IIT KHARAGPUR

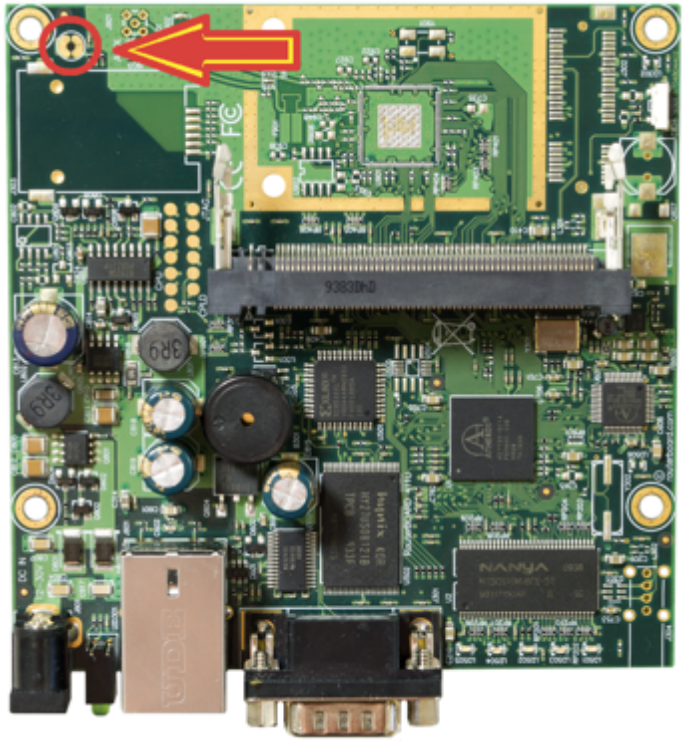
NPTEL ONLINE
CERTIFICATION COURSES

COMPUTER NETWORKS AND INTERNET PROTOCOLS

SOUMYA K GHOSH
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

SANDIP CHAKRABORTY
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

IP Routers



The Life of a Router

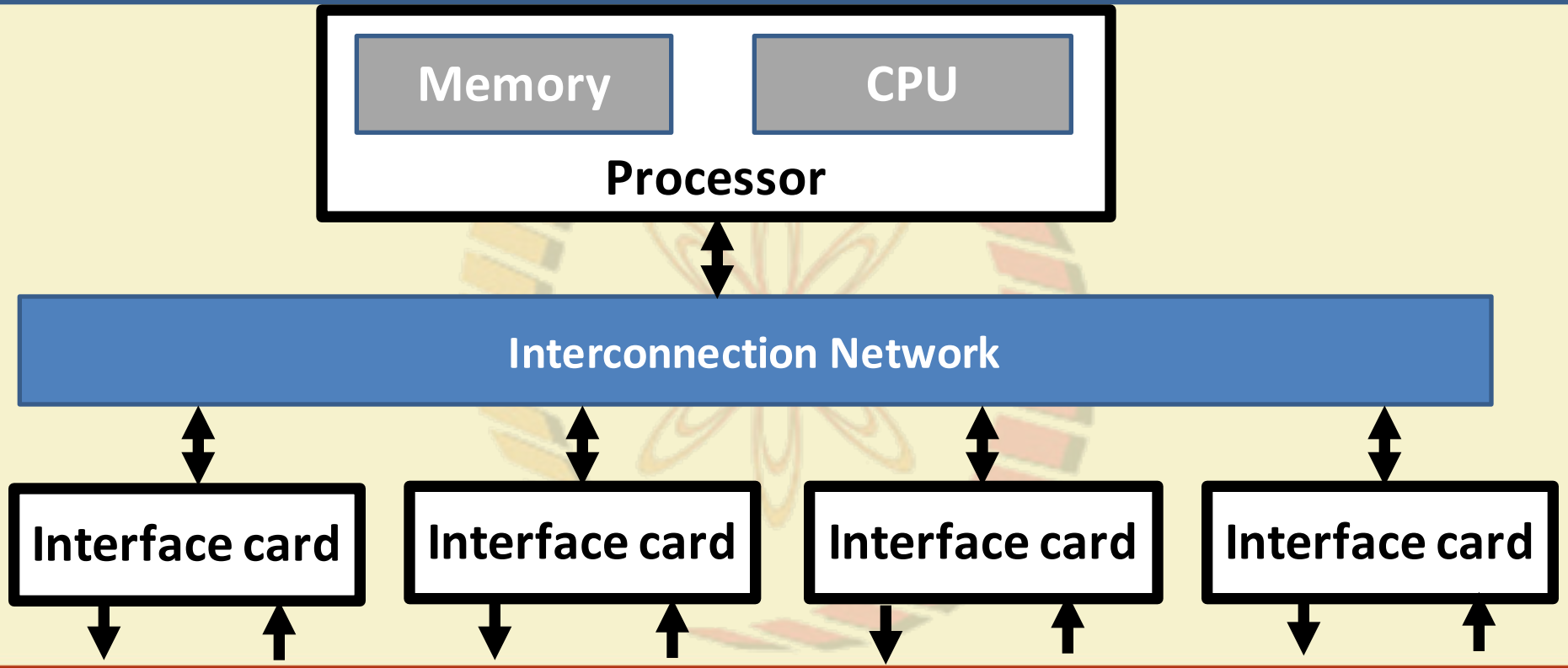
- Do
  - Find Path
  - Forward, forward, forward, forward, forward
  - Find Path
  - Forward, forward, forward, forward, forward
- Repeat until powered off

