高级网络安全研究与应用——

安全需求与安全应用

北京邮电大学

郑康锋 伍淳华

kfzheng@bupt.edu.cn wuchunhua@bupt.edu.cn

安全需求与安全应用——

请看图猜

Examples

Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6

Flow Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:20..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop

Examples

Routing

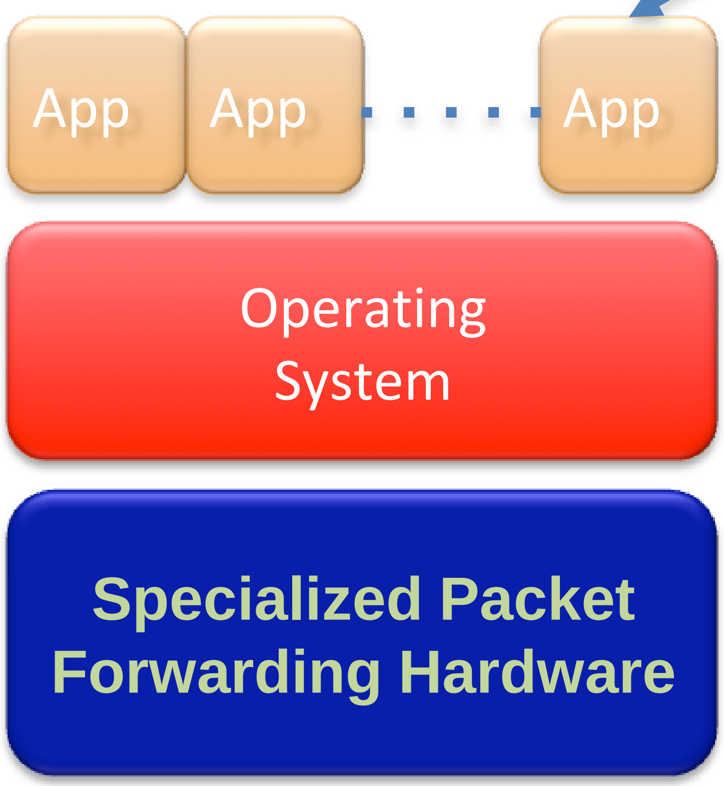
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	5.6.7.8	*	*	*	port6

VLAN Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f..	*	vlan1	*	*	*	*	*	port6, port7, port9

e have lost our way

Routing, management, mobility management, access control, VPNs, …



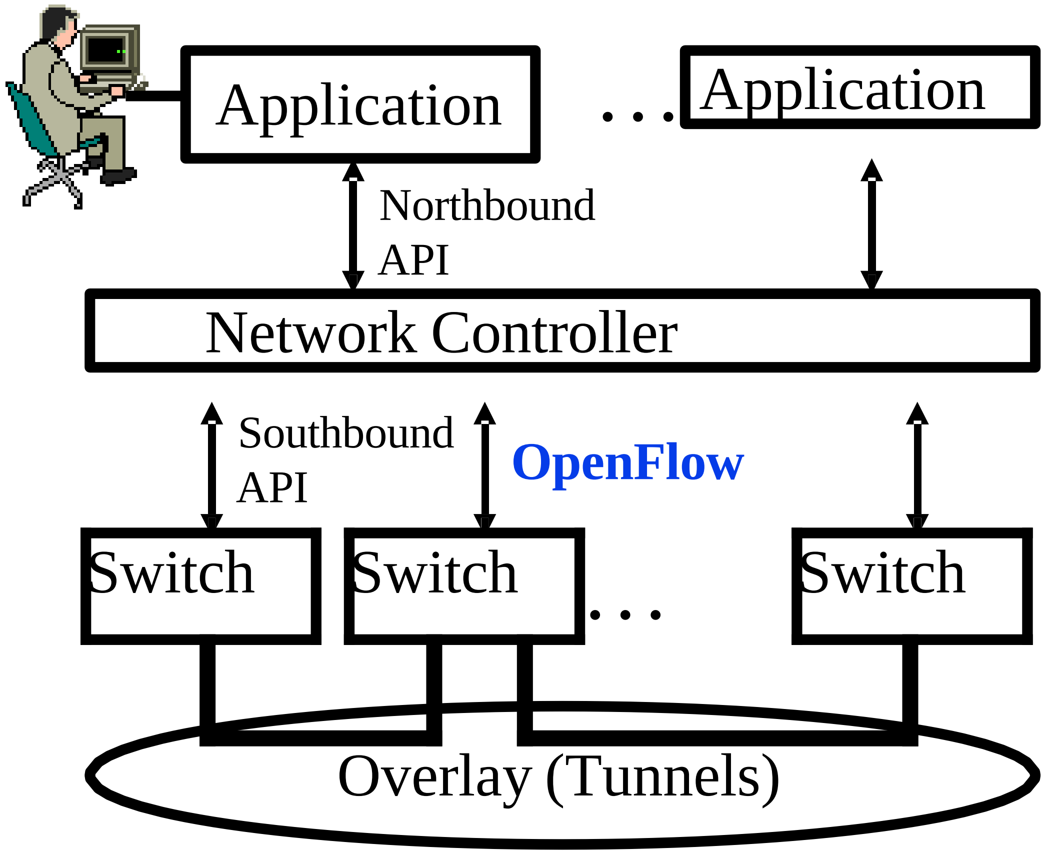
Million of lines of source code	5400 RFCs	Barrier to entry
500M gates 10Gbytes RAM	Bloated	Power Hungry

