臺灣大學 National Taiwan University



Software-Defined Networking

Prof. Ai-Chun Pang
Graduate Institute of Networking and Multimedia,
Dept. of Comp. Sci. and Info. Engr.,
National Taiwan University

Email: acpang@csie.ntu.edu.tw

http://www.csie.ntu.edu.tw/~acpang



Agenda

- What is Software-Defined Networking (SDN)?
- How does SDN work?
  - Infrastructure layer
  - Control layer
  - Application layer
- Research Issues
  - Scalability
  - Consistent network update
  - Flow scheduling
  - Security



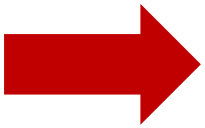
What is SDN?



Current Status & Motivation (1/2)

Source: Nick Mckeown, Stanford

- Specialized software
- Specialized firmware
- Specialized hardware
- Specialized interface



Control layer

Vertical-integrated **Horizontal-integrated**



Current Status & Motivation (2/2)

- Traditional network is manually configured
  - Operating error may cause network tear down
  - High **CAPEX** and **OPEX**
  - Network equipment is vulnerable to software bugs

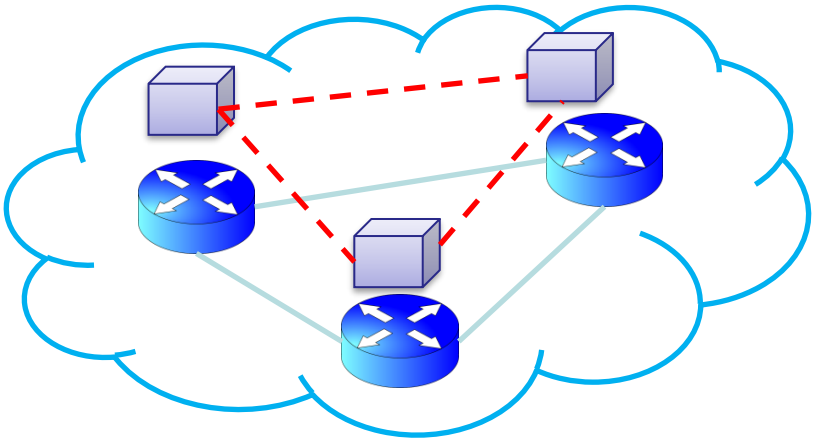
What is SDN?

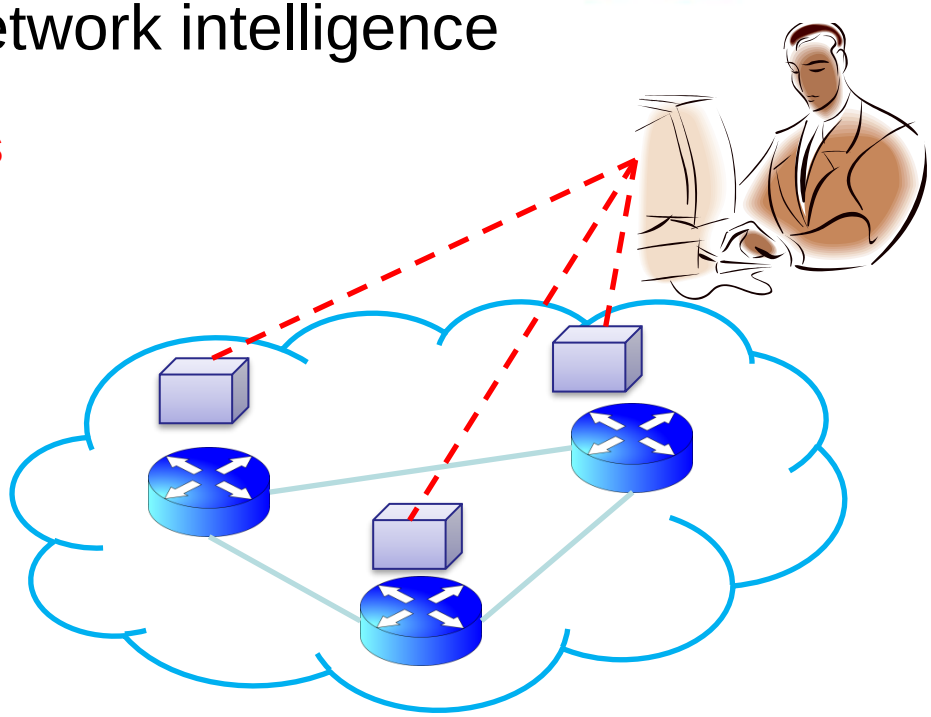
Source: Shmuel (Mooly) Sagiv, Tel Aviv Univ.

- SDN is an emerging network architecture
  - Decoupling of control and data planes
  - Logically centralizing network intelligence

ONF OPEN NETWORKING FOUNDATION

Control plane: Distributed algorithms
OSPF, BGP, ECMP, etc.
Data plane: Forwarding tables