- Two basic operations –
  - Construct the routing table **– the control plane**
  - Do a routing match and forward the packet to a dedicated interface – **the data plane**

Evolution of Router Architecture

- 1st Generation (Until 1980's): Standard Computer
- 2nd Generation (Early 1990's): Delegate to interfaces
- 3rd Generation (Late 1990's): Distributed Architecture
- 4th Generation (Early 2000's): Distributed over multiple racks
- 5th Generation (Today): Software Defined Routing

Basic Architectural Components of a Router



Router Hardware

- Processor is responsible for control functions **(route processors)**
  - Construct the routing table based on the routing algorithm

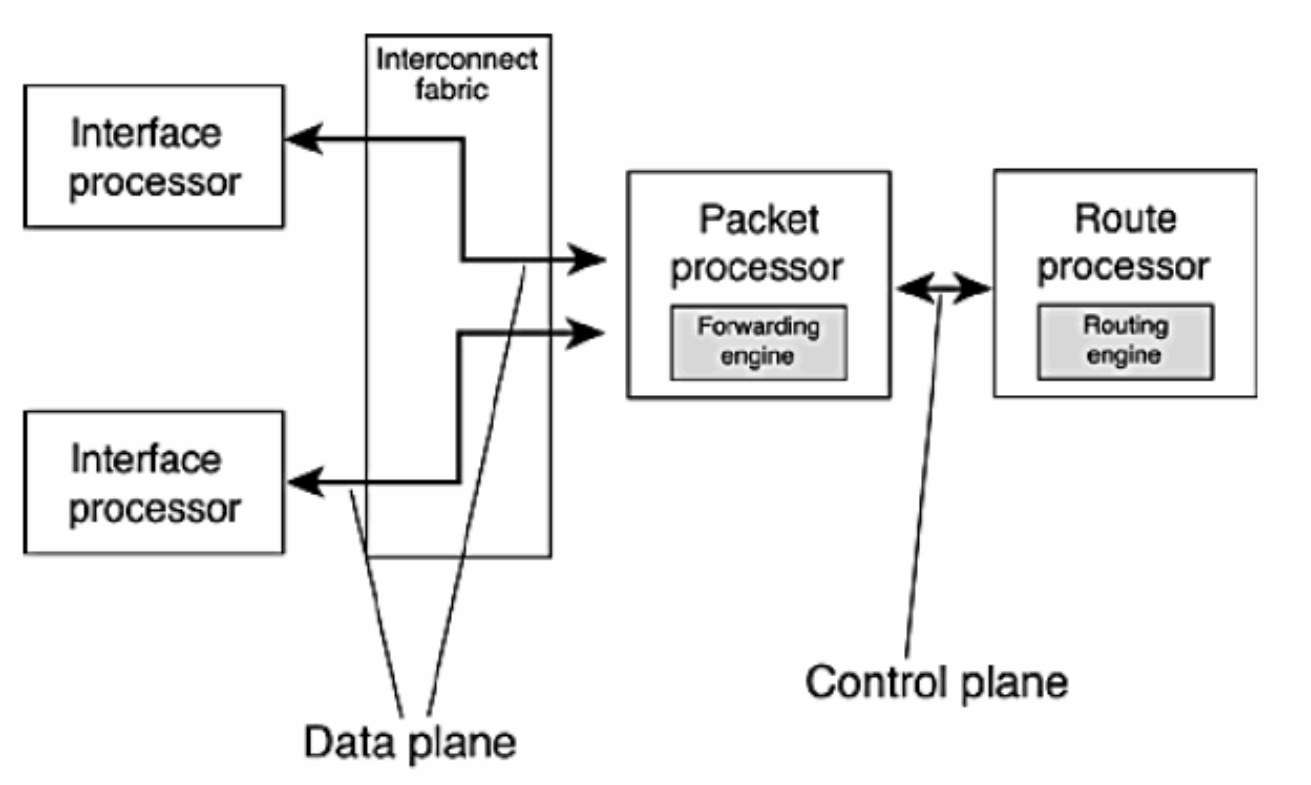
- Forwarding is done at the interface card
  - Route match needs to be very fast
  - Specialized hardware – Ternary Content-Addressable Memory (TCAM)

