SOFTWARE DEFINED NETWORK

PROF. (DR.) JYOTSNA KUMAR MANDAL

WHAT IS SOFTWARE DEFINED NETWORK (SDN) ?

- ***What is SDN?***
- Restructuring the current network infrastructure for improved network management.
- It is not a new technology-rather reshaping the current network.
- Control and data planes are decoupled from the traditional forwarding devices.

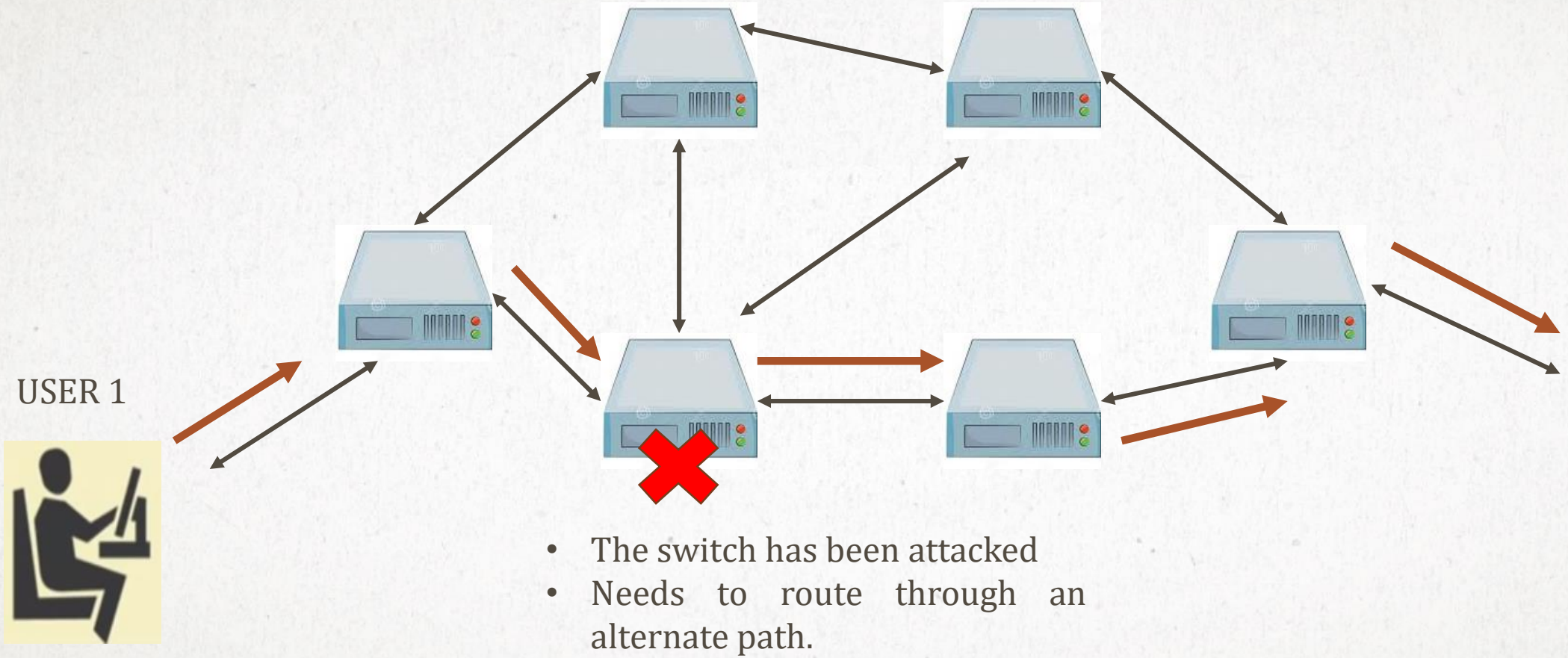
DEFINITION OF SOFTWARE DEFINED NETWORK (SDN)

- Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications.
- This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. The OpenFlow protocol is a foundational element for building SDN solutions.

CONTROL AND DATA PLANE

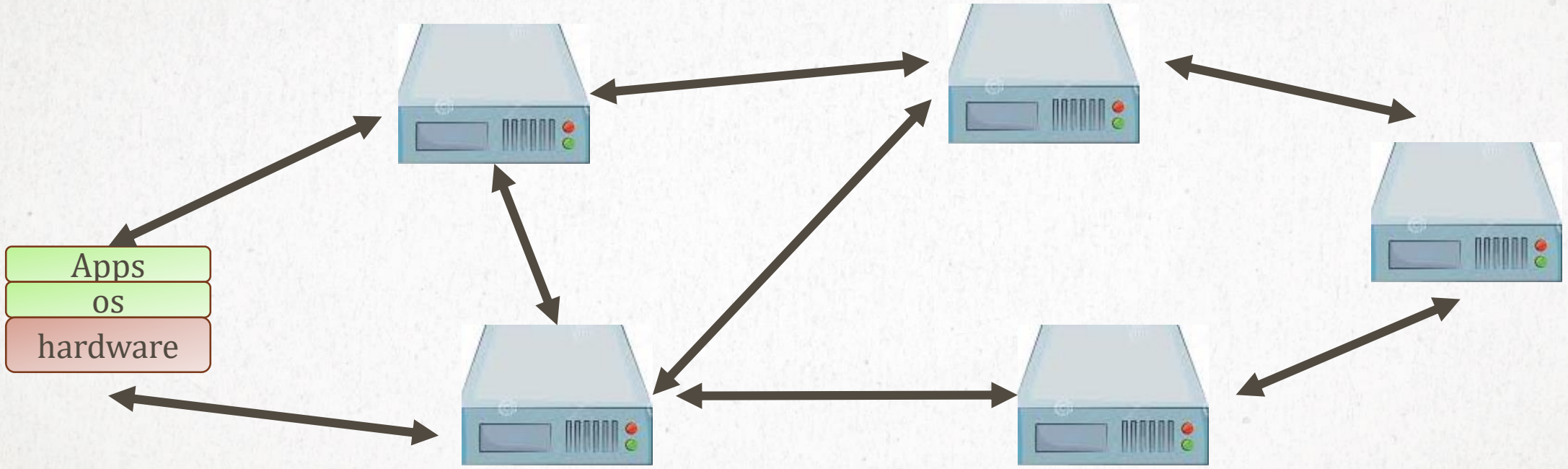
- CONTROL PLANE
- The module which takes all decisions, basically an instructor.
- The routing algorithm.
- DATA PLANE
- The module which carries out the tasks given by the control plane.
- Forwarding of Packets.

OVERVIEW OF CURRENT NETWORK

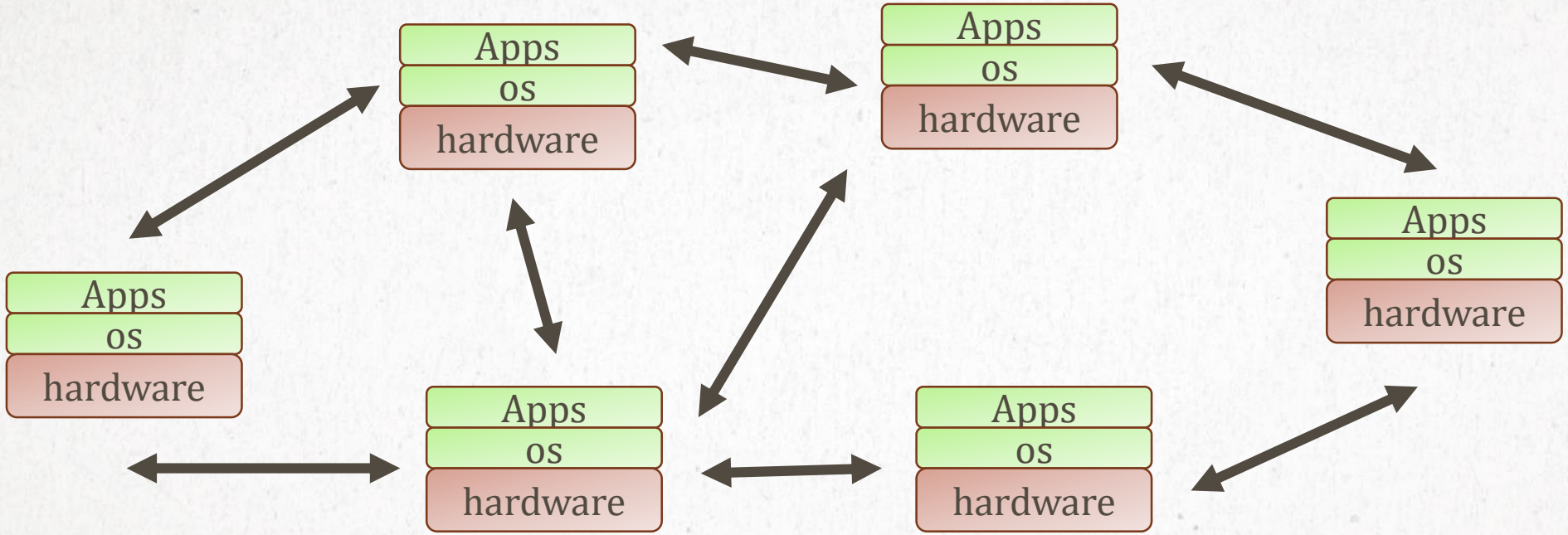


- The switch has been attacked
- Needs to route through an alternate path.
- Present: No centralized control

LIMITATIONS IN CURRENT NETWORK



LIMITATIONS IN CURRENT NETWORK



- Switches have forward traffic in a distributed manner
- They do not have a global view of the network

LIMITATIONS IN CURRENT NETWORK

- Traditional networking devices are proprietary
- ❑ Vendor-specific architecture of switches limits dynamic configuration according to application-specific requirements.
- ❑ Switches are required to configure according to installed operating system (OS).
- ❑ Centralized control is not feasible in traditional network.

LIMITATIONS IN CURRENT NETWORK

Thousands lines of code

Routing, mobility management, etc

Cost-expensive

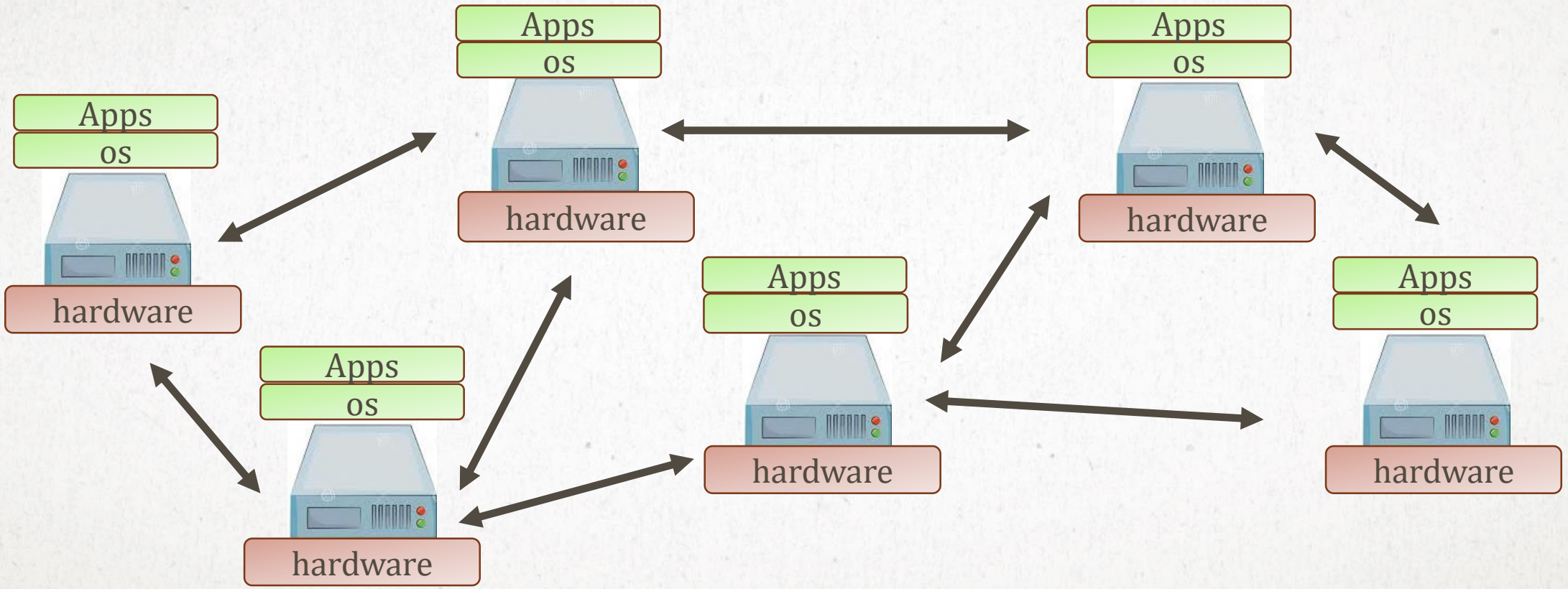
Millions of gates

App App App

Operating System

Specialized packet forwarding hardware

CURRENT NETWORK TO SDN



WHAT DOES SEPARATING CONTROL AND DATA PLANE MEAN?

