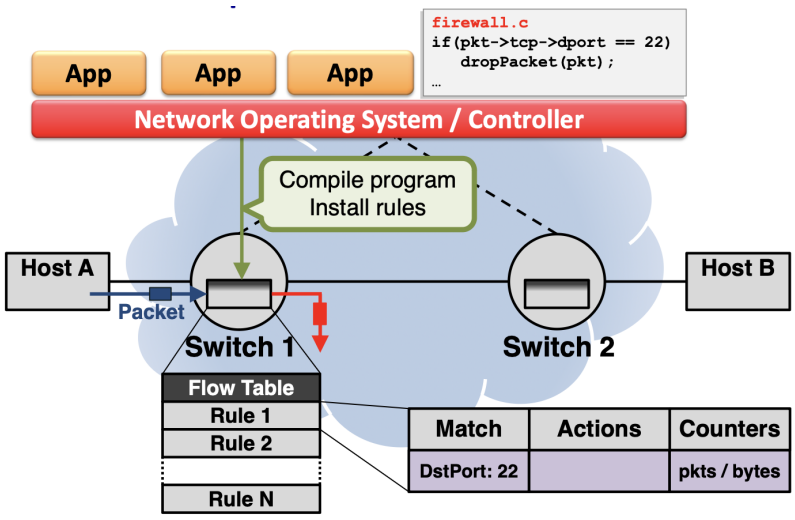
1. a packet is sent to one switch
2. if the rule is defined in this switch, use it (check by hardware, O(1));
3. else, the switch send packet_in message to controller

4. and then the Controller (Network OS) makes decision what's the packet's correct path
5. the APPs execute packet_in event handler
6. the Controller sends and installs rule to this switch for this forwarding packet
7. the rule installed contains Match + Actions + Counters
8. PS: if "Actions" is empty, it will discard the packet



Proactive Control (主动式)

1. the program is running on APP/NOS
2. Compile program install rules
3. if "Action" is empty, it will discard the packet
4. else, ……



SDN vs. OpenFlow

- SDN ≠ OpenFlow
  - SDN is a general network architecture, OpenFlow is a protocol or standard
- OpenFlow is an interface protocol for SDN
  - spoken by the controllers and switches
- OpenFlow is more than just a protocol

    - also including *switch specification*
    - can be seen as the de *factor standard implementation* of SDN: SDN / OpenFlow

SDN Applications

Dynamic Access Control (动态访问控制)

- Inspect(检查) first packet of a connection (检查权限...)
- Consult(参考) the access control policy
- Install rules to block or route traffic

Seamless Mobility / Migration (无缝迁移)

- See host send traffic at new location
- Modify rules to reroute the traffic

Server Load Balancing (负载均衡)

- Pre-install load-balancing policy
- Split traffic based on source IP

Others

- Routing
- Dynamic access control
- Seamless mobility/migration
- Server load balancing
- Network virtualization
- Using multiple wireless access points • Energy-efficient networking
- Adaptive traffic monitoring
- Denial-of-Service attack detection