Router Hardware

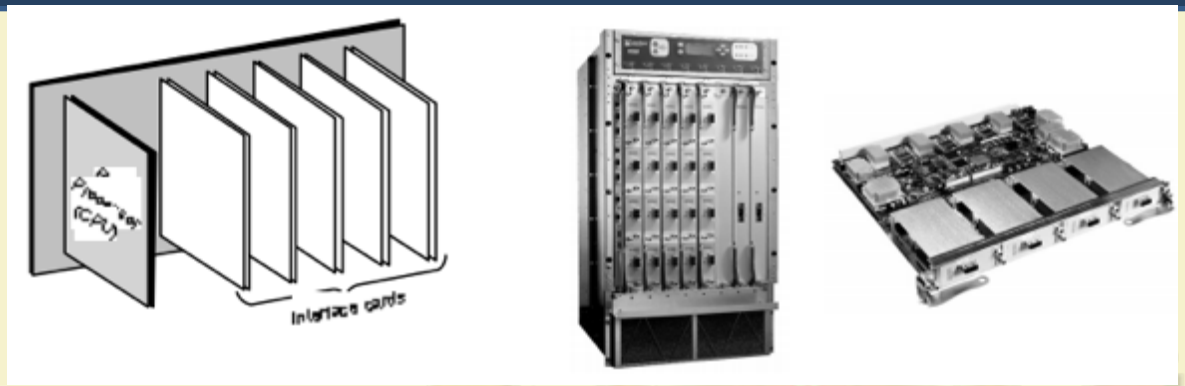
Routing Table Construction

Forwarding

Router Internals

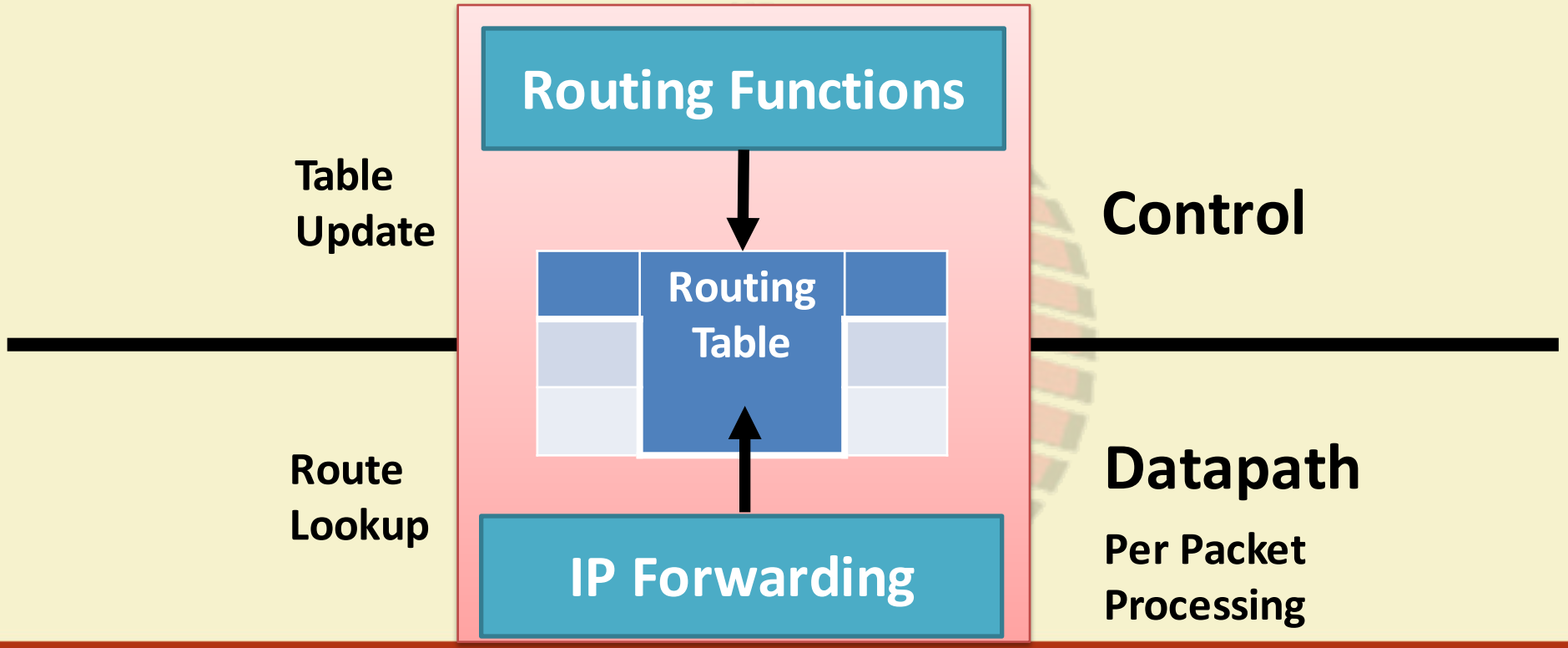


Slotted Chassis – The Physical Infrastructure for Routers



- Large routers are built as slotted chassis
  - Interface cards are inserted in the slots
  - Route processor is also inserted at a slot
- Simplifies repairs and upgrades of components

Functional Components



Control Plane in a Router

- Note that a router is a **special purpose computer**
- Implemented as a software (router OS) that supports the basic computing functionalities to run a router along with routing functionalities
- Routing protocols are implemented in the router OS
- Example: Cisco IOS

Routing Functions

- **Route Calculation**
- **Maintenance of the routing table**
- **Execution of the routing protocol**

- On commercial routers, routing functions are handled by a single general purpose processor, called the **route processor**