Many complex functions baked into the infrastructure

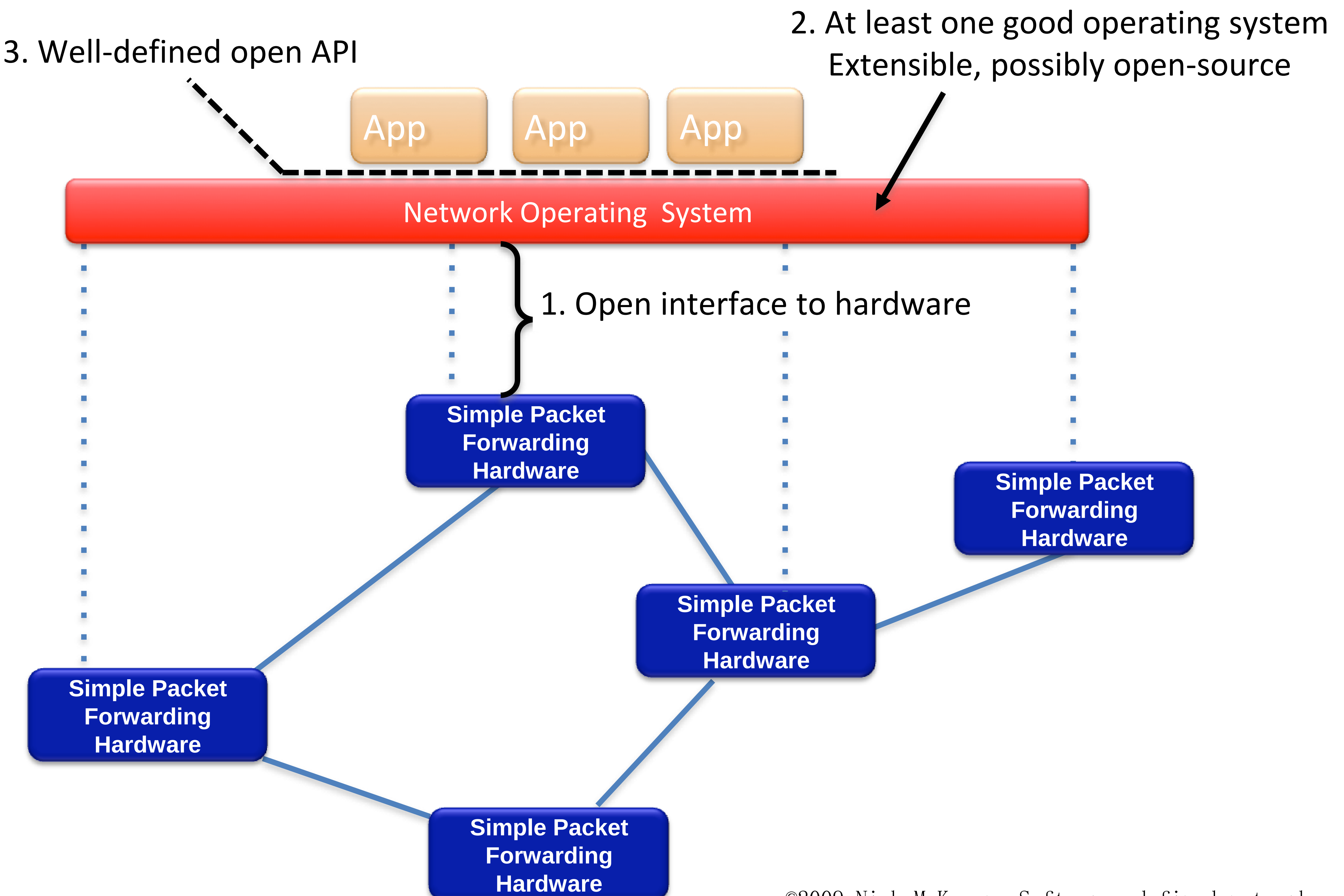
OSPF, BGP, multicast, differentiated services, Traffic Engineering, NAT, firewalls, MPLS, redundant layers, …

Origins of SDN

- SDN originated from OpenFlow
- Centralized Controller
  ⇒ Easy to program
  ⇒ Change routing policies on the fly
  ⇒ Software Defined Network (SDN)
- Initially, SDN=
  - Separation of Control and Data Plane
  - Centralization of Control
  - OpenFlow to talk to the data plane
- Now the definition has changed significantly.



The "Software-defined Network"



Original Definition of SDN

"*What is SDN?*

The physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices."

1. Directly programmable
2. Agile: *Abstracting control from forwarding*
3. Centrally managed
4. Programmatically configured
5. Open standards-based vendor neutral

The above definition includes *How*.

Now many different opinions about *How*.

⇒SDN has become more general. Need to define by *What*?

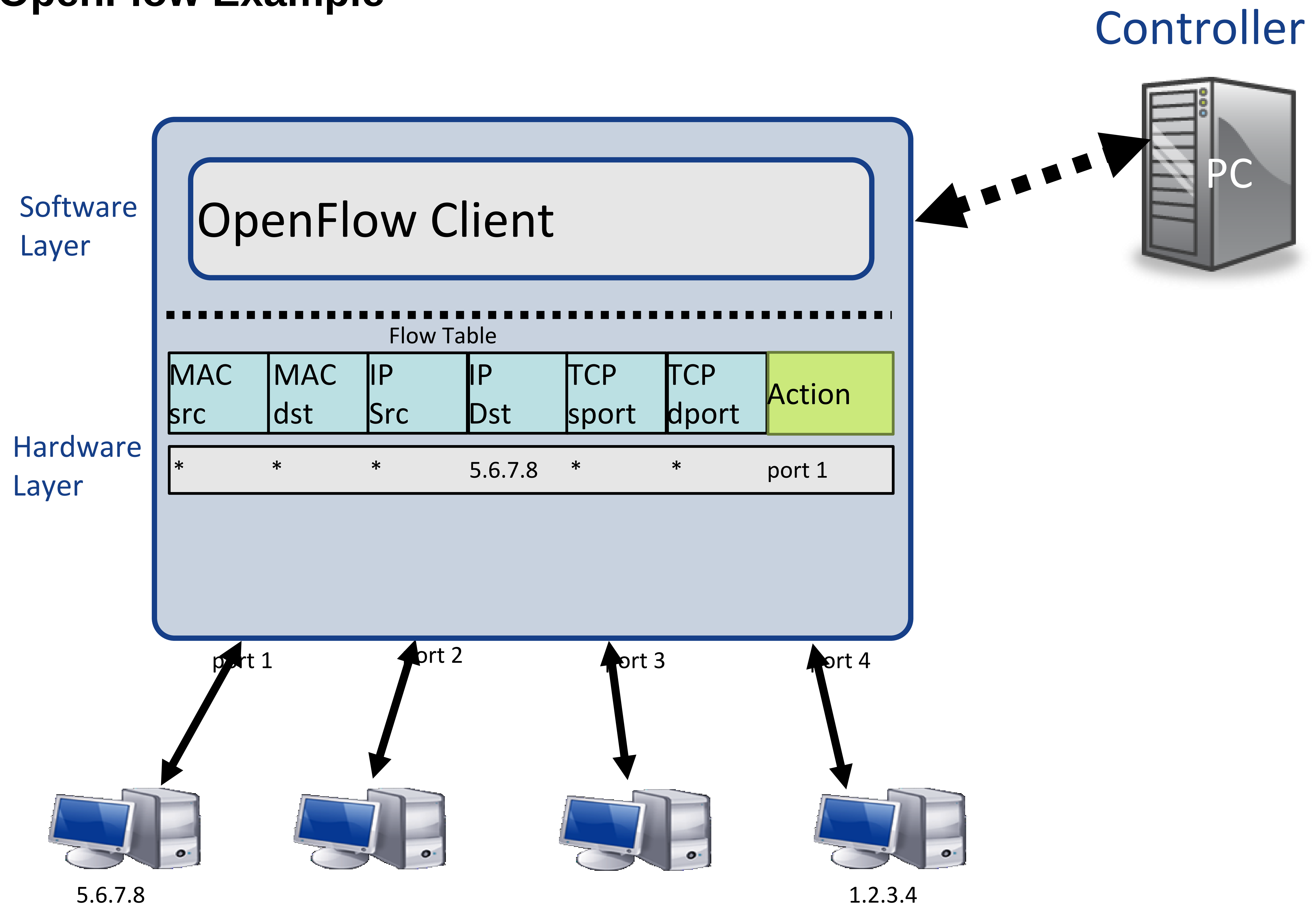
Ref: https://www.opennetworking.org/index.php?option=com_content&view=article&id=686&Itemid=272&lang=en