Goal

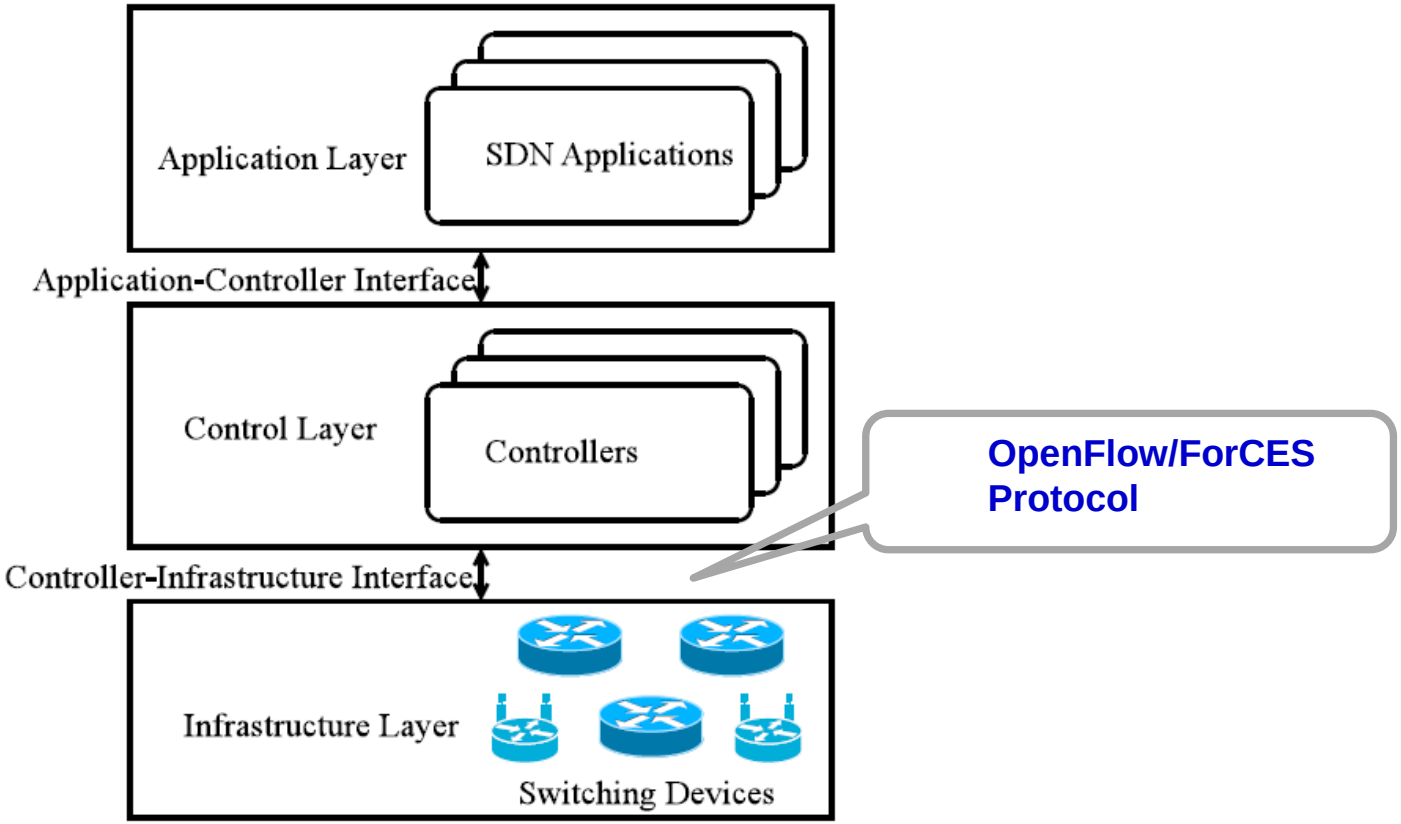
- Simplified and efficient network management
  - Programmable networks
    - Flexible and dynamically customizable networks
  - Network Operating System (NOS)
    - Provide global view
    - Ensure consistent network
  - Standard open interface (Northbound/southbound API)
    - Backward compatibility



How does SDN work?

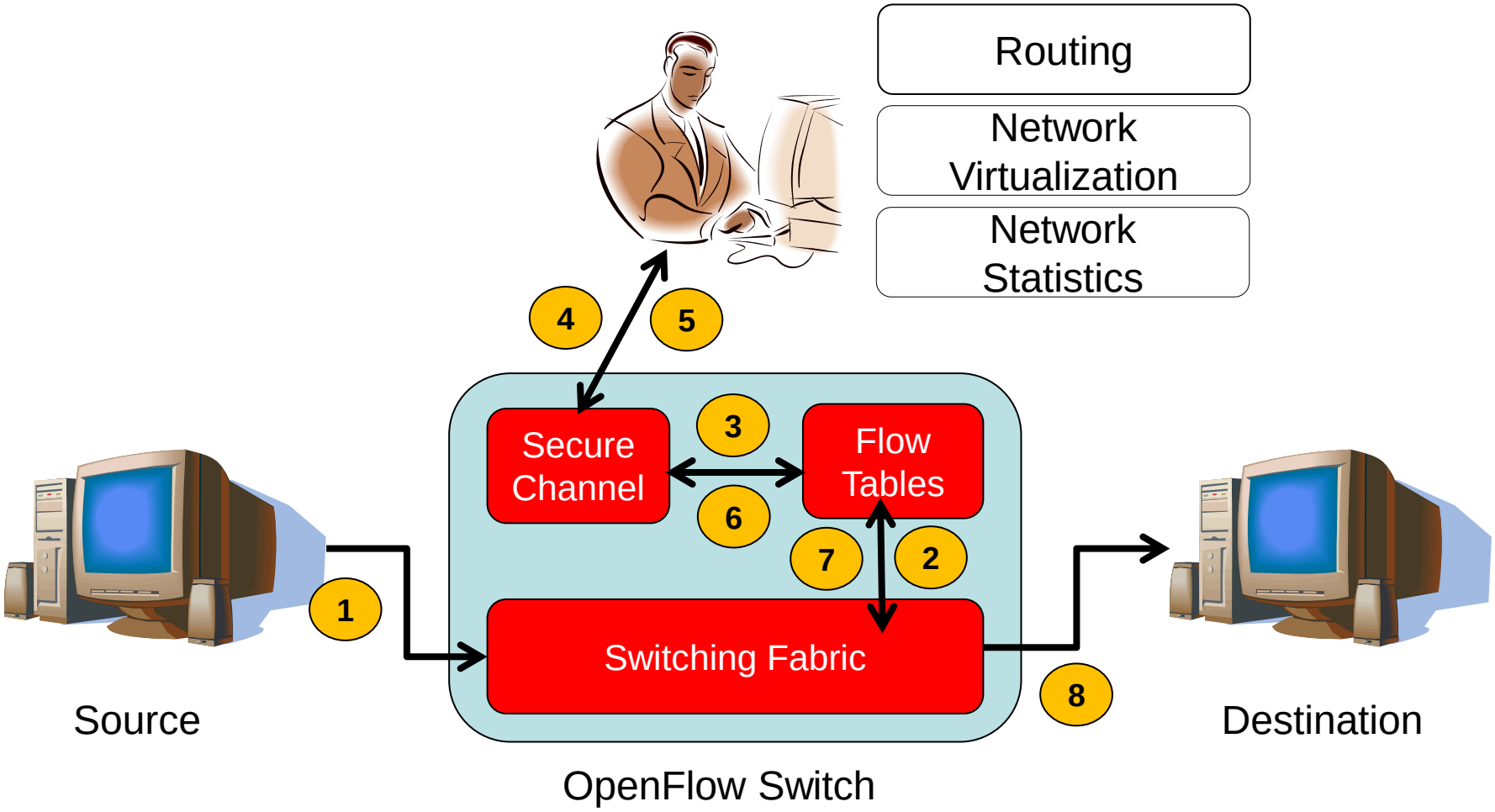
SDN Paradigm

- Reference model[1]