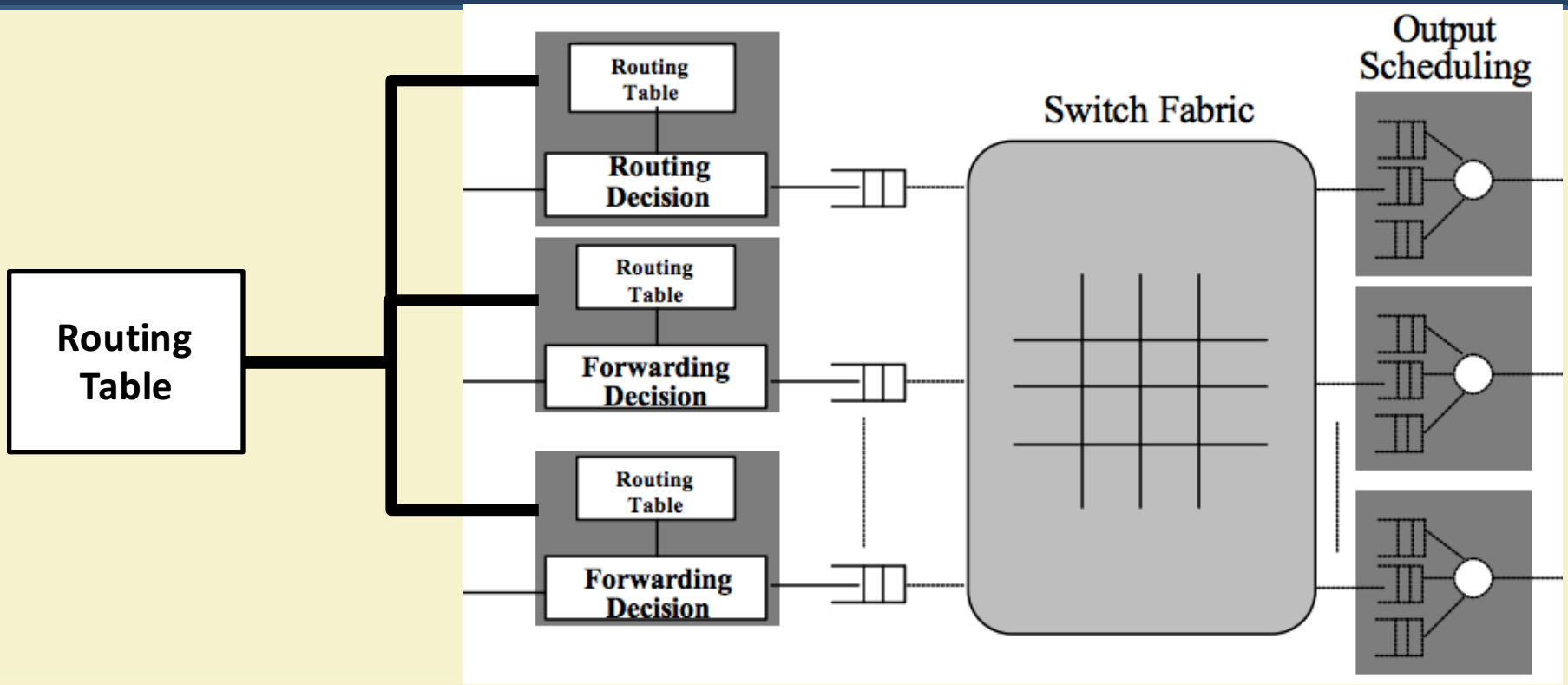
Data Plane of a Router

- Implement forwarding functionalities – make a route lookup and forward the packet at the destination interface
- Functionality is similar to a L2 switch – use switch fabric (the mapping from input ports to output ports) to forward the packet from one interface to another
- Maintains interface buffer – to implement store and forward functionality

IP Forwarding

- Per packet processing of the IP packets
- IP forwarding is distributed, handled by individual interface controllers