SDN Definition

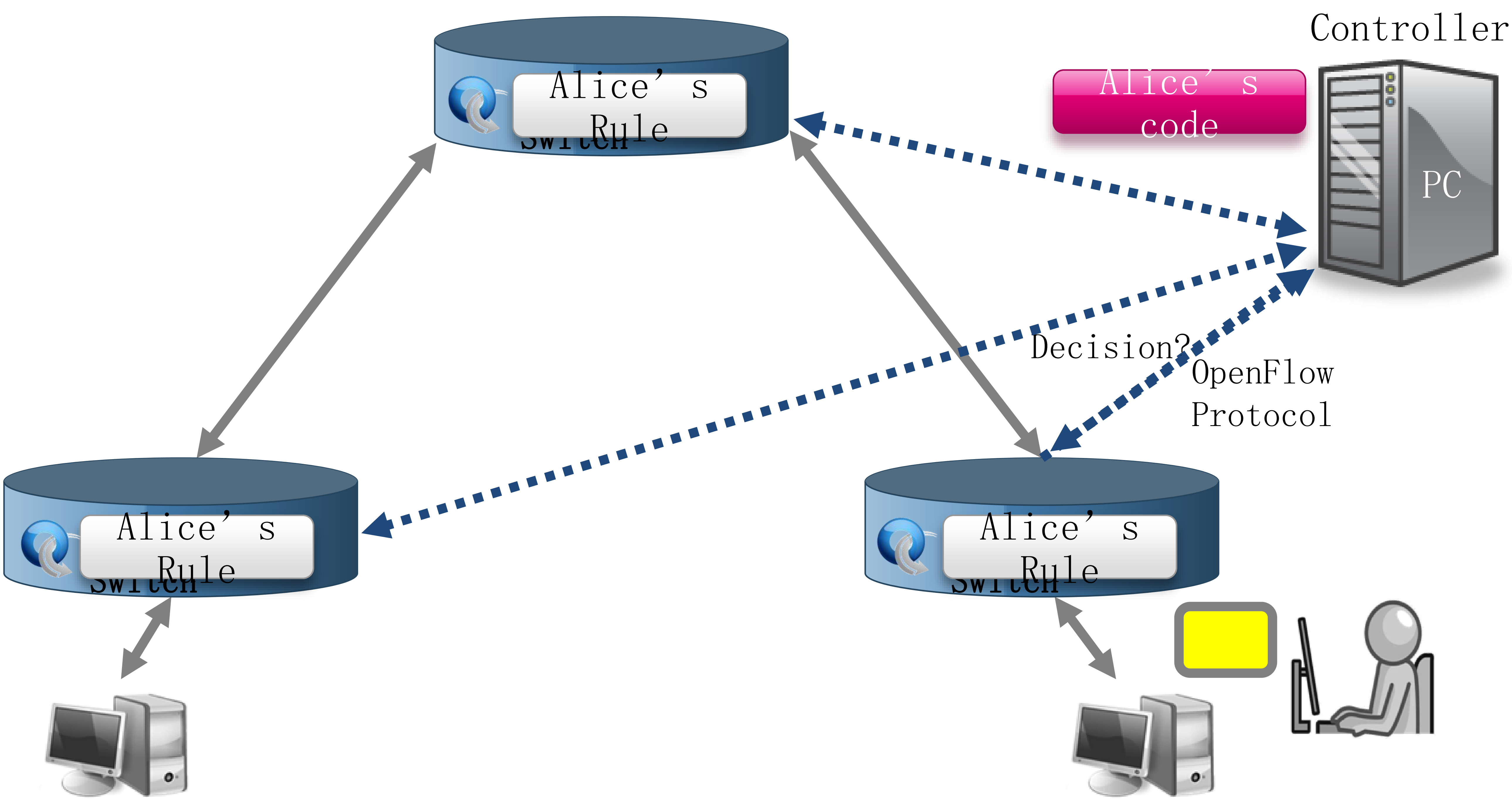
- SDN is a *framework* to allow network administrators to *automatically* and dynamically manage and control a *large number* of network devices, *services*, topology, traffic paths, and packet handling (quality of service) policies using high-level languages and APIs. Management includes provisioning, operating, *monitoring*, optimizing, and managing FCAPS (faults, configuration, accounting, *performance*, and security) in a *multi-tenant* environment.
- Key: Dynamic $\Rightarrow$ Quick

**

OpenFlow Example



OpenFlow usage



OpenFlow offloads control intelligence to a remote software

OpenFlow Basics

Flow Table Entries

Rule	Action	Stats

Packet + byte counters

1. Forward packet to zero or more ports
2. Encapsulate and forward to controller
3. Send to normal processing pipeline
4. Modify Fields
5. Any extensions you add!

Switch Port	VLAN ID	VLAN pcp	MAC src	MAC dst	Eth type	IP Src	IP Dst	IP ToS	IP Prot	L4 sport	L4 dport

\+ mask what fields to match

Examples

Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6

Flow Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:20..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop

Examples

Routing

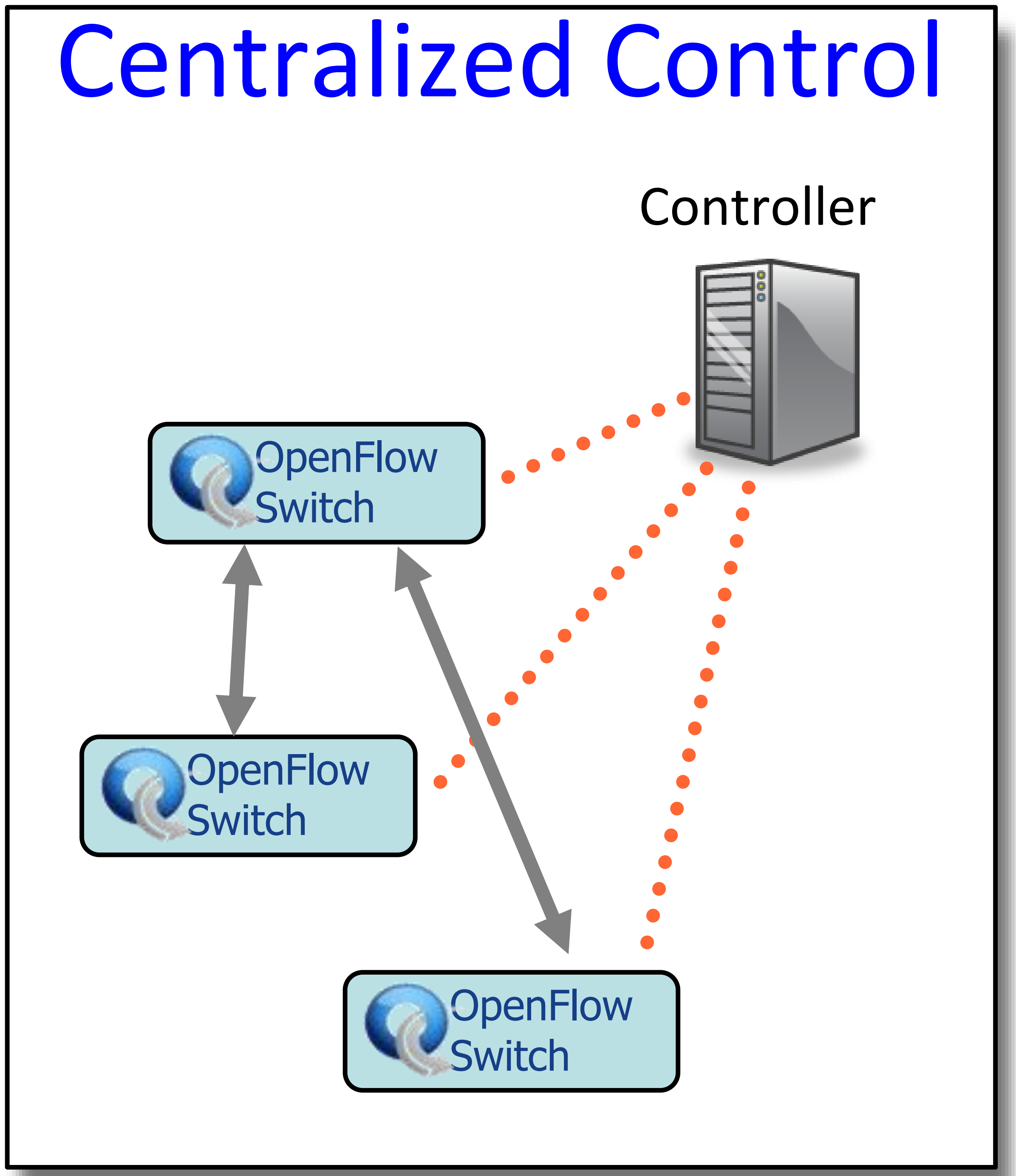
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	5.6.7.8	*	*	*	port6