But what if they are separate?

- Vendors only provide the hardware(data plane)
- We decide the control plane by writing custom logic – the software.

Advantages:

- Features are no longer limited to what the vendor provides.
- Community Development.
- Longer life of products.

HOW DOES SDN WORK?

- *Compared to traditional networks, a software defined network has 2 types of devices*
- Controller
- Switches
- *The switches in SDN are blind*
- No built-in features
- Need to be instructed by the controller

SDN CONTROLLER

- "Brains" of the network
- All policies, routing logic are placed in the controller.
- "Teaches" the switches what to do with an unknown packet

Let us now see a scenario where a packet wants to go from H1 to H3 in an SDN environment.

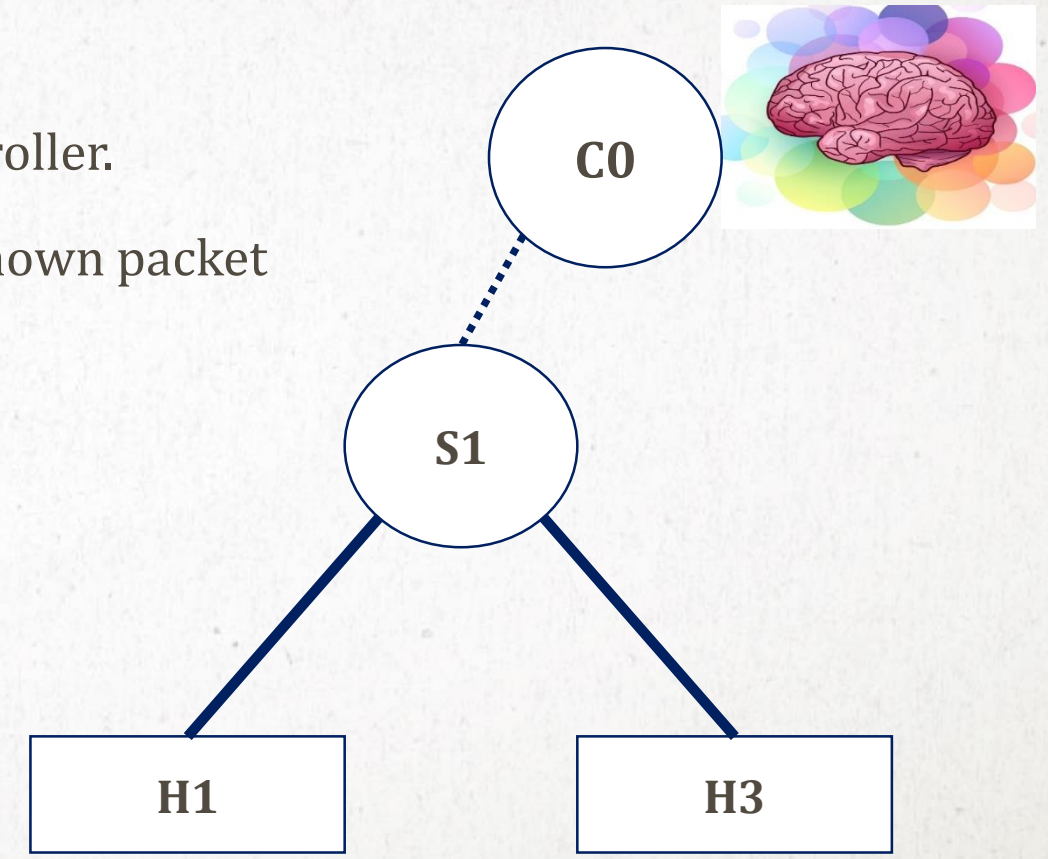


TABLE MISS AT S1

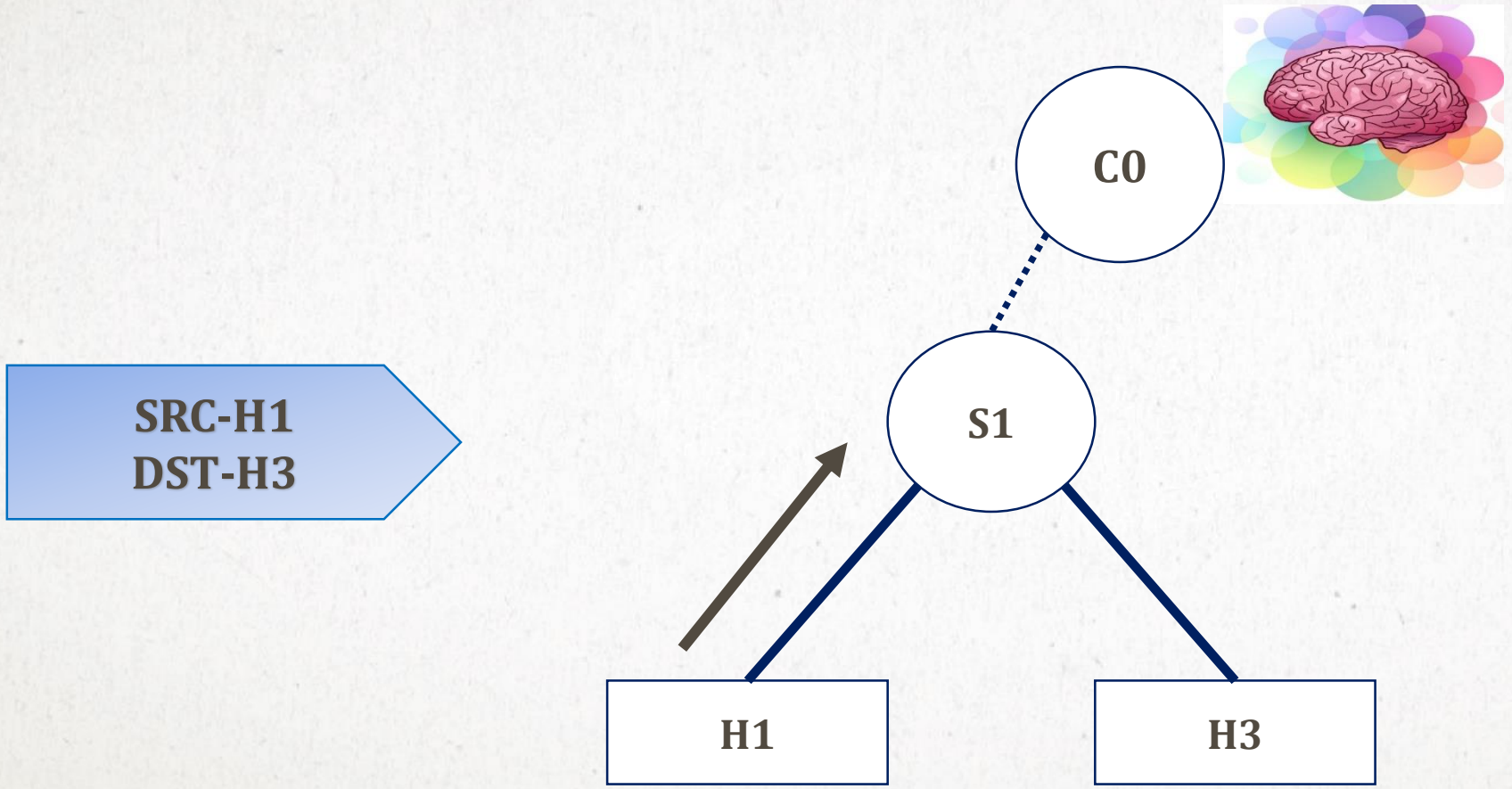
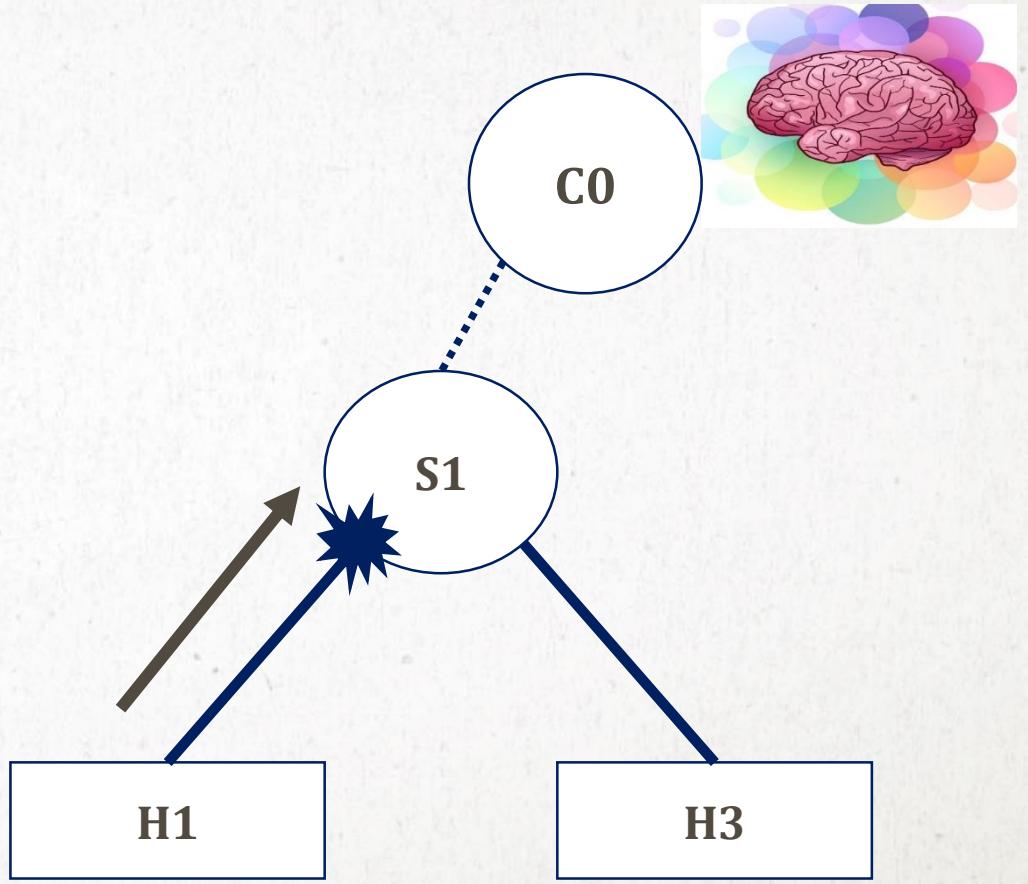


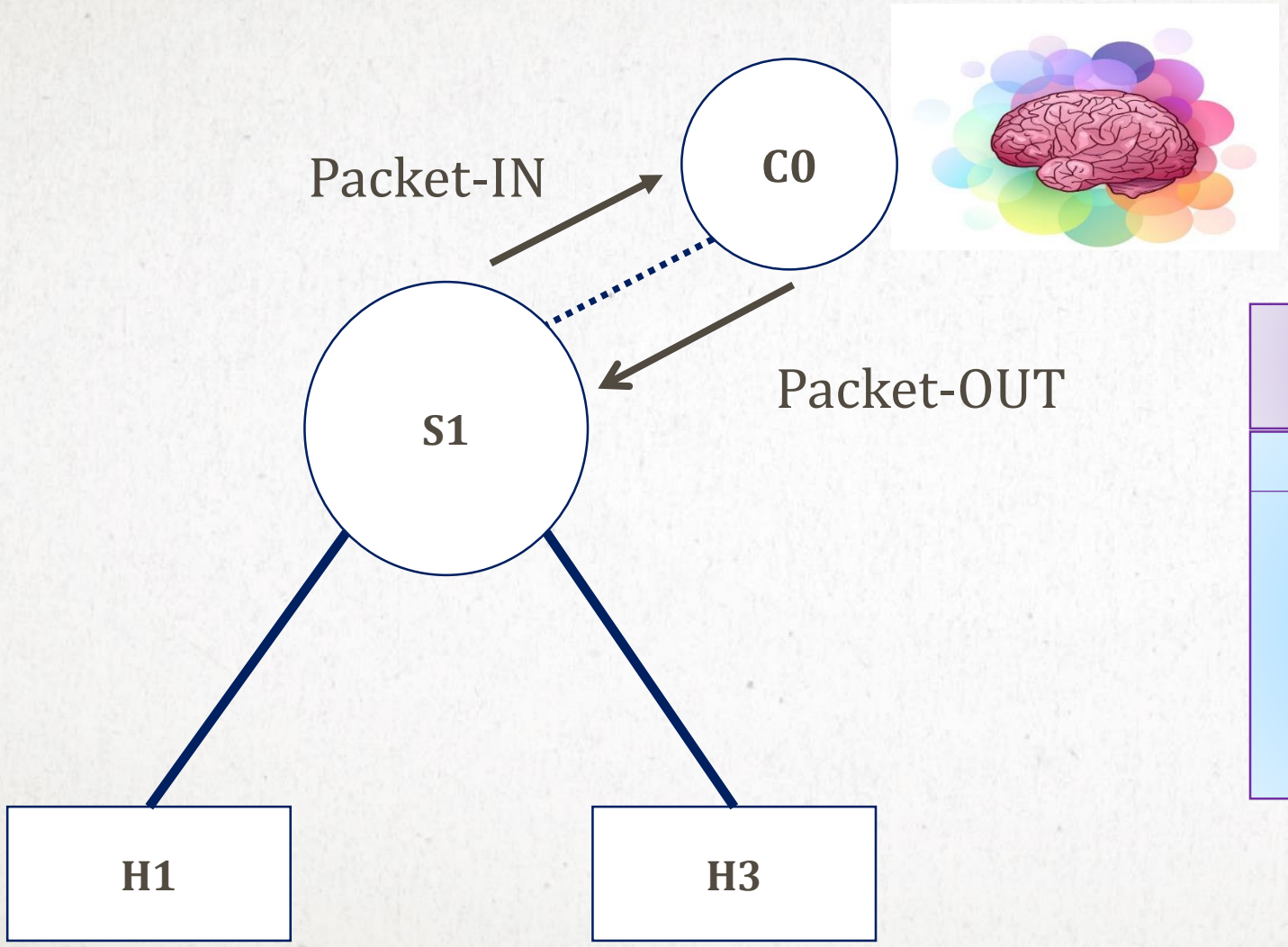
TABLE MISS AT S1

No rule is there for H1->H3 at the beginning.