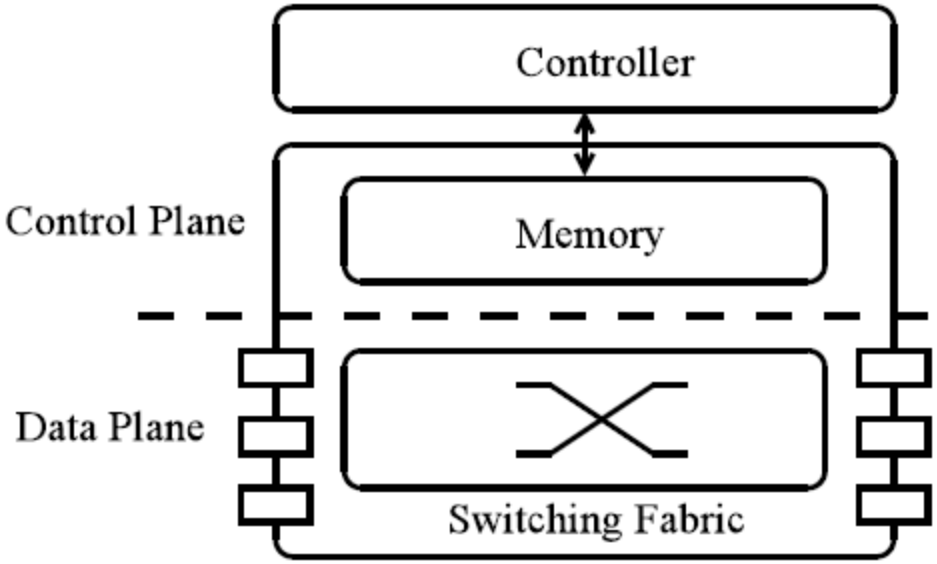
Operation


Routing
Network Virtualization
Network Statistics
4
5
Secure Channel
3
6
Flow Tables
7
2
1
Switching Fabric
8
Source
Destination
OpenFlow Switch



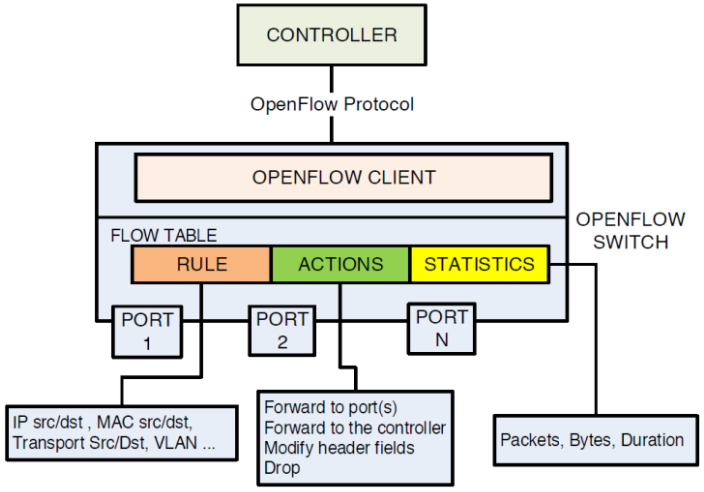
Infrastructure Layer[1]

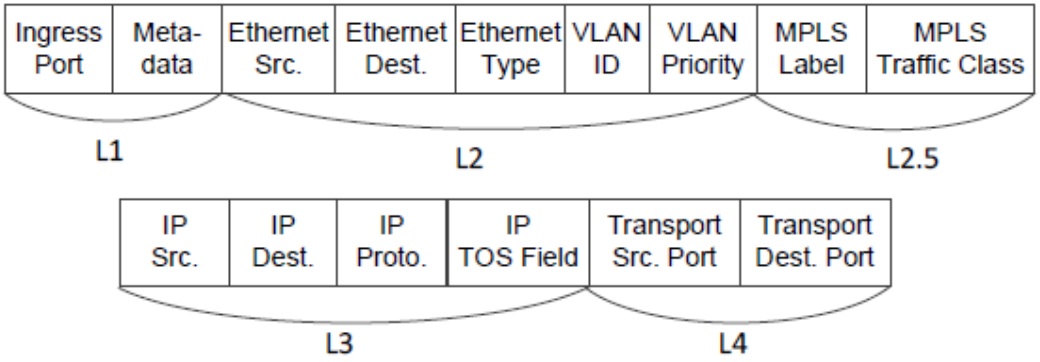
- Network equipment: router, switch, and middlebox
  - Control plane: **flow table** and **secure channel**
  - Data plane: **packet switching/forwarding**



Flow Table in Switch[2-3]

- SDN commonly uses TCAM to store rules in flow tables
  - **Match fields**: Match packets based on packet's header
  - **Action set**: Forward, drop, and modify
  - **Statistics**: Bytes, packets, duration



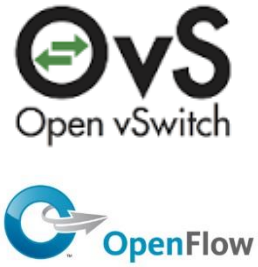


TCAM (Ternary content-addressable memory)