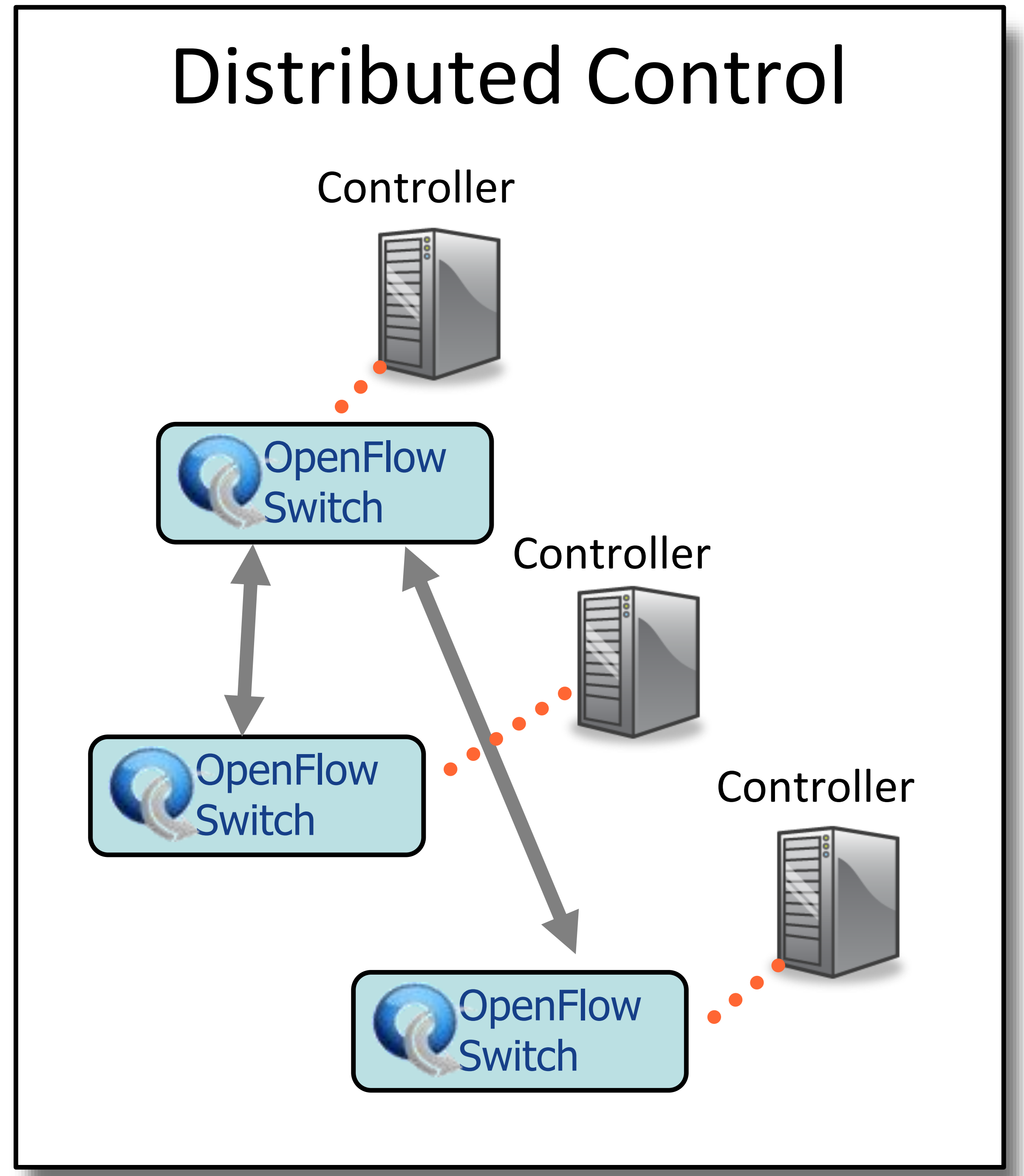
VLAN Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f..	*	vlan1	*	*	*	*	*	port6, port7, port9