Table Miss
?



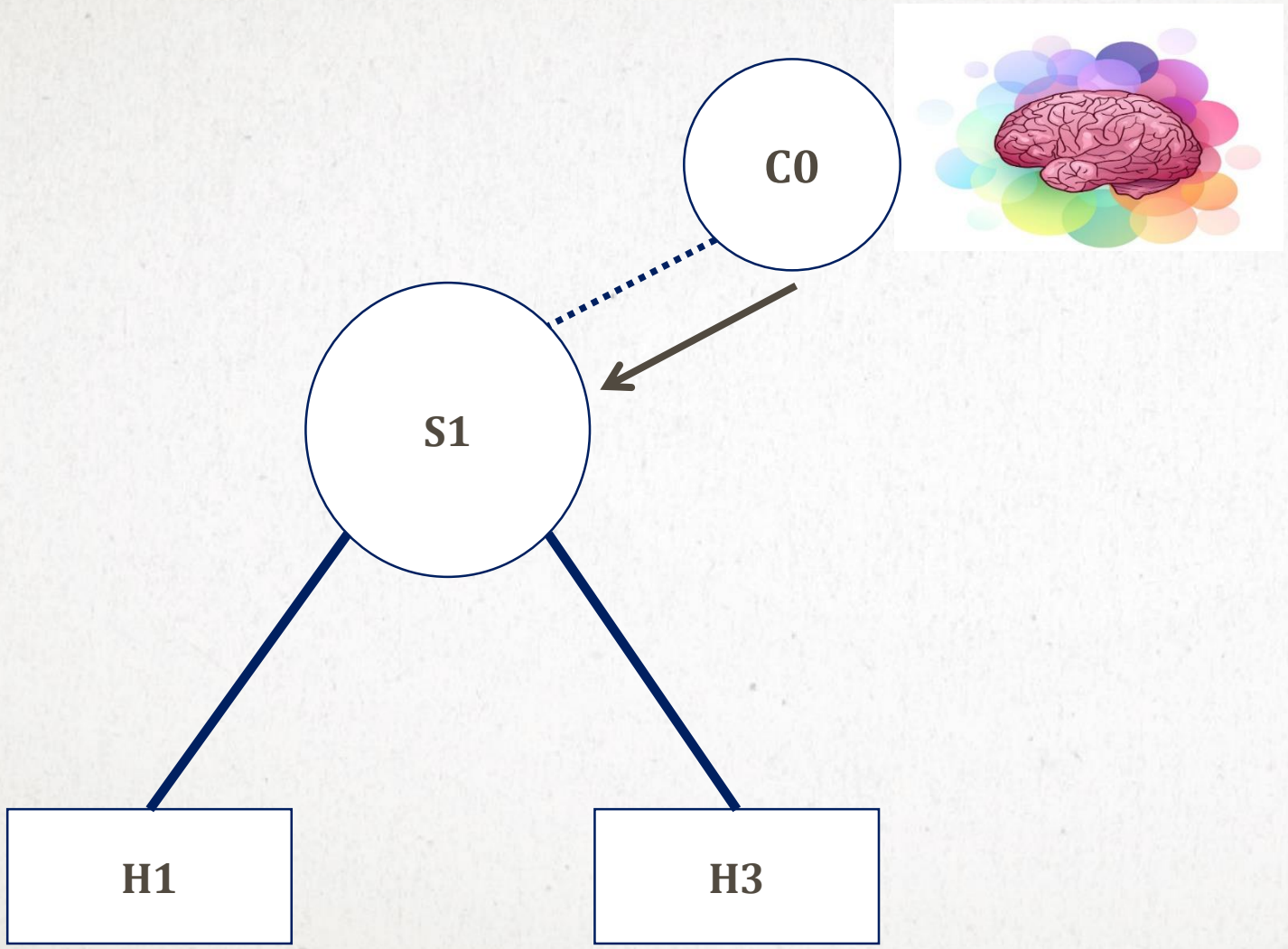
PACKET-IN CONTROLLER



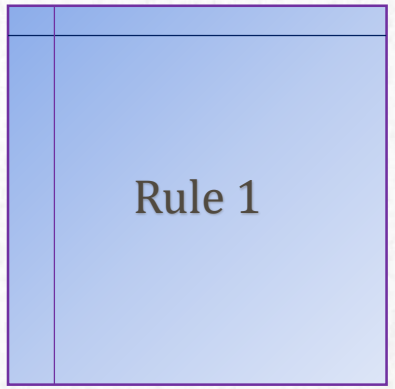
The controller generates the rule based on a software program (The routing logic)

The packet is buffered

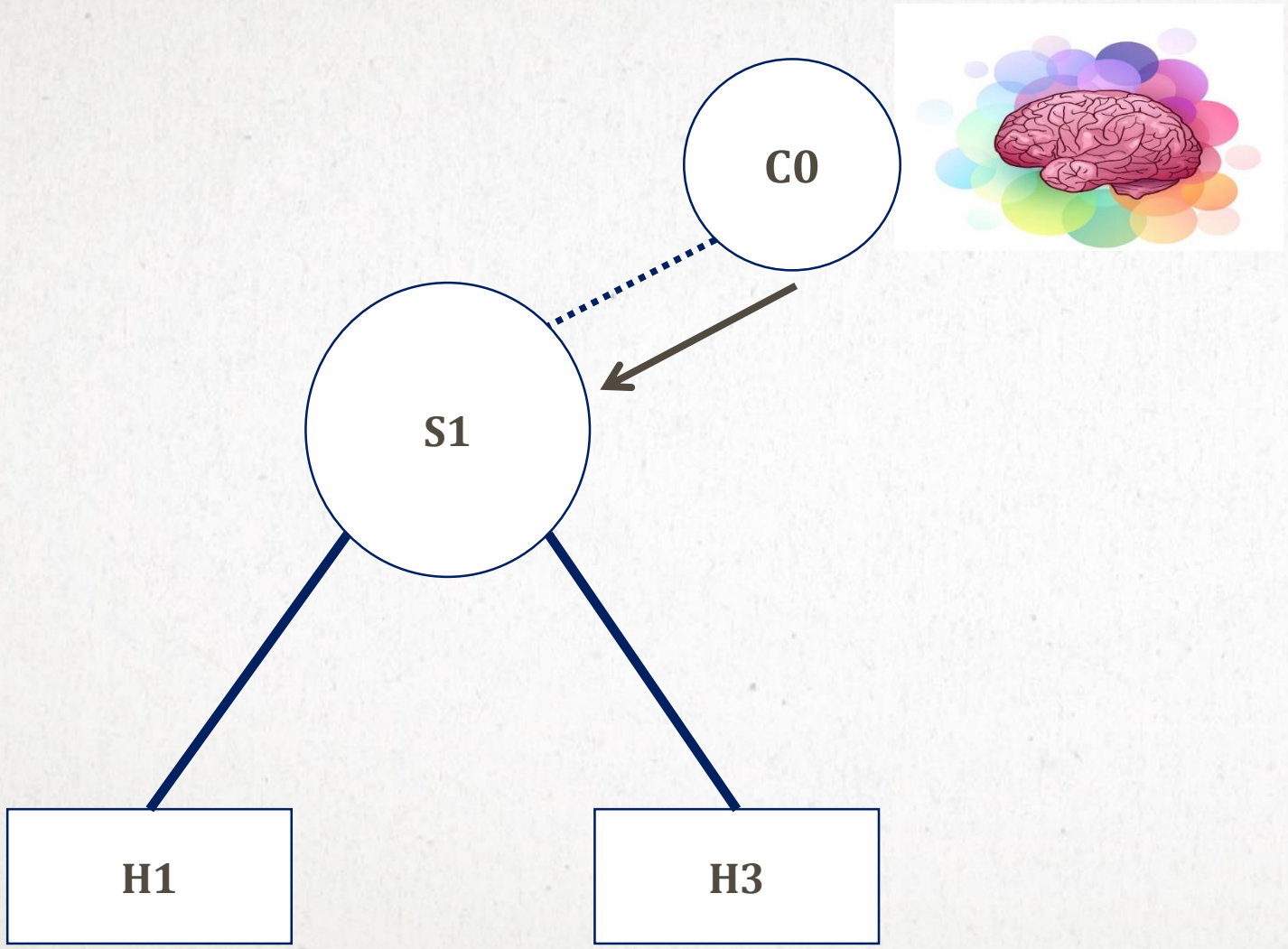
FLOW RULE SET AT S1



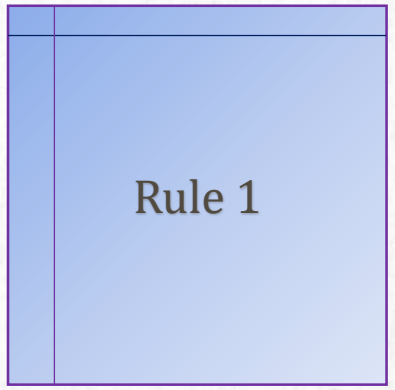
The rule is installed at the TCAM hardware of the switch.



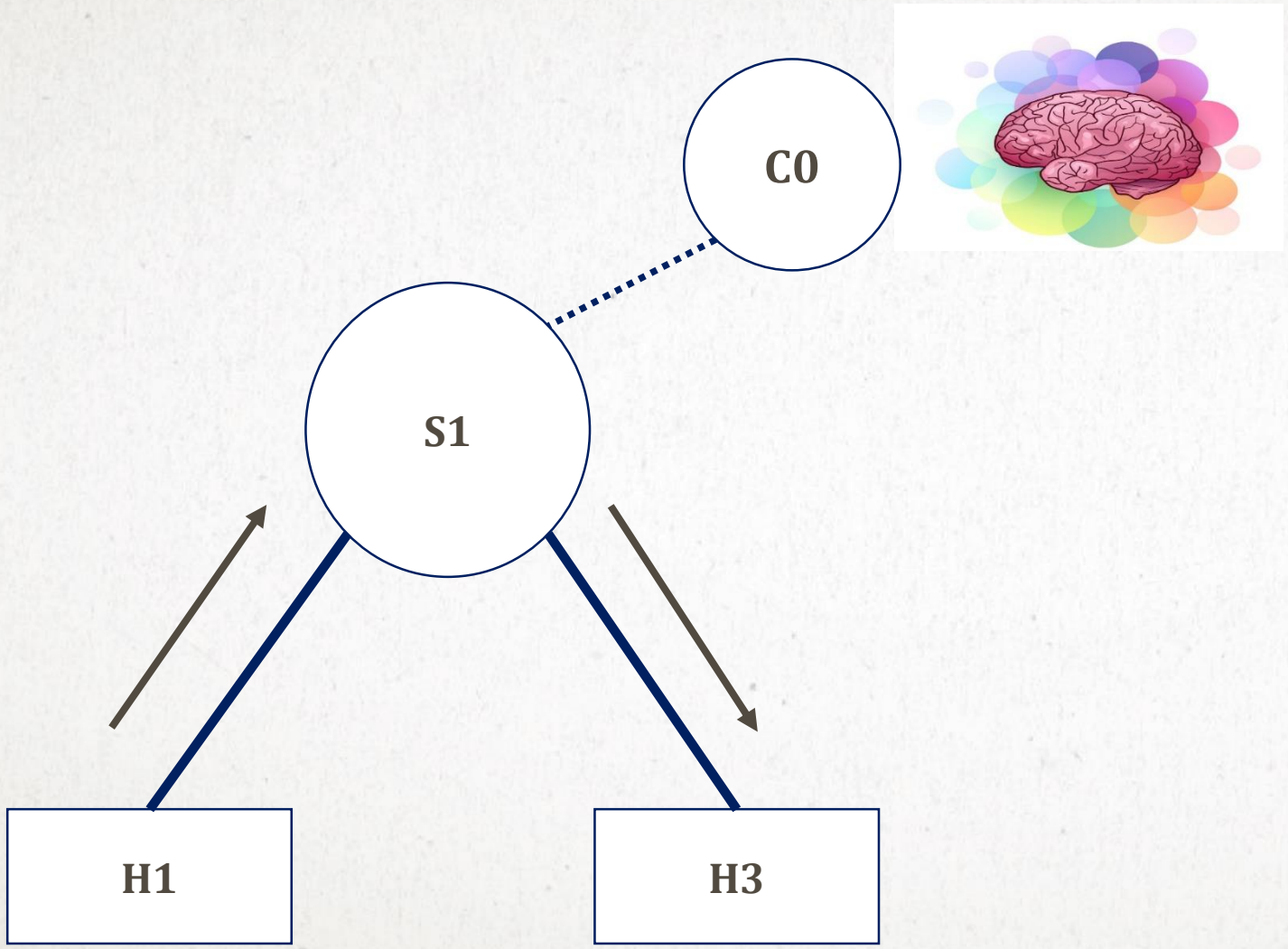
BUFFERED PACKET FORWARDED TO H3



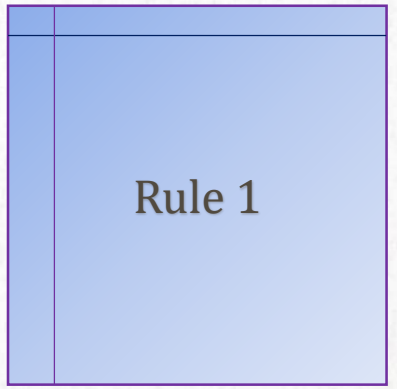
The buffered packet is forwarded based on that rule.