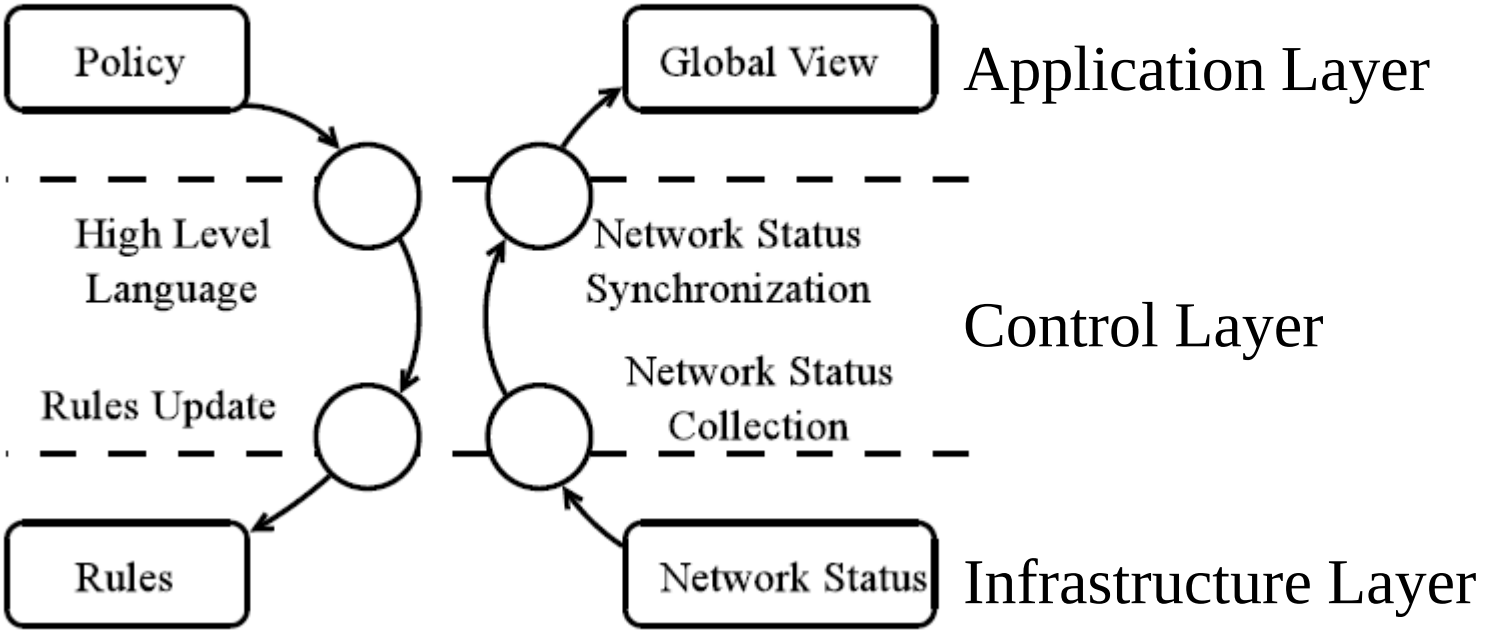
SDN Switching Devices[1]

OpenFlow Switch	Wireless AP	Network Hardware	Vendor
Representative	OpenWRT	NetFPGA	Pica8
Processing Speed	Low	Middle	High
Flexibility	High	Middle	Low



Control Layer

- Software-based SDN controller
  - Provide consolidated control functionality by open interface[1]





OpenFlow Controllers

Controller	Open Source	Language	Multi-threaded	GUI	Origin
NOX [32]	yes	C++/Python	no	yes	Nicira Networks
NOX-MT [46]	yes	C++	yes	no	Nicira Networks and Big Switch Networks
POX [47]	yes	Python	-	yes	Nicira Networks
Maestro [48]	yes	Java	yes	no	Rice University
Beacon [49]	yes	Java	yes	yes	Stanford University
SNAC [50]	no	C++/Python	no	yes	Nicira Networks
RISE [51]	yes	C and Ruby	non-guaranteed	no	NEC
Floodlight [52]	yes	Java	-	yes	Big Switch Networks
McNettle [53]	yes	Nettle/Haskell	no	no	Yale University
MUL [54]	yes	C	yes	yes	KulCloud
RYU [55]	yes	Python	-	-	NTT OSRG and VA Linux
OpenDaylight [56]	yes	Java	yes	yes	Multiple contributors



Application Layer

- Develop SDN applications/policies to manage the network
  - Using high-level API provided by controller
- SDN applications
  - Access control
  - Load balancing
  - Network virtualization
  - Energy efficiency