Centralized vs Distributed Control

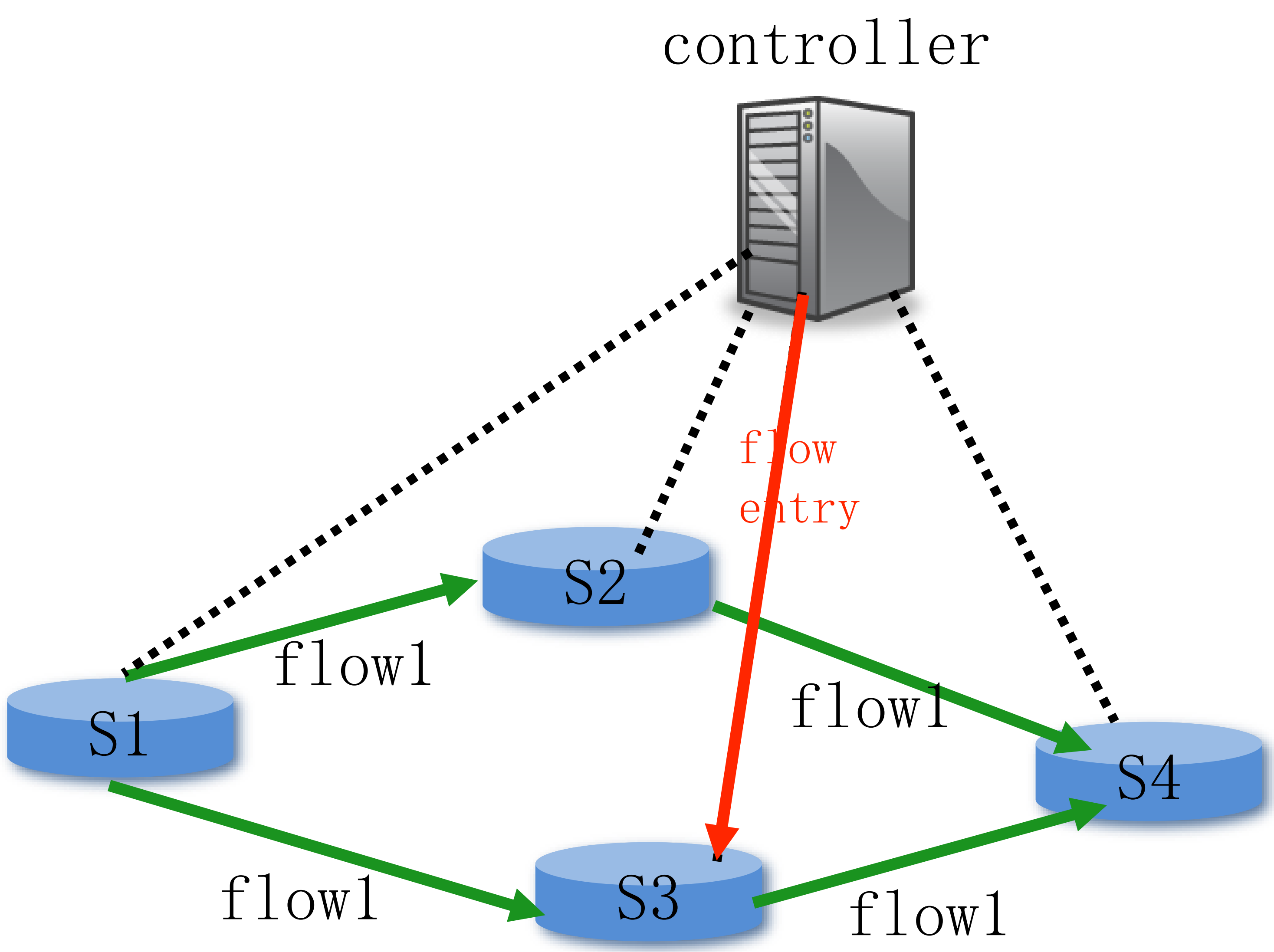
Both models are possible with OpenFlow





Traffic migration

- SDN is used as a tool to achieve the goal of migration.
- As the SDN controller can gain a central view of whole network, also with the help of flowtable.

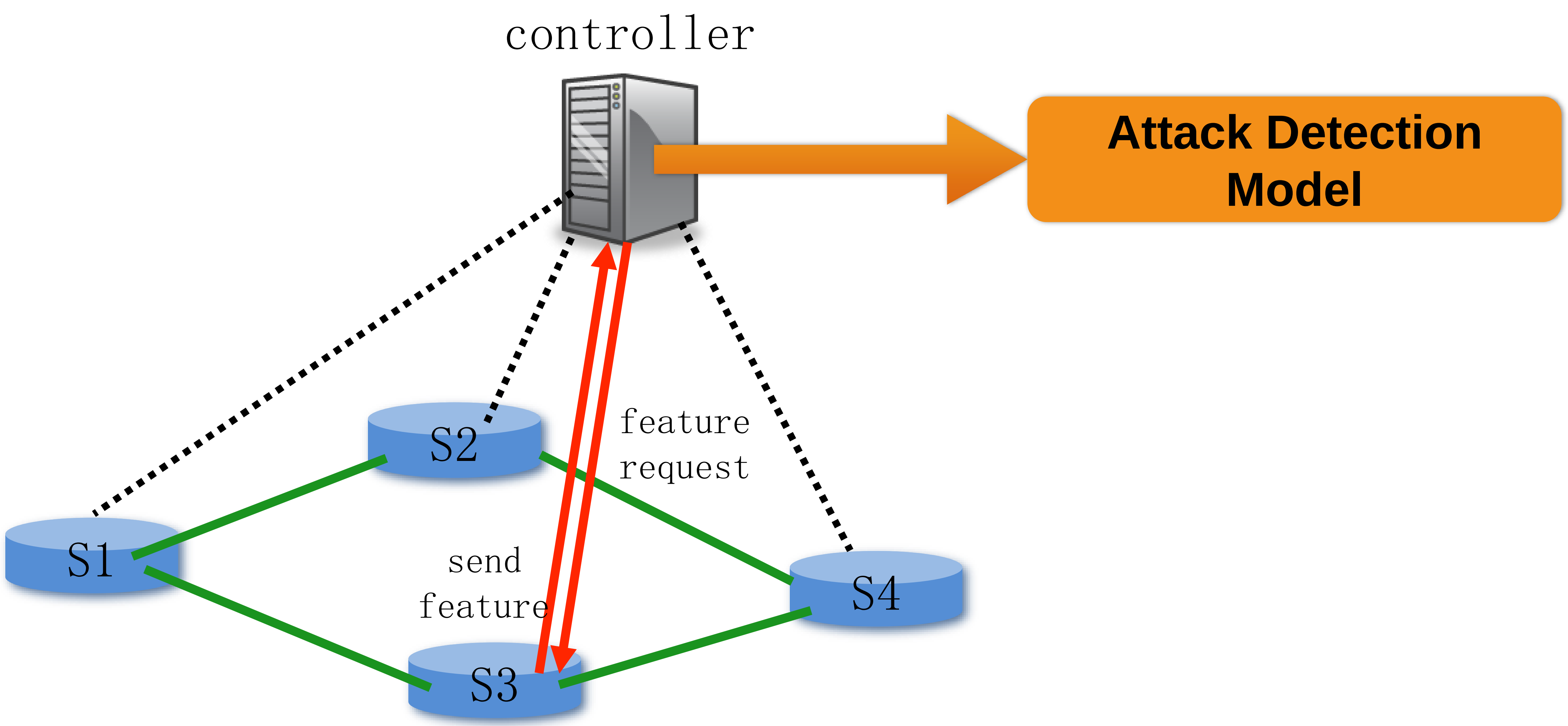


S2's FlowTable			
flow	from	to	action
empty			

S3's FlowTable			
flow	from	to	action
flow1	port 1	port 2	forward

Attack detection

- Using southbound interface(eg. OpenFlow protocol) to collect the features of traffic



Moving target defense

- Moving target defense(MTD) is a new way to make the network "dynamic" to attackers' view.
- So far, researches have proposed about three kind of methods
  - Network mapping and reconnaissance protection
  - Service version and OS hiding
  - Random host or route mutation

Network mapping and reconnaissance protection

Algorithm 1 MTD against network reconnaissance

Require: Probabilities $Pr_{SA} < Pr_A < Pr_{PA} < Pr_R < 1$

```
hash table action_buffer ← NULL
while (new TCP packet p is received) do
  if (p is illegitimate traffic) then
    if (p.dest_port not in action_buffer) then
      r ← random real number ∈ [0, 1]
      store r in action_buffer
    else
      r ← as in action_buffer
    end if
    switch (r)
     case r < Pr_SA:
      respond with TCP SYN-ACK
     case r < Pr_A:
      respond with TCP ACK
     case r < Pr_PA:
      respond with TCP PUSH-ACK with random payload
     case r < Pr_R:
      respond with TCP RST packet
     default:
      drop silently
    end switch
  end if
end while
```

Randomly respondence;

Protect from DoS attacks

Kampanakis, Panos, Harry Perros, and Tsegereda Beyene. "SDN-based solutions for Moving Target Defense network protection." *A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on*. IEEE, 2014.