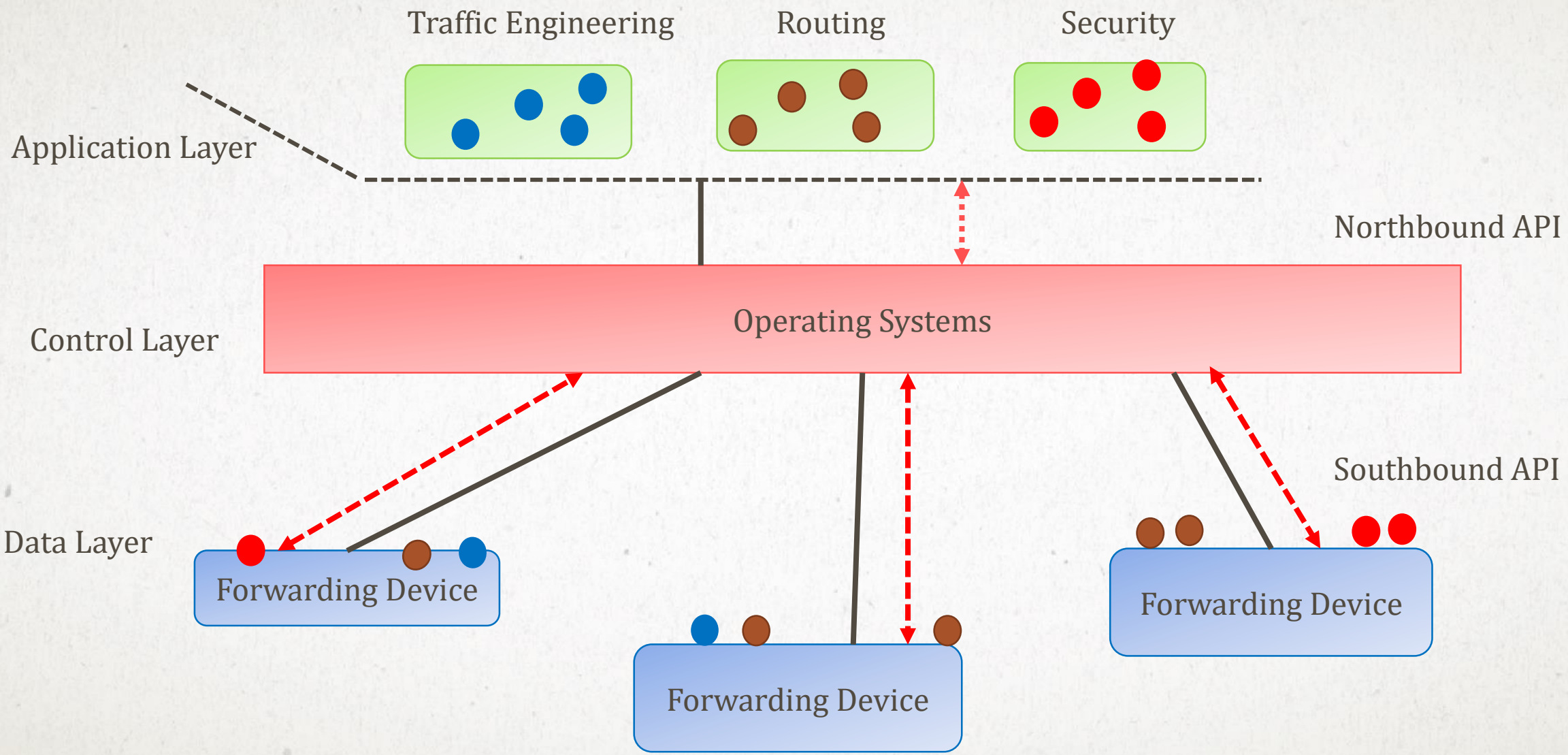
ALL FUTURE PACKETS FROM H1->H3



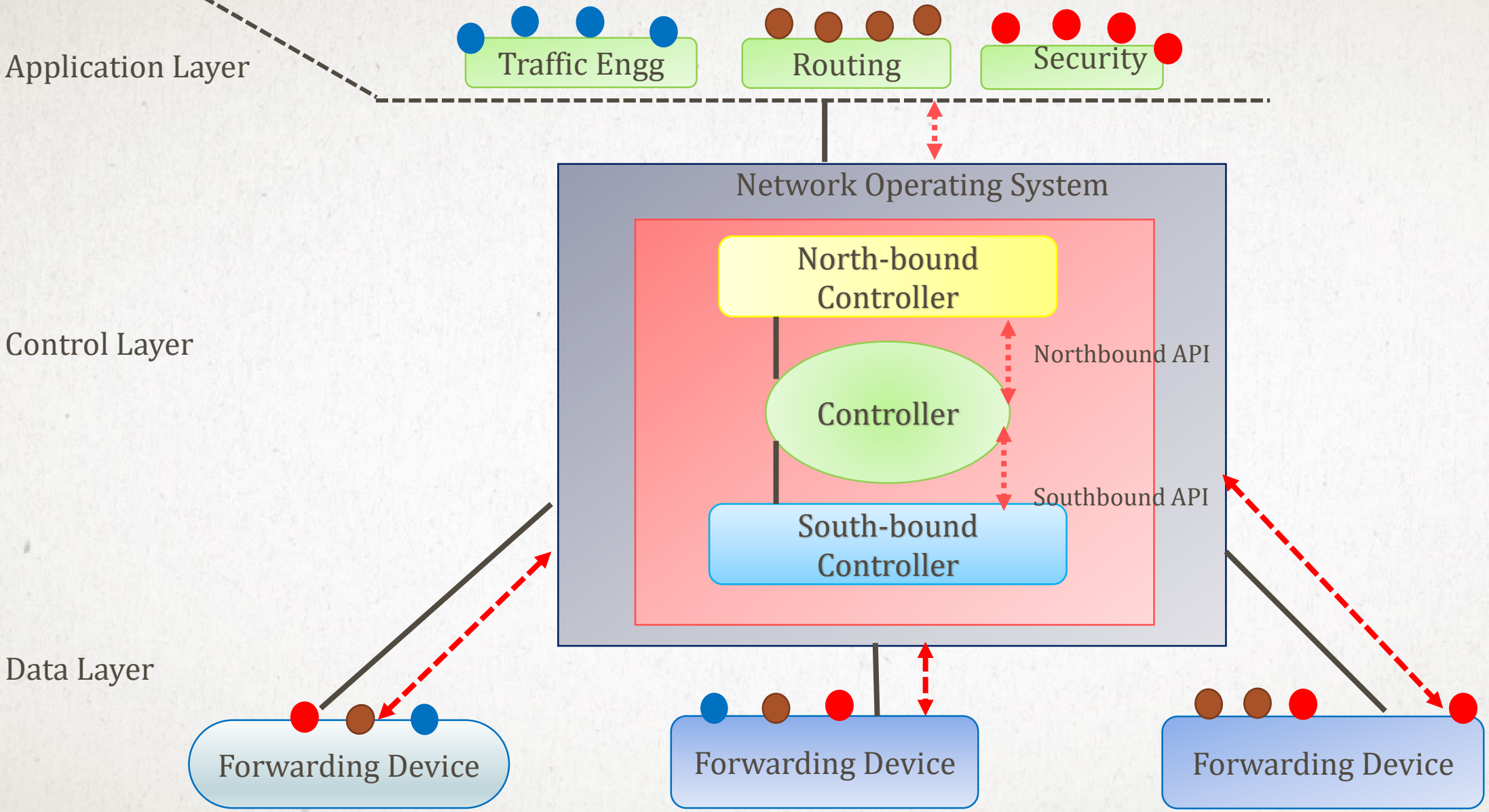
All the future packets from h1->h2 follow the same rule.



SDN ARCHITECTURE



SDN Architecture



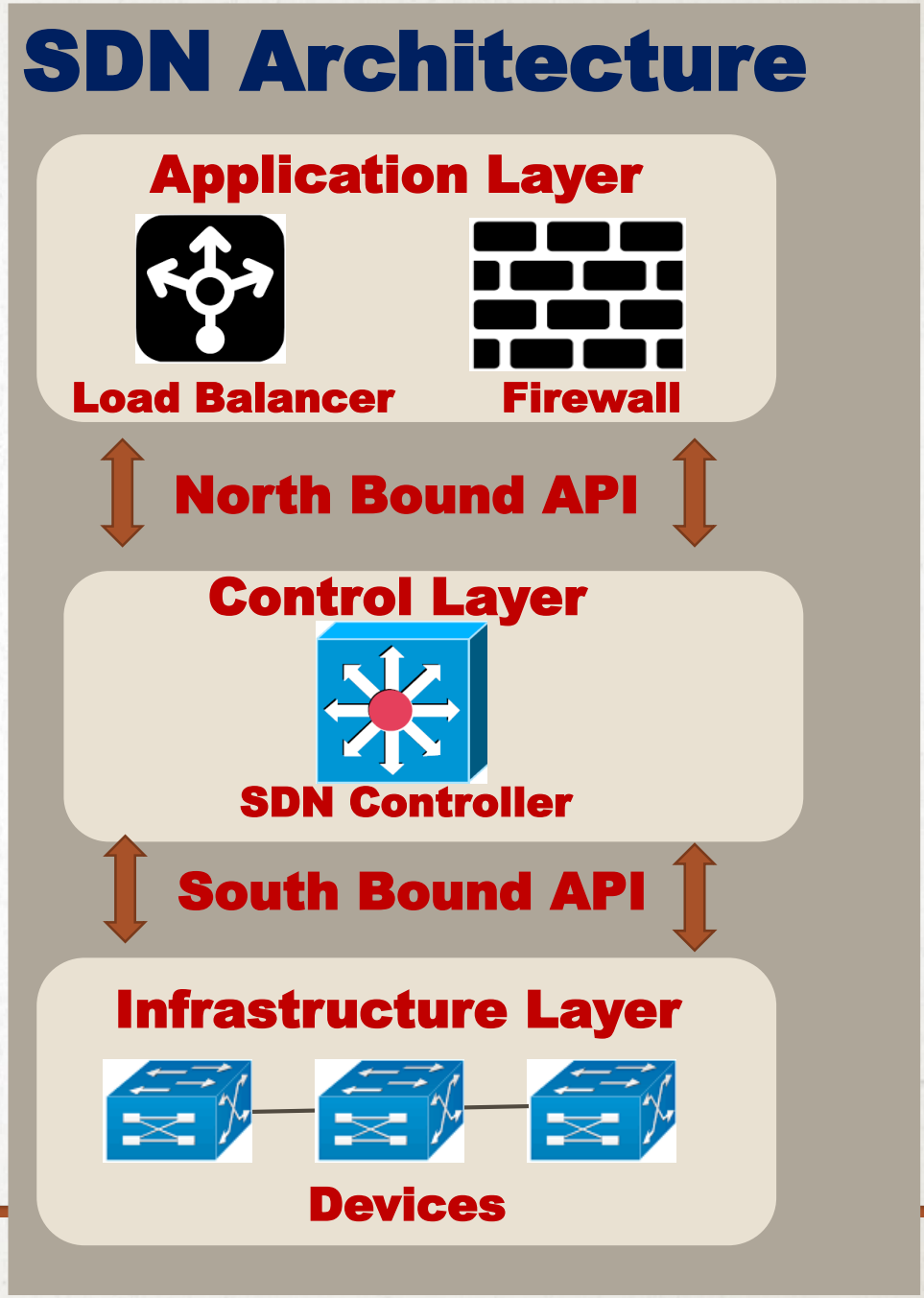
SDN ARCHITECTURE

A typical representation of SDN architecture comprises three layers:

- **The application layer,**
- **The control layer and**
- **The infrastructure layer.**

SDN ARCHITECTURE

These three layers communicate using respective northbound and southbound application programming interfaces (APIs)



HOW SDN WORKS?