SDN Research Issues



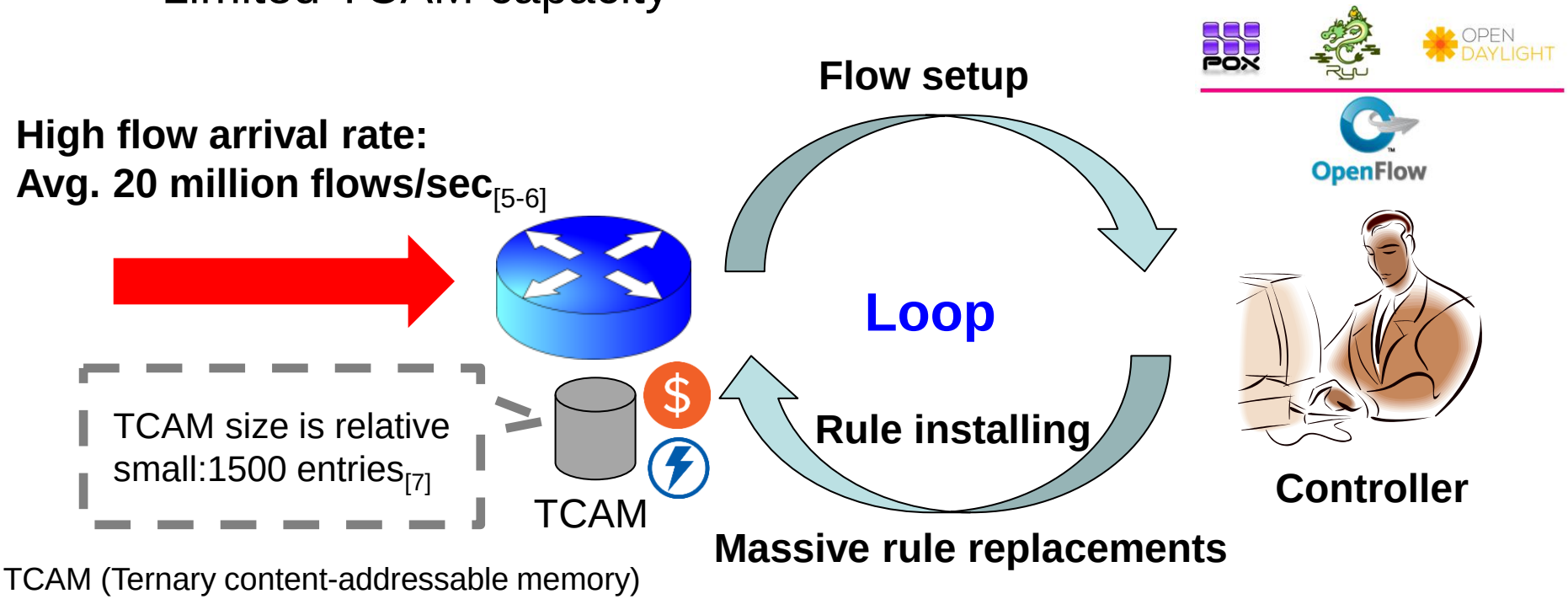
SDN Research Issues

- Controller scalability
- Consistent network update
- Flow scheduling
- Security

Controller Scalability

- Massive flow setups sent to controller
  - High flow arrival rate
  - Fine-grained flow control
  - Limited TCAM capacity

Why?

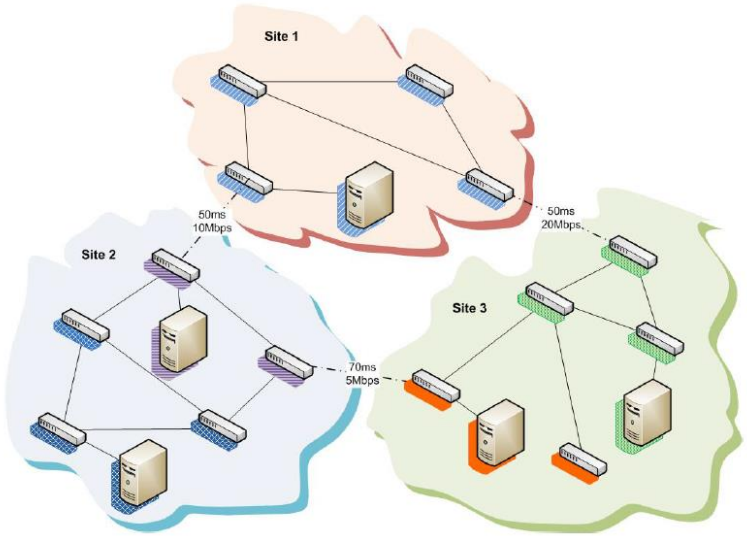


TCAM (Ternary content-addressable memory)



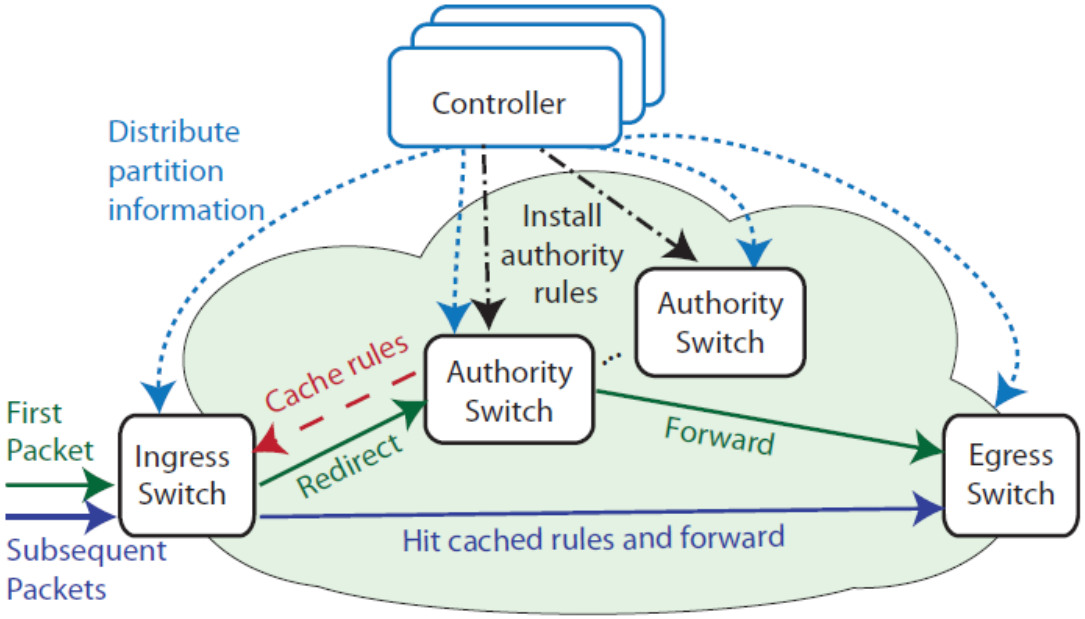
Solutions for Scalability (1/2)

- Capability enhancement for the centralized controller[8]
  - Parallelism mechanisms (Multi-threading and multi-core CPU)
  - I/O batching
- Cooperation among multiple distributed controllers[9-10]
  - Leverage multiple controllers to share the handling of flow setup requests
  - Horizontal/vertical control models



Solutions for Scalability (2/2)

- Switch-assisted[1]
  - Keep flow setups in data plane
    - Redirect flow setup sent to an authority switch





Consistent Network Update (1/2)

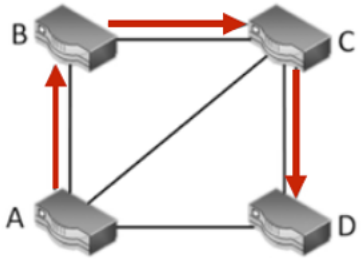
- What is network update?
  - Change network **state** to achieve some **goal**, e.g.,
    - **Goal**: VM migration
    - **State**: Forwarding entries and traffic distribution