Service version and OS hiding

- Service version and OS hiding

Algorithm 2 MTD against OS fingerprinting

```
while (new TCP packet p destined to target is received) do
  if (p is illegitimate traffic) then
    if (p has TCP SYN set) then
      s ← random 32-bit number
      respond with TCP SYN-ACK and s as the seq#
    else
      generate random payload and respond
    end if
  end if
end while
```

When the TCP SYN-ACK seq has the information about OS or service version

Kampanakis, Panos, Harry Perros, and Tsegereda Beyene. "SDN-based solutions for Moving Target Defense network protection."
A World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on. IEEE, 2014.

Random Host Mutation

- Features:
  - 1. IP mutation is transparent to end host;
  - 2. IP mutation is performed with **high unpredictability and rate** to maximize the distortion of attacker’ s knowledge and increase deterrence of attack planning.
- Principles: each host associate with an unused address range, in every mutation interval, picking up a virtual IP to associate with this host.
- contributions:
  - allocating unused ranges to hosts
  - mutate within allocated ranges
- Solving method – model as a constraint satisfaction problem, solved by the tool called Yices, a kind of SMT solvers

Random Host Mutation

- Details about the algorithm

Algorithm 1 NOX controller algorithm

determine unused ranges.
determine range-to-subnet assignments
for all packets p from OF-Switches **do**
    if p is a Type-A DNS response for host h_i **then**
        set DNS *addr* to current $vIP(h_i)$, $TTL \simeq 0$
    else if p is a TCP-SYN or UDP from h_i to h_j **then**
        if $p.src$ is internal **then**
            install *in* flow in src OF-switch with
                action $srcIP(p) := vIP(h_i)$
            install *out* flow in src OF-switch with
                action $dstIP(p) := rIP(h_i)$
        end if
        if $p.dst$ is rIP **then**
            if h_i access to h_j is authorized **then**
                install *in* and *out* flows in dest OF-switch
            end if
        else[$p.dst$ is vIP]
            install *in* flow in dest OF-switch with
                action $dstIP(p) := rIP(h_j)$
            install *out* flow in dest OF-switch with
                action $srcIP(p) := vIP(h_j)$
        end if
    end if
    for all mutation of each host h_i **do**
        set $vIP(h_i)$ to a new vIP
    end for
end for

Jafarian, Jafar Haadi, Ehab Al-Shaer, and Qi Duan. "Openflow random host mutation: transparent moving target defense using software defined networking." *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012.

Random Host Mutation

- Process of communication

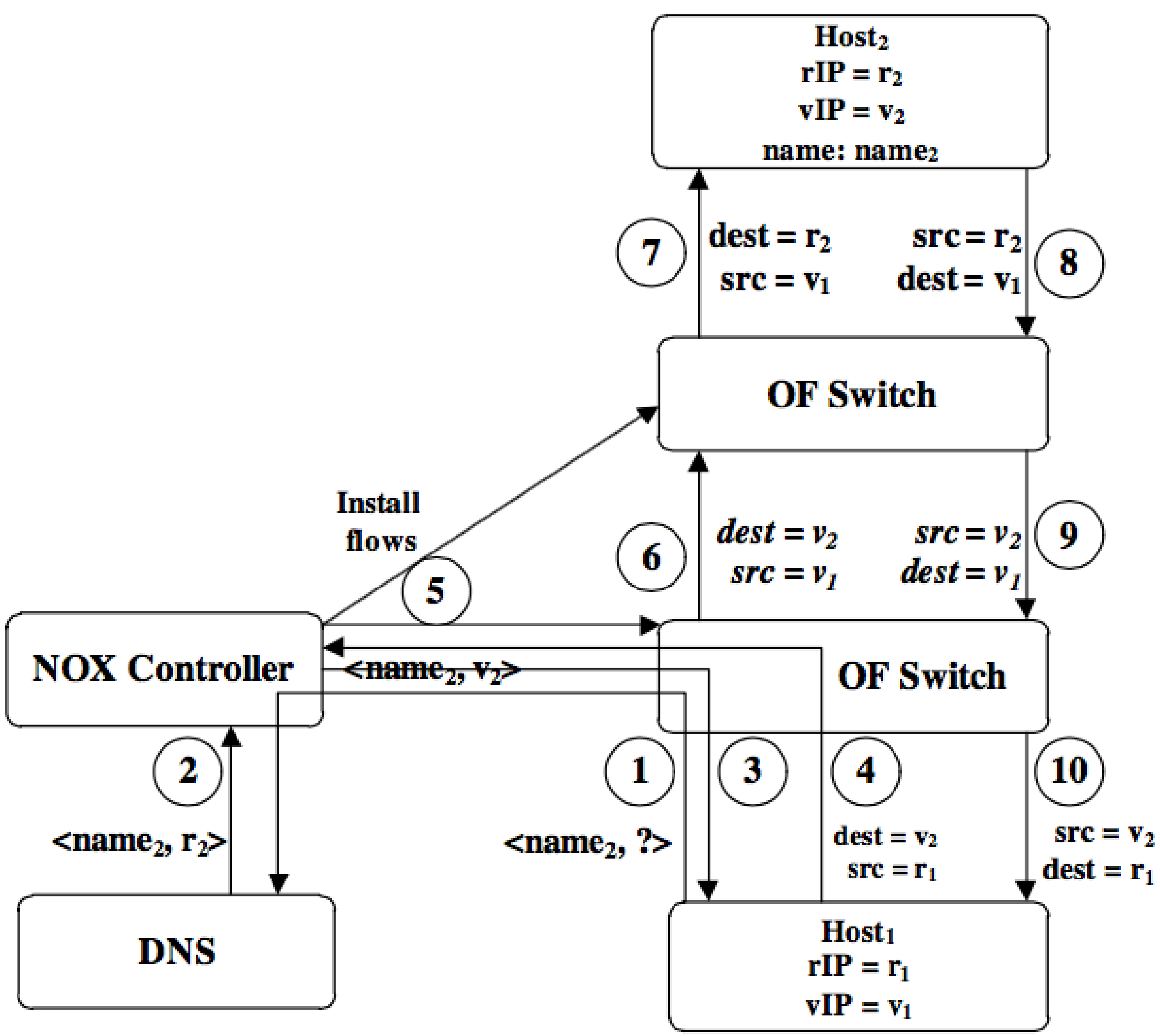
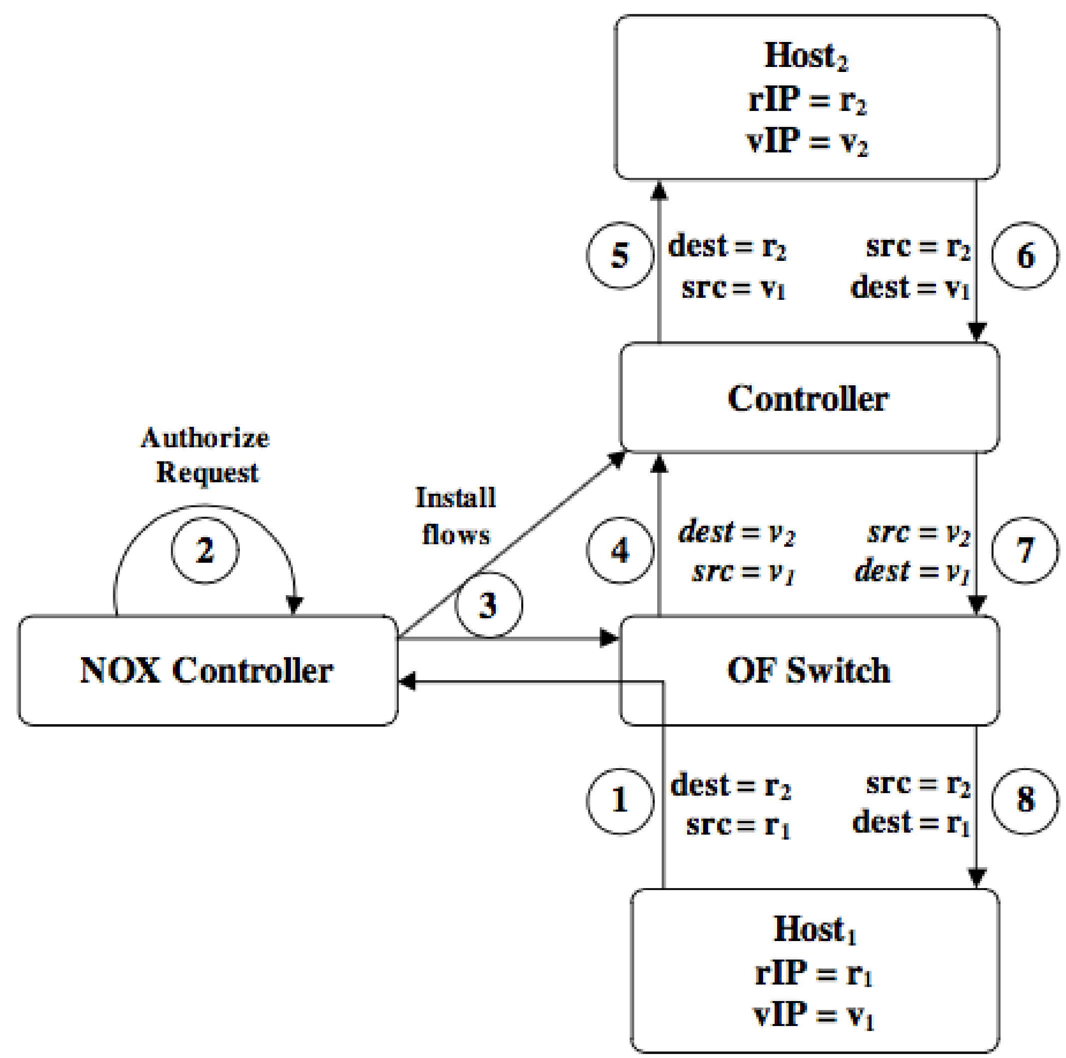