- SDN encompasses several types of technologies which include functional separation, network virtualization and automation using programmable feature of SDN,
- Originally SDN focuses on the separation of data plane from network control plane,
- Data plane physically moves data from one network segment to another, &
- Network control plane makes decision how packet flows within the network.

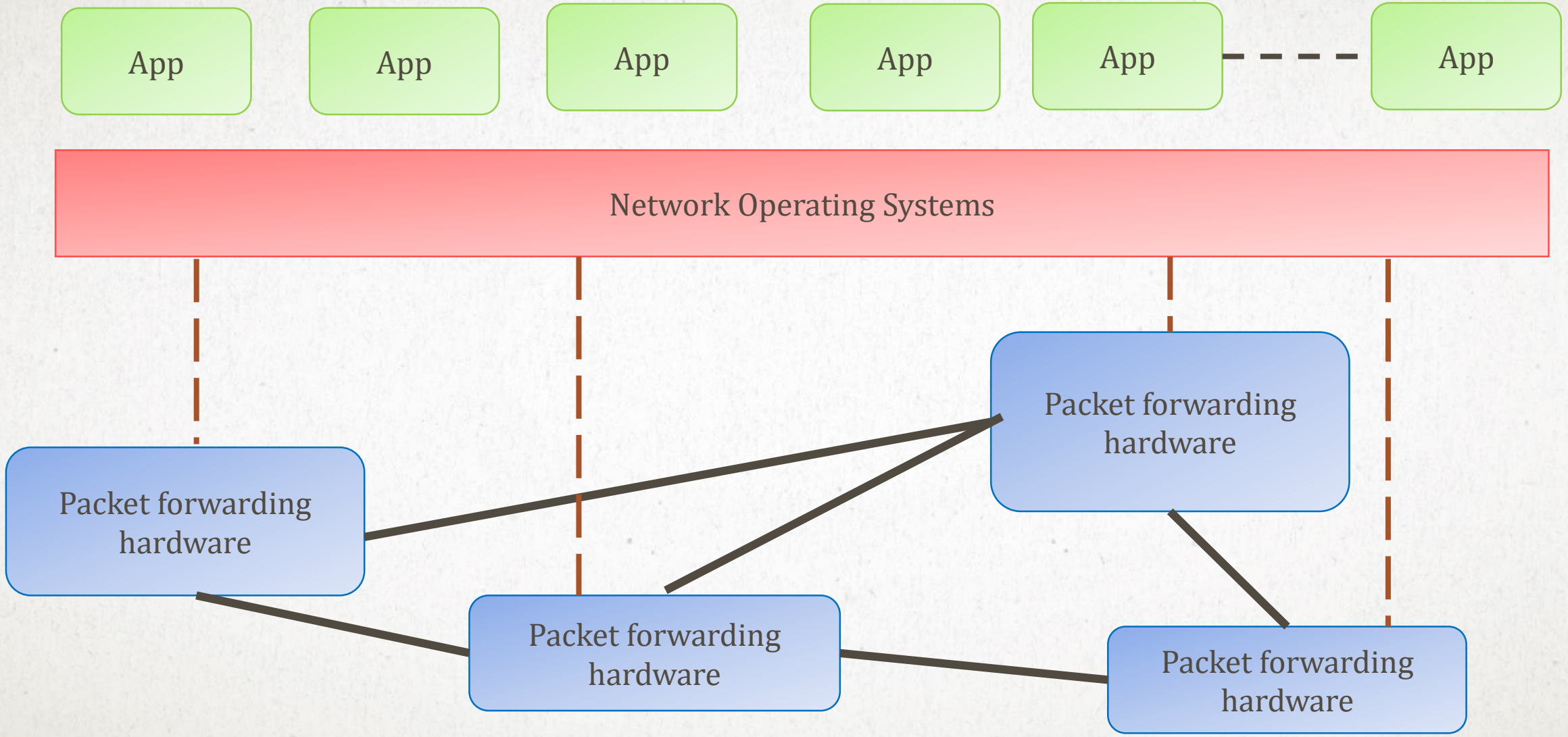
BENEFITS OF SDN:-

- Prioritizing & Deprioritizing specific network traffic
- High granularity of control
- End-to-end network management
- End-to-end network visibility
- Centralized controller based policy management
- High level of security by deploying security policies across the network
- Management of Suspicion Traffic and
- Virtualization and hardware resources and services

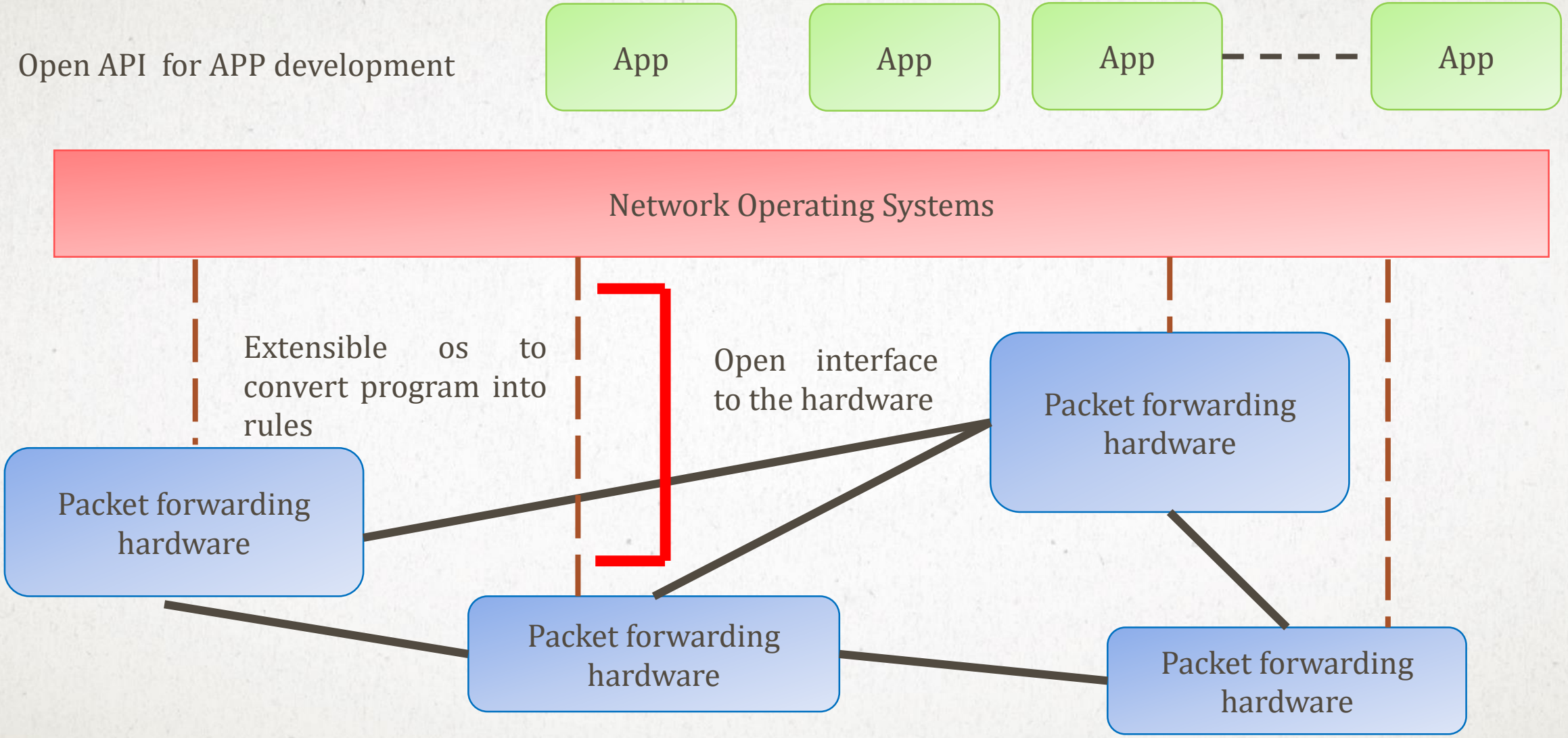
HOW DO WE IMPLEMENT SDN?

- OpenFlow (OF) is considered one of the first software-defined networking (SDN) standards
- It was originally defined to act as a communication protocol in SDN environment
-
- OpenFlow enables the SDN Controller to directly interact with the forwarding plane of network devices such as switches and routers

SDN BASED APPROACH TO OPEN THE INNOVATION



INTERFACE BETWEEN CONTROL AND DATA PLANE



WHAT IS OPEN FLOW?

- Protocol for controlling the forwarding behaviour of Ethernet switches in a SDN.
- Intially released by Clean Slate Program at Standford, spcifications now maintained by Open Networking Forum.

WHAT IS OPENFLOW ?

OpenFlow is a Layer 2 communications protocol that gives access to the forwarding plane of a network switch or router over the network

HOW OPENFLOW WORKS?

- OpenFlow enables network controllers to determine the path of network packets across a network of switches
- The controller is a separate entity and is distinct from the switches or infrastructure
- This helps in a sophisticated traffic management unlike access control list (ACL) or routing based mechanism
- OpenFlow allows single, open protocol to manage switches from multiple venders.

HOW OPENFLOW WORKS?

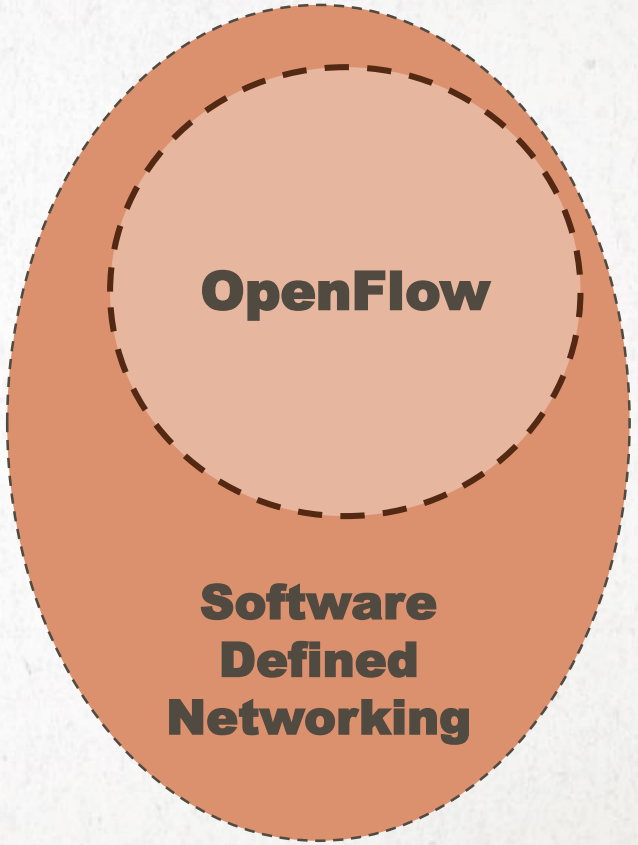
- OpenFlow allows remote administration of a layer 3 switch's packet forwarding tables
- This is achieved by adding, modifying and removing packet matching rules and actions.
- Routing decisions can be made periodically or ad hoc by the controller and translated into rules and actions with a configurable lifespan
- This is then deployed to a switch's flow table, leaving the actual forwarding of matched packets to the switch at wire speed for the duration of those rules
- Packets which are unmatched by the switch can be forwarded to the controller

OPENFLOW IMPLEMENTATION:

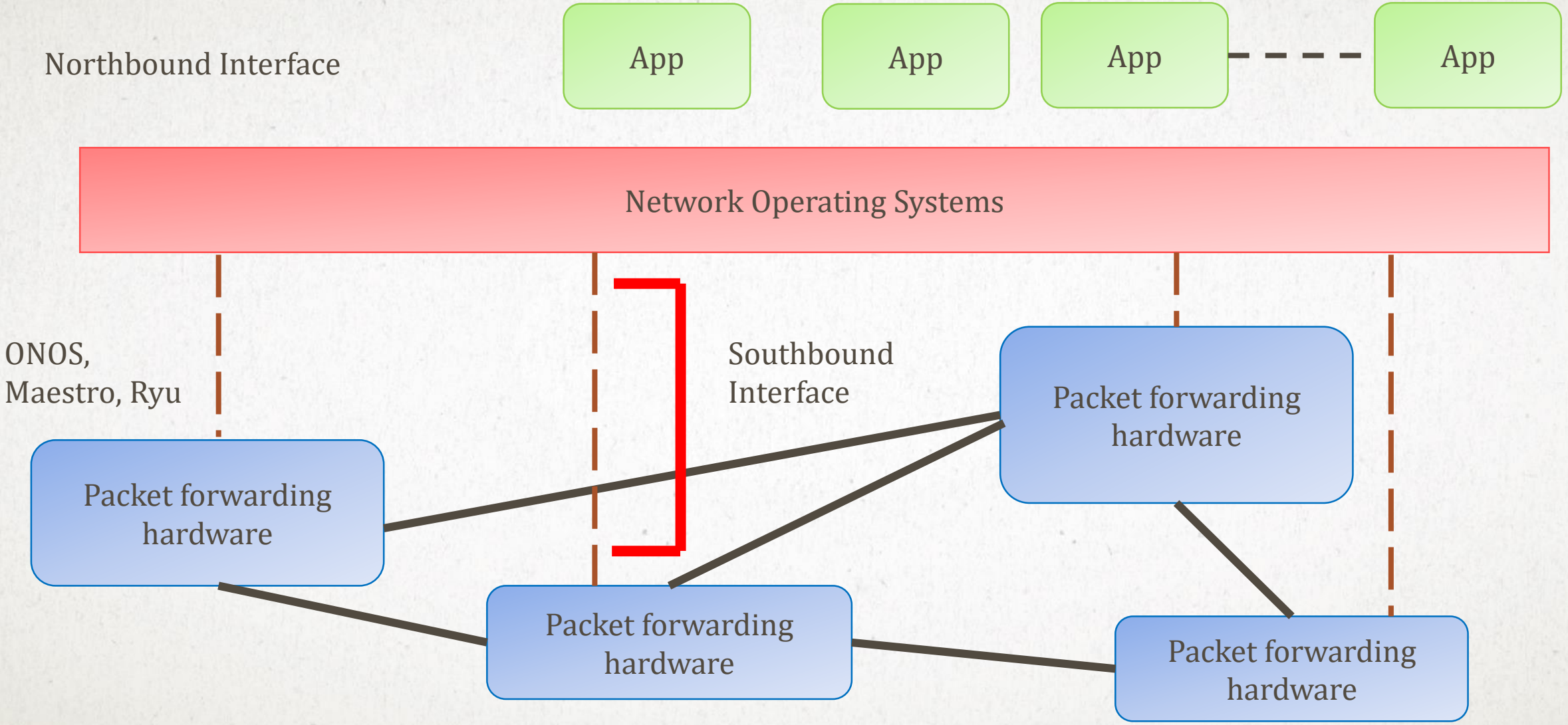
- OpenFlow protocol is layered on top of the Transmission Control Protocol (TCP)
- OpenFlow prescribes the use of Transport Layer Security (TLS)
- Controllers should listen on TCP port 6653 for switches that want to set up a connection
- OpenFlow is mainly used between the switch and controller on a secure channel

OPENFLOW & SDN

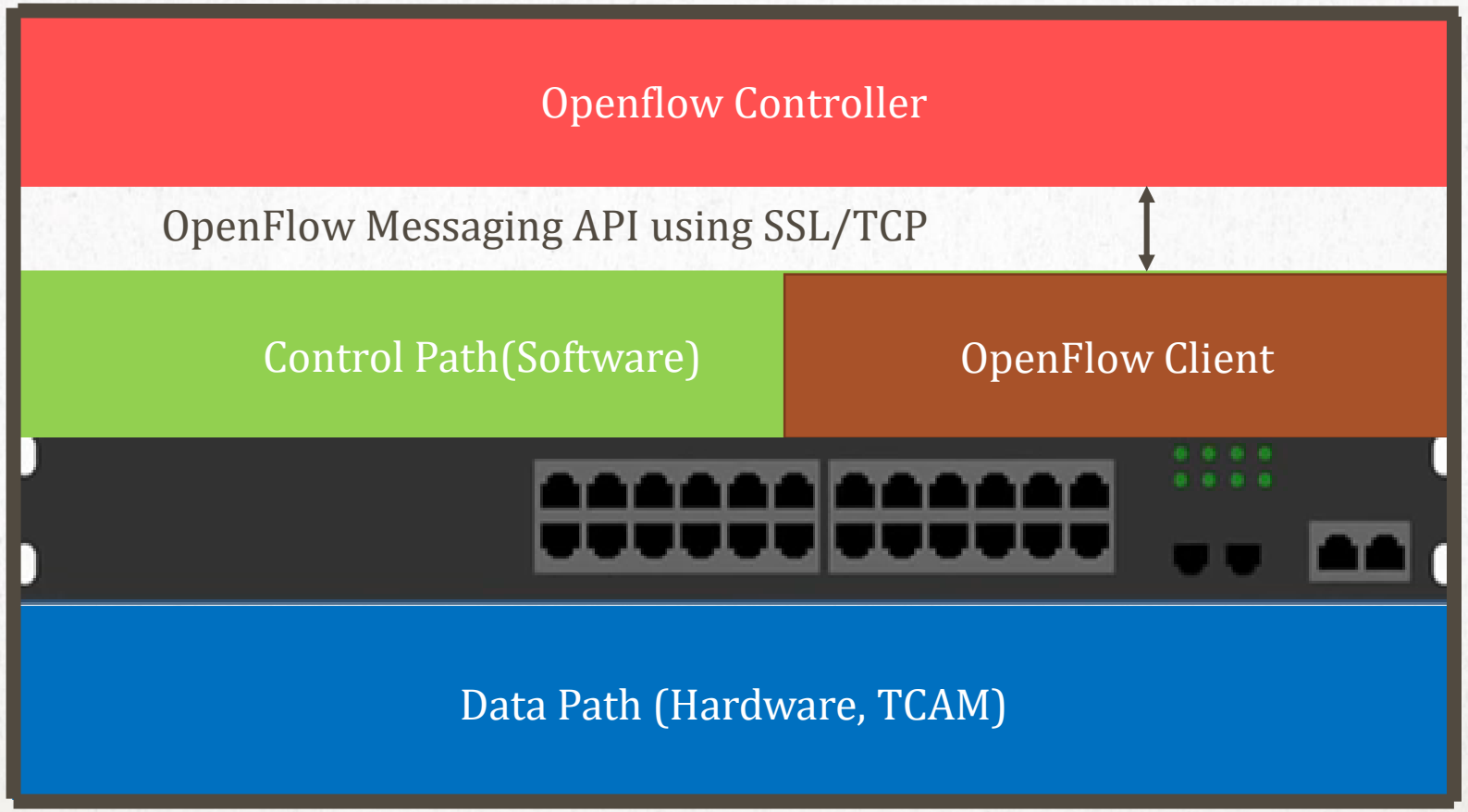
- OpenFlow is not SDN
- OpenFlow is one flavour or subset of SDN