- Problem may occur[12]
  - 20% of failures come from careless planned maintenance
    - Forwarding black-hole/forwarding loop
    - Link congestion and policy violation
- Consistent network update
  - **Prevent** specific problems during network update



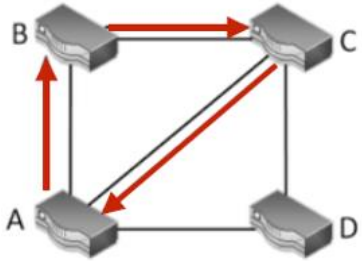
Consistent Network Update (2/2)

- Forwarding loop
  - Reason: **Asynchronous switch update**



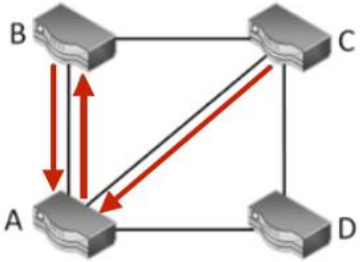
A → B
B → C
C → D

Initial State



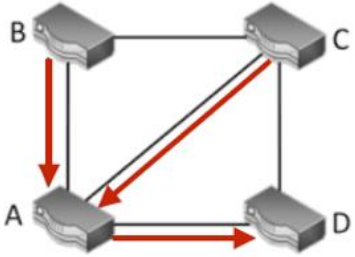
A → B
B → C
C → A

Loop State 1



A → B
B → A
C → A

Loop State 2

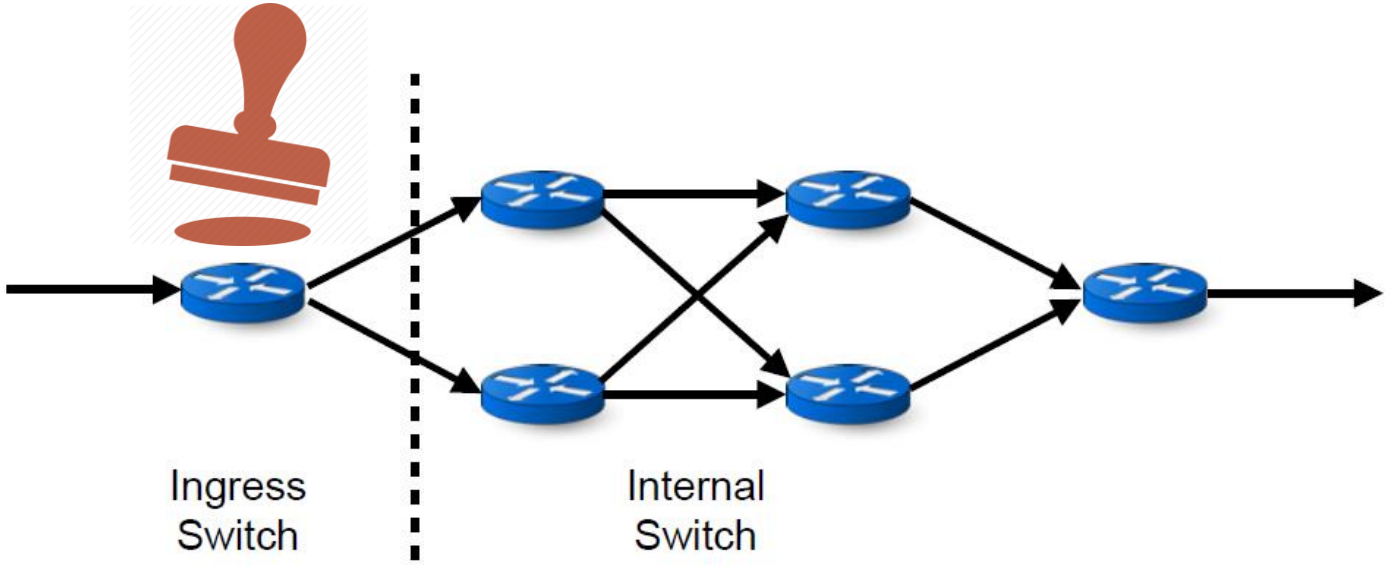


A → D
B → A
C → A

Final State

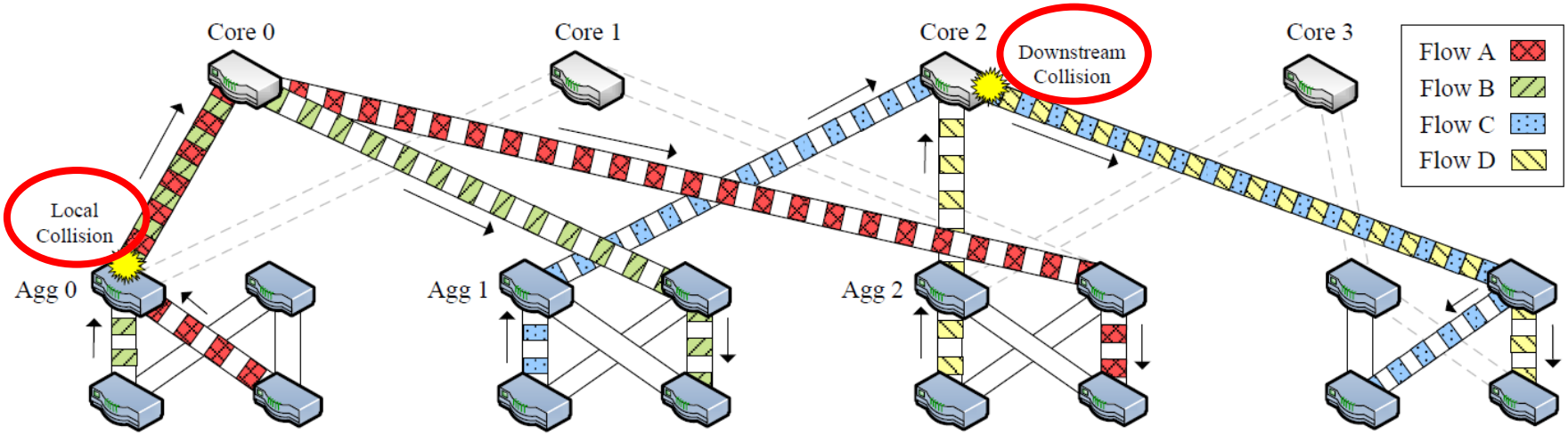
Solutions for Consistent Network Update

- Goal: Ensure common consistent properties
  - Blackhole/loop free, congestion free, and waypoint enforcement
- A simple solution: Two-phase update[13]
  - Add new rules into internal switches and ingress switch
  - Stamp packets with new version tag (VLAN or MPSL labeling)