Figure 2: Communication via name



ıre 3: Communication via rIP address

Jafarian, Jafar Haadi, Ehab Al-Shaer, and Qi Duan. "Openflow random host mutation: transparent moving target defense using software defined networking." *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012.

Random Route Mutation

- View the whole network as a directed graph G=(V, E), V represents for hosts, E represents for links
- Assume there is a flow, which source is S, destination is D, the goal of RRM
  - find a route between S and D that satisfies the following constraints for every interval of the flow duration:
    - **Capacity constraint**: the new route should not include those nodes or links that do not have the bandwidth requirement for the flow;
    - **Overlap constraint**: to increase unpredictability and achieve good load balancing, the new route should avoid those intermediate nodes that appear in recently used routes;
    - **QoS constraint**: the mutated routes should maintain the required quality, such as bounded delays or number of hops.
  - If there are more than one satisfying routes for the flow, the RRM procedure should choose one satisfying route randomly.

Duan, Qi, Ehab Al-Shaer, and Haadi Jafarian. "Efficient random route mutation considering flow and network constraints." *Communications and Network Security (CNS), 2013 IEEE Conference on*. IEEE, 2013.

Random Route Mutation

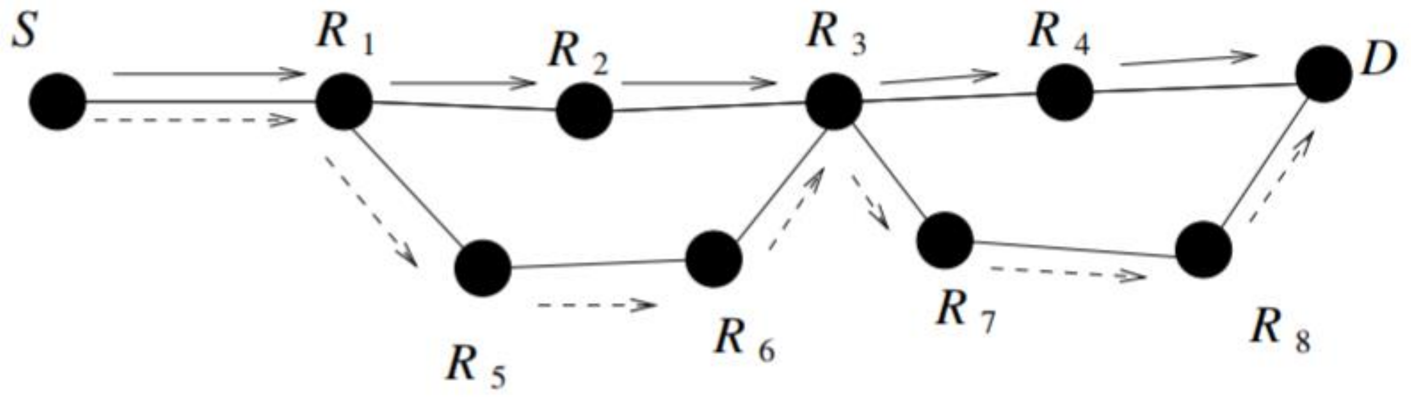


Fig. 1. An example of route change

- Overlay networks were proposed to improve the reliability of the Internet and to facilitate performance sensitive services, such as mission critical networks, high performance cloud computing etc., that are difficult or impossible to deploy natively on the Internet.
- We need to find a mapping between overlay nodes and substrate nodes with the following constraints:
  - **Unique Mapping Constraint**: every overlay node should be mapped to exactly one substrate node, and two distinct overlay nodes should not be mapped to the same substrate node.
  - **Disjoint Route Constraint**: the placement must guarantee that there are enough number of disjoint routes between the corresponding substrate source and destination of the overlay flows.
  - **Unpredictability Constraint**: there should be enough difference between the old and new placement.