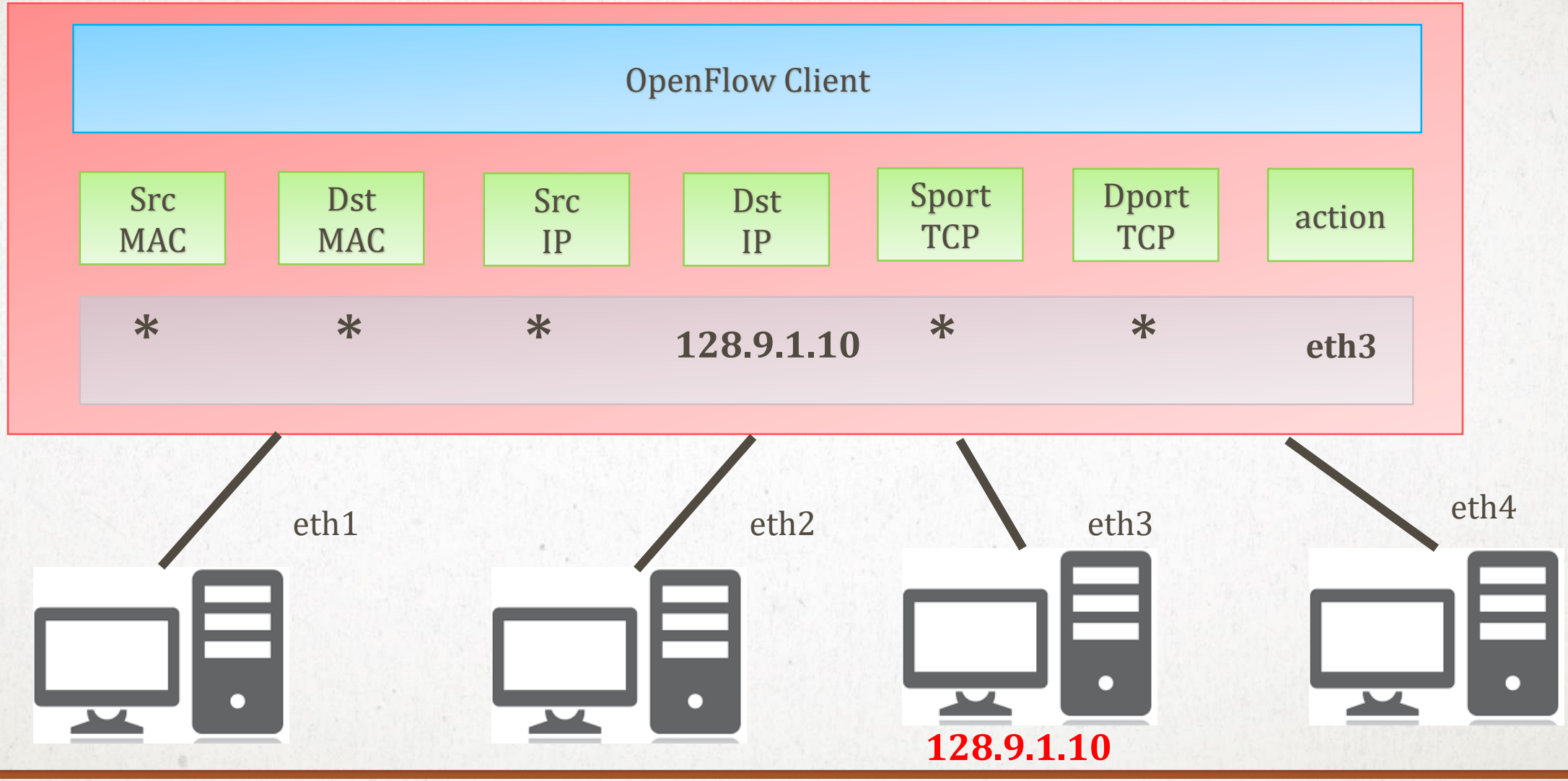
SDN MESSAGING INTERFACE



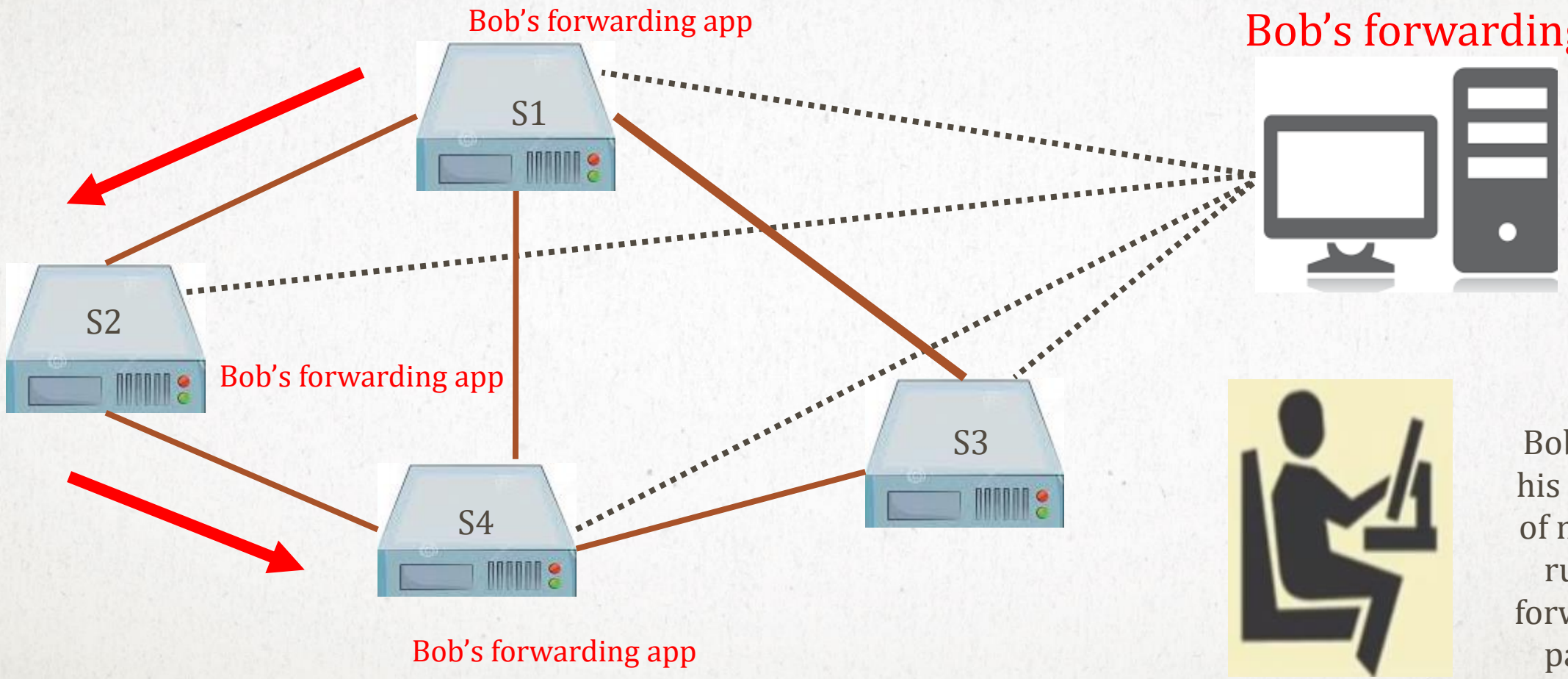
HOW OPEN FLOW WORKS



OPENFLOW EXAMPLE

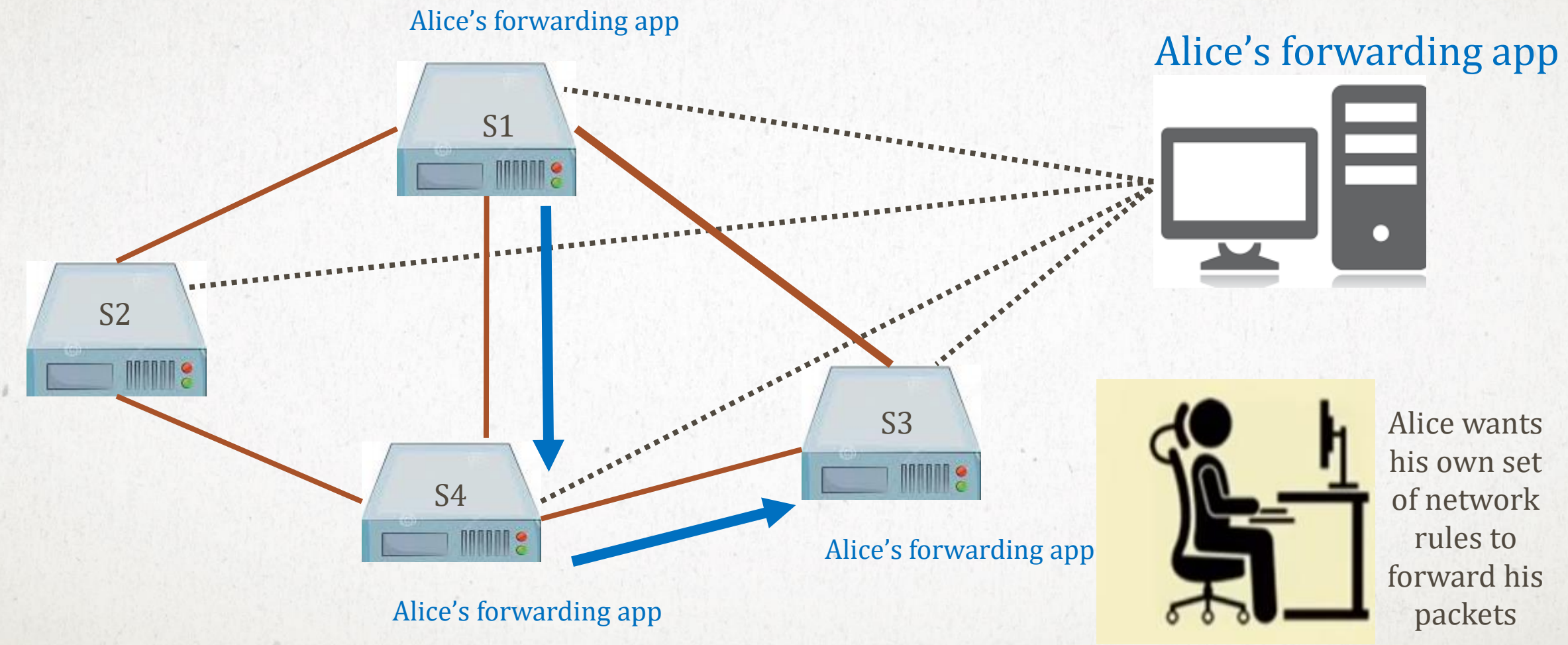


POWER OF SDN



Bob wants his own set of network rules to forward his packets

POWER OF SDN



Alice wants his own set of network rules to forward his packets

POWER OF SDN

Bob's forwarding app
Alice's forwarding app

S1

S2

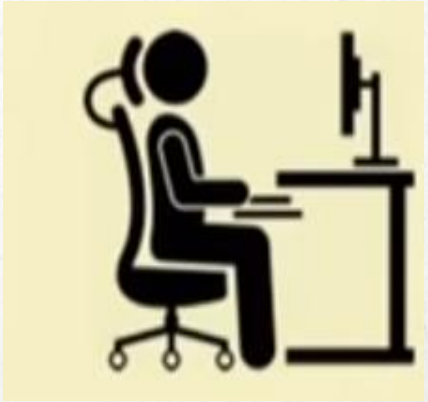
S3

S4

Bob's forwarding app
Alice's forwarding app

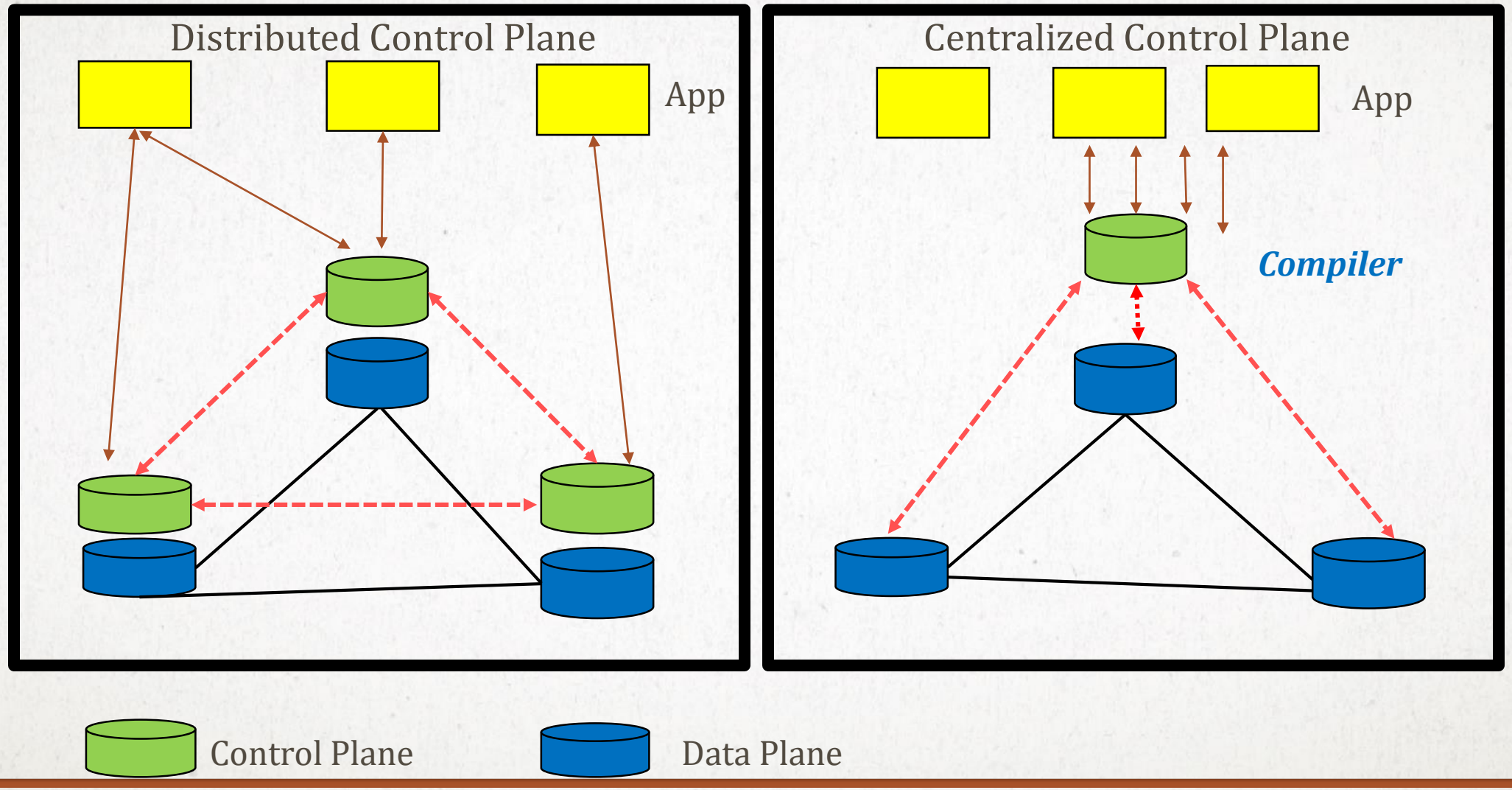
Bob's forwarding app
Alice's forwarding app

Alice's forwarding app

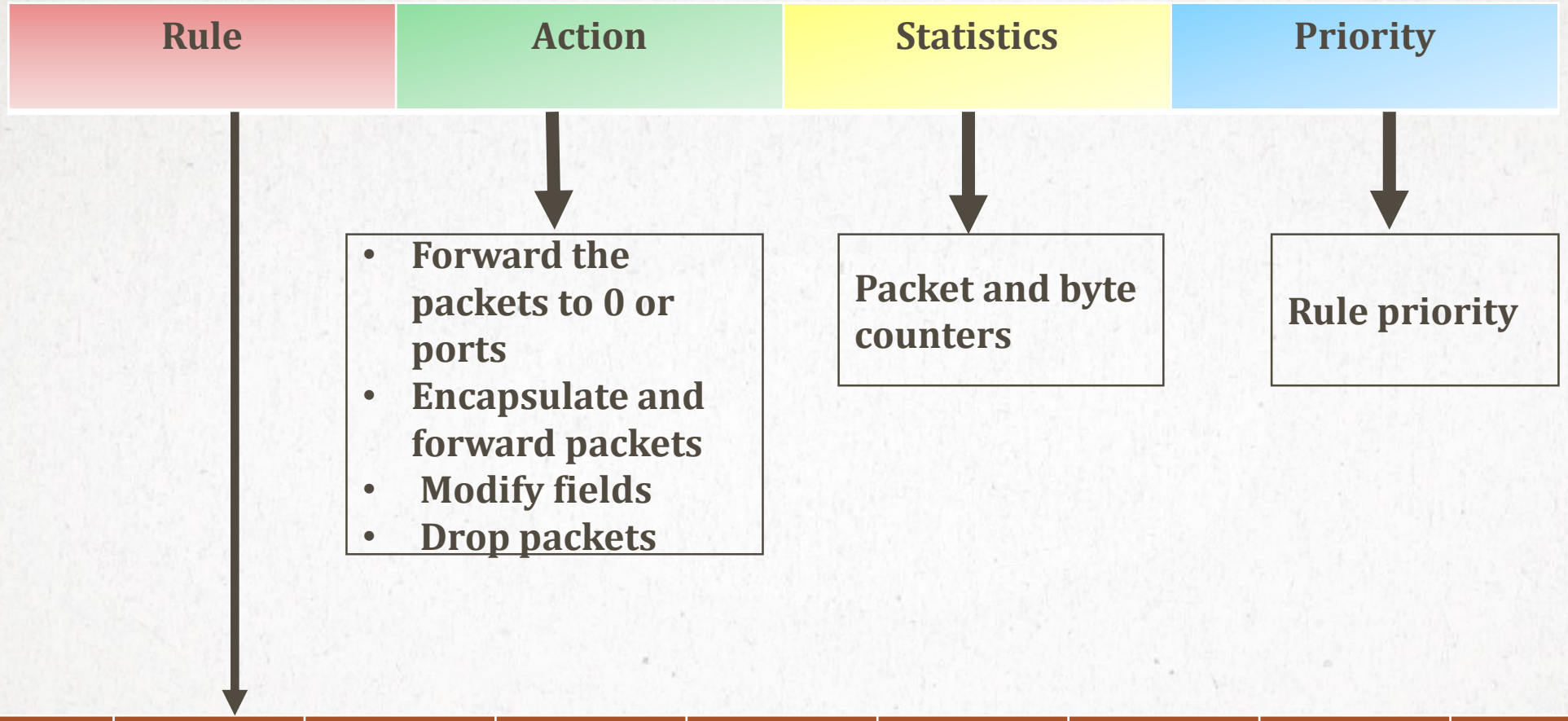


Alice wants his own set of network rules to forward his packets

POWER OF SDN (TRADITIONAL VS SDN)