Flow Scheduling

- Traditional routing protocol may cause substantial bandwidth loss due to long-term collisions
  - Example: ECMP[14]



ECMP (Equal-Cost Multi-Path routing)

Solution for Flow Scheduling

- Observation
  - Network congestions are mainly caused by elephant/large flows
- Hedera[14]: A well-known solution in data center

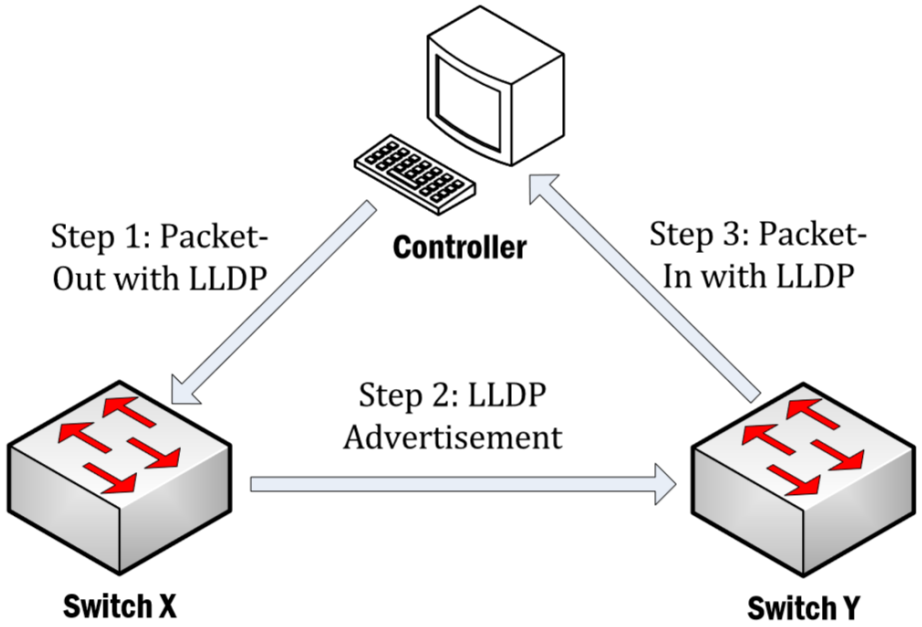


Security Issues

- SDN enables new opportunities to solve some legacy network security issues, and also faces several new challenges
  - Existing issues
    - DDoS attack
    - Network scanning attack
  - New SDN issues
    - **Link fabrication attack**[15]
    - Policy enforcement attack

Link Fabrication Attack (1/2)

- SDN application requires topology to control the network behavior
- Topology discovery service[15]
  - LLDP packet

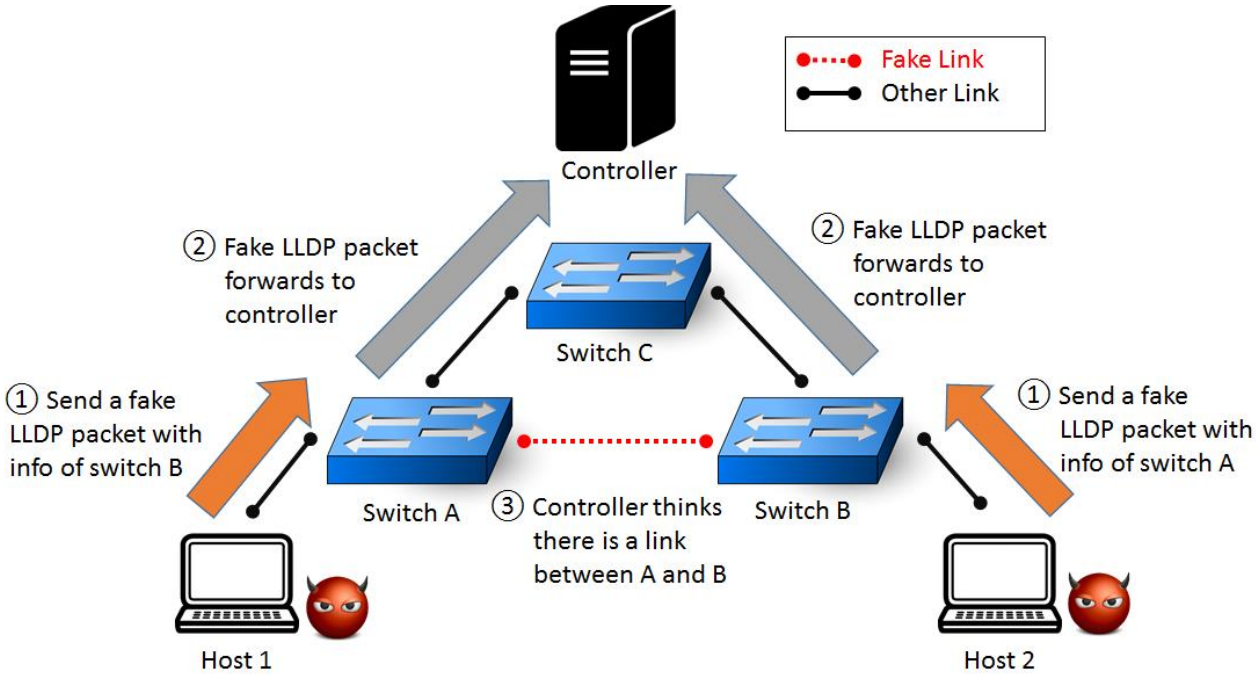


LLDP (Link Layer Discovery Protocol)