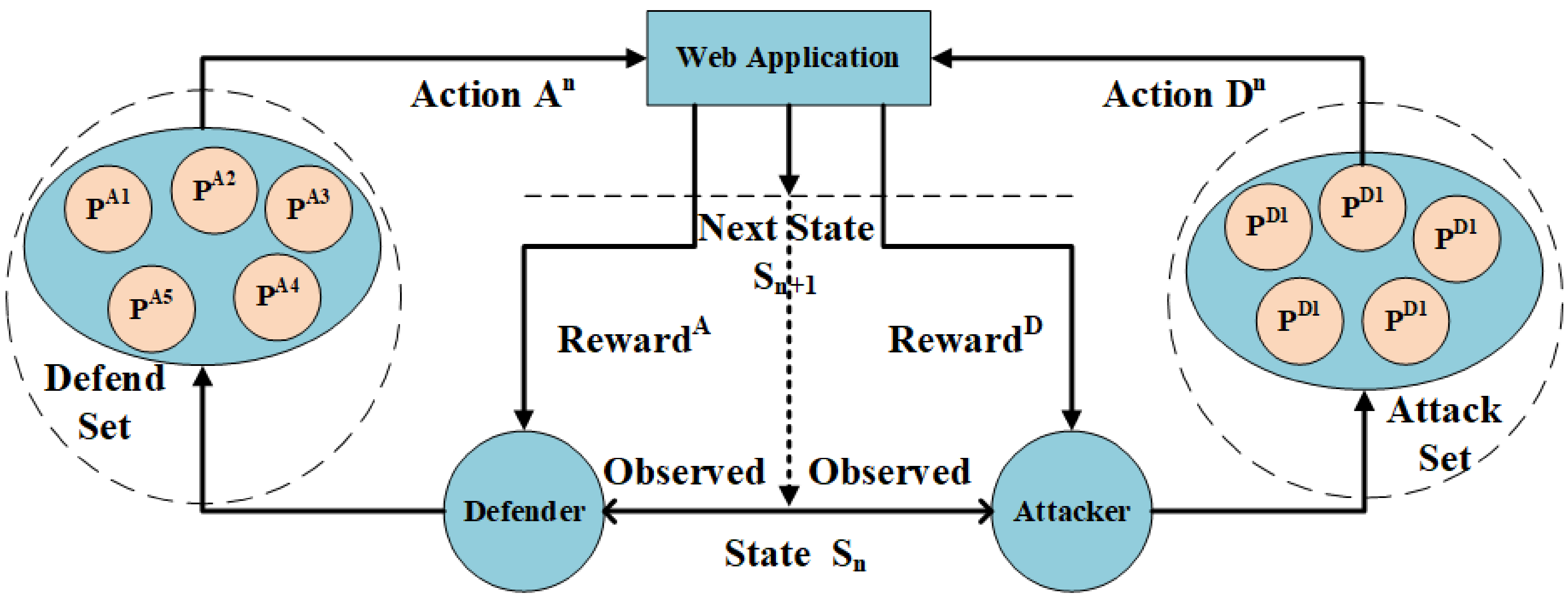
RHM & RRM

- Using these ways, the information that attackers achieved are expired.

- These two methods are used to defense against eavesdropping, worms, DoS, or other hitlist attacks on the specific node or link.

			Attack																	
			Scan	**Reconnaissance**	**Directory Traversal Attack**	**DDoS**	**CSRF**	**Redirect attack**	**XSS**	**SQL Injection**	**CRLF Injection**	**OS injection**	**Trojan**	**File Upload**	**Worm**	**Rootkit**	**Buffer Overflow**	**WEB Bot**	**Interfere/ Jamming**	**Side Channel**
Defense	Network	HTML Elements	L1	L1	L1	L1	L1	L1	L1	M1	L1	L1	L1	L1	L1	L1	L1	H1	L1	L1
		Token	L1	L1	L1	L1	H2	H2	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1
		IP/IPv6	H2	H2	L1	H2	L1	L1	L1	L1	L1	L1	L1	L1	H2	L1	L1	L1	L1	L1
		Port	H2	H2	L1	H1	L1	L1	L1	L1	L1	L1	L1	L1	H2	L1	L1	L1	L1	L1
		Route/ Topology	H2	H2	L1	L2	L1	L1	L1	L1	L1	L1	L1	L1	M2	L1	L1	L1	L1	L1
		TCP/ICMP	H1	H1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1
		MAC	H1	H1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	H2	L1
		Proxy	H2	H2	L1	H1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1
	Host	WEB Server	H2	M2	L1	L1	L1	L1	L1	L1	L1	H1	L1	L1	M1	L1	H1	L1	L1	M1
		Operation System	H2	H2	L1	L1	L1	L1	L1	L1	L1	H2	H2	H1	H2	H2	H2	L1	L1	M1
		Database	H2	M2	L1	L1	L1	L1	L1	H2	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1
		Programming language	M2	L1	L1	L1	L1	L1	L1	H2	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1
		File Information	M1	L1	H2	L1	L1	L1	L1	M1	L1	L1	H1	H2	H1	L1	L1	L1	L1	L1
		Database Information	M1	L1	L1	L1	L1	L1	L1	H1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1
		Software Diversity	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	H2	L1	L1	H1
		VM	L1	L1	L1	L1	L1	L1	L1	L1	L1	H1	H2	L1	L1	H2	H2	L1	L1	H2
		Instruction Set	L1	L1	L1	L1	L1	L1	H2	H2	H2	H2	L1	L1	M1	L1	H2	L1	L1	L1
		Service Version	H2	H2	L1	L1	L1	L1	M1	M1	L1	L1	L1	L1	M1	L1	L1	L1	L1	L1
		Address Space Layout Randomization	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	L1	H1	L1	H2	L1	L1	L1

MTD策略选择--博弈论



- 在系统状态j中，攻击者选择的策略为A_n，防御者选择的策略为 D_m
- $\eta_j(A_n, D_m)$ 表示D_m对 A_n的防御效率（$0 \leq \eta \leq 1$），攻击者造成的危害为 $DM(A_n)$。
- 攻击者回报函数：$RA\left(S_j, A_n, D_m\right) = DM(A_n) * (1 - \eta_j(A_n, D_m))$
- 防御者回报函数：$RD\left(S_j, A_n, D_m\right) = DM(A_n) * \eta_j(A_n, D_m)$

MTD策略选择--博弈论

- 防御者最终选择的策略： $\pi_*^D = \arg NASH^D(RD(S_j, A_*, D_*), RA(S_j, A_*, D_*))$
- 攻击者最终选择的策略： $\pi_*^A = \arg NASH^A(RD(S_j, A_*, D_*), RA(S_j, A_*, D_*))$

(π_*^D, π_*^A) 是达到纳什均衡的攻防策略组合。π_*^D是防御者对攻击者策略的最好回应， π_*^A则是攻击者对防御者策略的最好回应。

$$RD(\pi_*^D, \pi_*^A) \geq RD(\pi^D, \pi_*^A)$$

$$RA(\pi_*^D, \pi_*^A) \geq RA(\pi_*^D, \pi^A)$$

混合策略纳什均衡求解

纳什均衡：在网络攻防博弈中，攻防双方只有采取纳什均衡策略才能获得各自的最大化收益。

问题和讨论