OPEN FLOW TABLE



EXAMPLES OF OPEN FLOW TABLES

Switching

Switch Port	VLAN ID	MAC SRC	MAC DST	ETH Type	IP SRC	IP DST	IP ToS	TSPORT	TCP DPORT	Action
*	*	*	12.3F:..	*	*	*	*	*	22	eth2

Firewall

Switch Port	VLAN ID	MAC SRC	MAC DST	ETH Type	IP SRC	IP DST	IP ToS	TSPORT	TCP DPORT	Action
*	*	*	*	*	*	*	*	*	22	drop

EXAMPLES OF OPEN FLOW TABLES

Forwarding

Switch Port	VLAN ID	MAC SRC	MAC DST	ETH Type	IP SRC	IP DST	IP ToS	TSPORT	TCP DPORT	Action
*	*	*	*	*	*	202.2.*.*	*	*	*	eth2

Flow Switching

Switch Port	VLAN ID	MAC SRC	MAC DST	ETH Type	IP SRC	IP DST	IP ToS	TSPORT	TCP DPORT	Action
*	*	00:1F:..	14:B2:..	0800	202*1.*.*	212.19.*.*	*	80	8080	eth2

EXAMPLES OF OPEN FLOW TABLES

Source Routing

Switch Port	VLAN ID	MAC SRC	MAC DST	ETH Type	IP SRC	IP DST	IP ToS	TSPORT	TCP DPORT	Action
*	*	*	*	*	16.2.3*	202.2.*.*	*	*	*	eth2

VLAN Switching

Switch Port	VLAN ID	MAC SRC	MAC DST	ETH Type	IP SRC	IP DST	IP ToS	TSPORT	TCP DPORT	Action
*	2	*	14:B2:..	*	*	*	*	*	*	eth2, eth3

SECURITY ISSUES FOR OPENFLOW BASED CONTROLLER

- Flooding and DOS Attack
- Host Hijacking Attack
- Tampering Attack
- Spoofing Attack

FLOODING AND DOS ATTACK

- Attackers may device a DOS attack to make the controller down
- Control plane in SDN requires request from forwarding palne for flow rules.
- This led to no response from the controller and thus traffic becomes stand still.
- Hence if the attackers find that the OpenFlow switches are used, they will device resumption attacks.

HOST HIJACKING ATTACK

- It is a spoofing attack by exploiting Host Tracking Service
- The Host Tracking Service (HTS) is a network-wide view and an essential service in SDN controller.
- The issue with HTS is that it gets poisoned through host impersonation attack, man-in-the-middle attack or DoS attack
- If the controller is successfully hijacked sensitive information such as password, traffic management rules, routes etc. will be compromised.

TAMPERING ATTACK

- In SDN, Northbound API and Southbound API messages are used to manage the network
- In tampering attacks these Northbound API and Southbound API messages might be spoofed to insert malicious flow rules.
- This may lead to flow of traffic across SDN in random manner and misbehaviour of the entire network

SPOOFING ATTACK

- In the event of successfully spoofing attack, then the attacker can create and update the entries of flow table in SDN network components.
- Network management teams may not get a visible view of the actual flows from the production controller.
- Thus the attacker would have to control the network entirely
- Some other attacks which compromise the performance of the controller are replay attacks, host impersonate attacks, etc.
- Those attacks use different vulnerabilities in control plane to manipulate network efficiency.

VULNERABILITY OF SDN

Sl No	Security Object	Reversion	Description
1	Access Control	Open access	Susceptibility to be accessed by any element with-out restriction
2	Authentication	Nonidentification	Lack of identification in a distinctive way
3	Confidentiality	No-secrecy	Revealing its features and disclosing its communications
4	Non-Repudiation	Non-traceability	Not tracking its actions, its events and their actors
5	Integrity	Alterability	Susceptibility to be tampered in whole or in part
6	Availability	Disruption	Resources are partially or totally inaccessible

MANAGING THE CONTROLLER IN SDN

- Controller is the main component of the SDN
- There is a major possibilities of failure of the controller.
- Hence, the controller has to be prevented from malicious attacks or from failure due to technical reason.
- However, in case the controller is unavailable due to any reason there have to be a backup service so that the devices connected to the controller is managed with a stop gap solution.

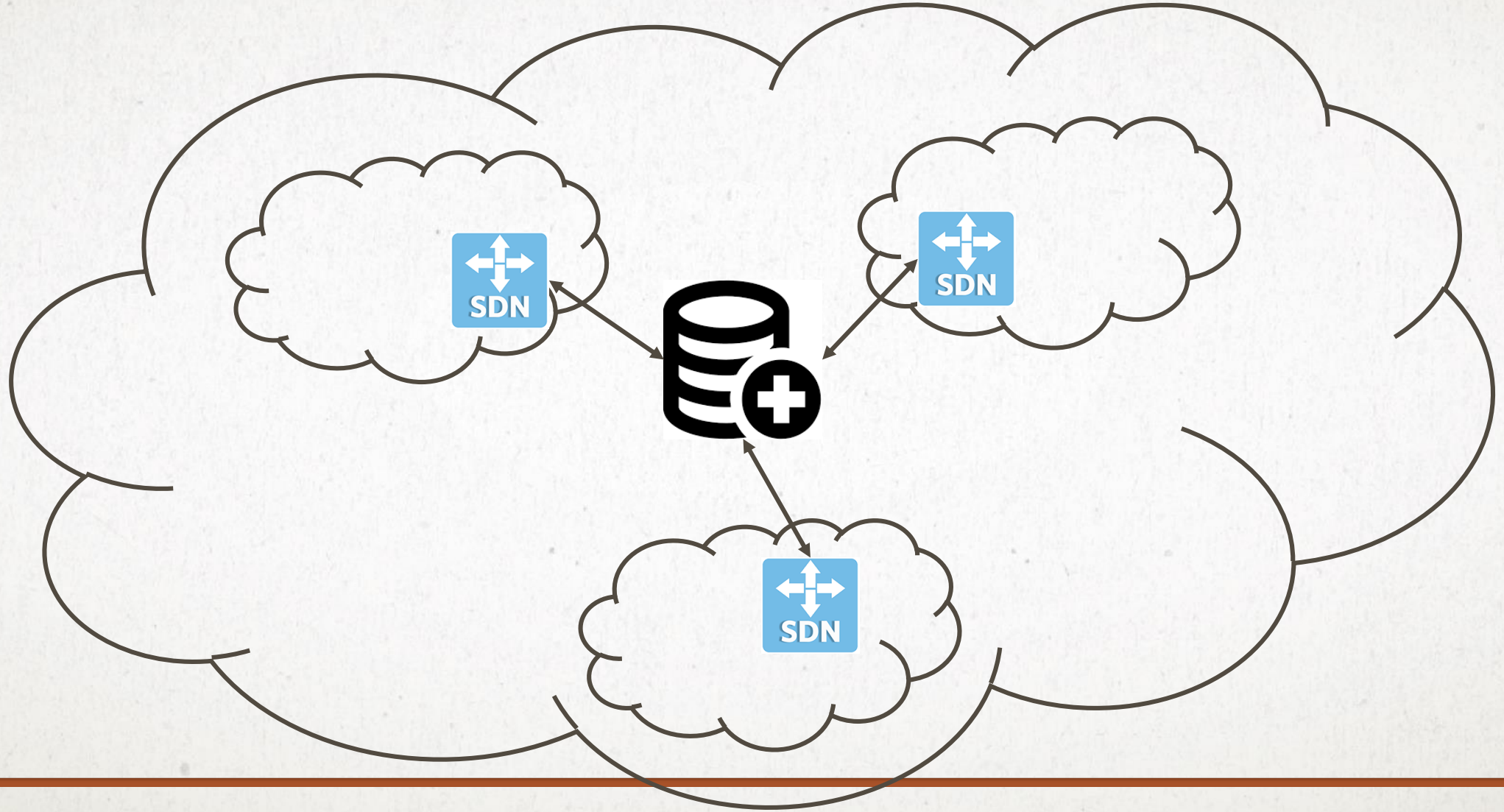
PROVISIONING OF CONTROLLER IN THE EVENT OF FAILURE

- In the event of the failure of the controller in the single controller environment there is a massive chances of the network breakdown.
- Other solutions related to multiple controller environment either in cluster form or replicated scenario leads to degradation in the performance.
- Therefore, a requirement is felt for a solution which will have the advantages ofd both scenario.

THE SOLUTION:

- A solution is being proposed by Dutta and Chatterjee[], where the entire network is divided into a number of subnets.
- These subnets will generally cater to different VLAN in a multi-VLAN environment
- Each of these subnets which is a SDN is being managed by dedicated controller
- Each controller manages the subnets as per the topology and policies.
- This is being shown in fig in the next slide.

SCHEMATIC ARCHITECTURE OF SDN WITH BRS



PROPOSED TECHNIQUE

- In the event of failure of a particular controller the BRS will assign the job of the controller which is not working to a topologically adjacent controller.
- In this event the current state of the controller is required.
- This information will be fetched from the BRS to the assigned controller.
- Once the existing controller is revoked, the current state will be assigned to the existing controller by BRS again.
- Hence the controllers managing various subnets are required to update their current state to the BRS over a regular period of time.
- The proposed BRS help to achieve better reliability and performance.

STEPS OF PROPOSED TECHNIQUE

- Controller Failure Detection
- Authentication Process
- Controller Recovery Process

CONTROLLER FAILURE DETECTION

- In this proposed architecture the BRS will ping each controller on a regular time interval.
- Once the BRS detect a controller failure, it will send a request to the topologically adjacent controllers for management of the Control Plane of that subnet on adhoc basis.
- One of the adjacent controllers will be selected for management of an additional subnet.
- The assigned controller will be required to undergo an authentication process.
- Once the existing controller is recovered and start its normal function, the control plane management will be returned to the exiting controller

AUTHENTICATION PROCESS

- The authors use the arithmetic encoding technique for authentication between the BRS and the respective SDN Controller.
- The BRS will have Probability of Occurrence (PO) information shared with respective controller.
- The validity of this information is of fixed duration of time.
- Beyond this time, a new PO is shared suing the same mechanism.

ALGORITHM FOR ARITHMATIC ENCODING

//Algorithm for Arithmetic Encoding

Begin

Step 1: Create interval range for each character based on PO

Step 2: Take a string as a user input

Step 3: Initialize n = string length

Step 4: Initialized low value (L) =0, High value (H) =1

Step 5: Take L & H of the first character. Determine the range R=H-L.

Step 6: Calculate Low next (Ln) & High next (Hn) for next characters (string length <=n) using this bellow procedure and update the values of L & H.

Repeat { Ln=L+R*lower interval

Hn=L+R*higher interval

Swap L=Ln

Swap H=Hn }

Proceed to step 5 End

CONTROLLER RECOVERY PROCESS

- In this proposed architecture after recovery of the exiting controller for a particular subnet, the BRS will hand over the control plane to it.
- The same process of authentication using the exiting technique is followed for handing over process.
- Once handing over process will complete, the assigned controller is relieved from the additional load of management of a subnet

CHALLENGES WITH SDN

- Security is both a benefit and a concern with SDN technology. The centralized SDN controller presents a single point of failure and, if targeted by an attacker, can prove detrimental to the network.
- Ironically, another challenge with SDN is there's really no established definition of *software-defined networking* in the networking industry. Different vendors offer various approaches to SDN, ranging from hardware-centric models and virtualization platforms to hyper-converged networking designs and controller-less methods.
- Some networking initiatives are often mistaken for SDN, including white box networking, network disaggregation, network automation, and programmable networking. While SDN can benefit and work with these technologies and processes, it remains a separate technology.
- SDN technology emerged with a lot of hype around 2011 when it was introduced alongside the OpenFlow protocol. Since then, adoption has been relatively slow, especially among enterprises that have smaller networks and fewer resources. Also, many enterprises cite the cost of SDN deployment to be a deterring factor.

SUMMARY

- SDN encompasses several types of technologies, including functional separation, network virtualization, and automation through programmability.
- Originally, SDN technology focused solely on the separation of the network control plane from the data plane. While the control plane makes decisions about how packets should flow through the network, the data plane actually moves packets from place to place.
- In a classic SDN scenario, a packet arrives at a network switch, and rules built into the switch's proprietary firmware tell the switch where to forward the packet. These packet-handling rules are sent to the switch from the centralized controller.
- The switch -- also known as a *data plane device* -- queries the controller for guidance as needed, and it provides the controller with information about the traffic it handles. The switch sends every packet going to the same destination along the same path and treats all the packets the exact same way.

REFERENCES:

- NPTEL Lectures “Software Defined Networking” by Prof Sandip Chakrabarty.
- NPTEL Lectures “Software Defined Networking” by Prof Sudip Mishra.
- “Introduction to Software Defined Networking” by Prof Raj Jain.
- https://searchnetworking.techtarget.com/definition/software-defined-networking-SDN
- www.opennetworking.org/sdn-definition/
- https://en.wikipedia.org/wiki/Software-defined_networking

THANK YOU