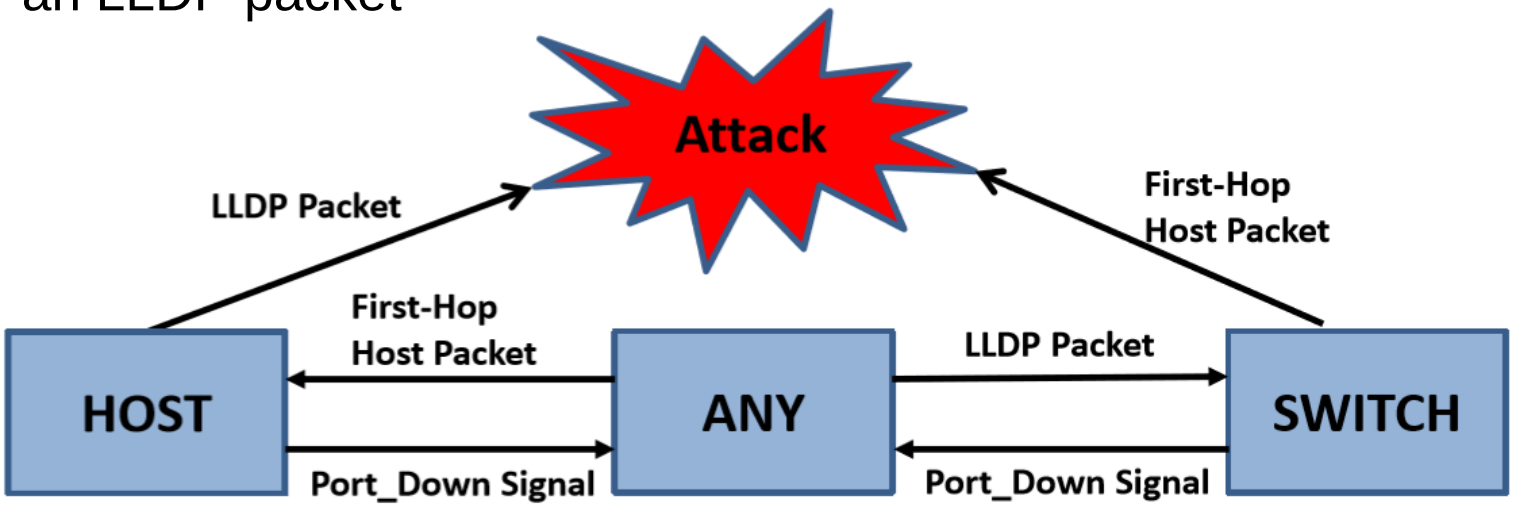
Link Fabrication Attack (2/2)

- However, topology discovery services provided by controllers can be tricked by adversaries
  - **Link fabrication attack**[15]



Solution for Link Fabrication Attack

- Each port of switch has a flag to represent whether the port is connecting to a host/switch
- Controller uses the transition graph to update the flag value and detect the attack
  - For example: An alert is raised if a port connecting to a host receives an LLDP packet





Reference

[1] W. Xia, Y. Wen, C. H. Foh, D. Niyato and H. Xie, “A Survey on Software-Defined Networking,” in *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 27-51, 2015.

[2] Y. Jarraya, T. Madi, and M. Debbabi, “A Survey and a Layered Taxonomy of Software-Defined Networking, ” IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp.1955-1980, 2014.

[3] B. A. A. Nunes, M. Mendonca, X. N. Nguyen, K. Obraczka, and T. Turletti, “A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks, ” IEEE Communications Surveys & Tutorials, vol. 16, no. 3, pp.1617-1634, 2014.

[4] F. Hu, Q. Hao, and K. Bao, “A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation, ” IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 2181-2206, 2014.

[5] A. Tavakoli, M. Casado, and S. Shenker. Applying nox to the datacenter. HotSDN, pages 1–6, 2009.

[6] M. Moshref, M. Yu, A. Sharma, and R. Govindan. Scalable rule management for data centers. USENIX Conference, pages 157–170, 2013.

[7] A. R. Curtis, J. C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and S. Banerjee. Devoflow: Scaling flow management for high performance networks. ACM SIGCOMM Comput. Commun. Rev., pages 254–265, 2011.

[8] David Erickson. The beacon openflow controller. ACM SIGCOMM Workshop on HotSDN, pages 13–18, 2013.



Reference

[9]R. Sherwood, M. Chan, A. Covington, G. Gibb, M. Flajslik, N. Handigol, T. Y. Huang, P. Kazemian, M. Kobayashi, J. Naous, S. Seetharaman, D. Underhill, T. Yabe, K. K. Yap, Y. Yiakoumis, H. Zeng, G. Appenzeller, R. Johari, N. McKeown, and G. Parulkar. Carving research slices out of your production networks with openflow. SIGCOMM Comput. Commun. Rev., 40(1):129–130, 2010.

[10] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker. Onix: A distributed control platform for large-scale production networks. USENIX Conference, pages 1–6, 2010.

[11] M. Yu, J. Rexford, M. J. Freedman, and J. Wang. Scalable flowbased networking with difane. ACM SIGCOMM, pages 351–362, 2010.

[12] A. Markopoulo, G. Iannaccone, S. Bhattacharyya, C.N. Chuah, Y. Ganjali, and C. Diot, “Characterization of Failures in an Operational IP Backbone Network,” IEEE/ACM Transactions on Networking, 2008.

[13] M. Reitblatt, N. Foster, J. Rexford, C. Schlesinger, D. Walker, “**Abstractions for network update**,” ACMSIGCOMM, 2012.

[14] M. A. Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, “Hedera: dynamic flow scheduling for data center networks, ” In *Proceedings of the 7th USENIX conference on Networked systems design and implementation* (NSDI), pp. 1-12, 2010.

[15] S. Hong, et al. "Poisoning Network Visibility in Software-Defined Networks: New Attacks and Countermeasures.“ in NDSS, 2015.