- Special hardware devices are used - TCAM

Per Packet Processing – Basic Architectural Components

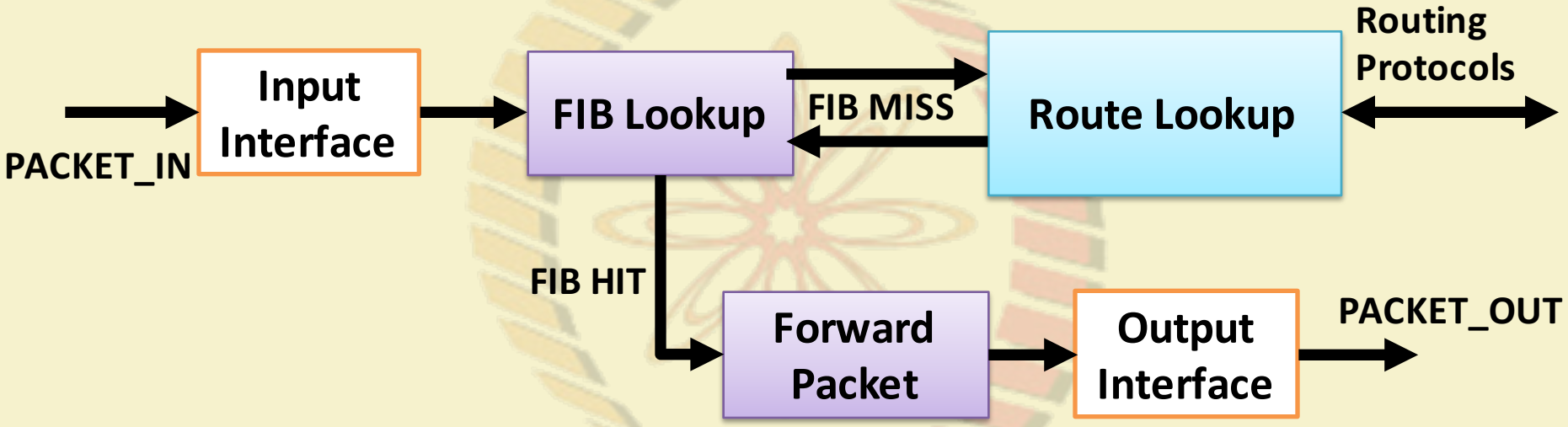


Forwarding Information Base (FIB)

- The interfaces maintains a *forwarding information base* (FIB) – a mapping from input interface to output interface

- A replica of the routing table used at the interfaces for making the forwarding decision

Forwarding Information Base (FIB)



Difference between RIB and FIB

- **Routing Information Base (RIB) –** The routing table, implemented in software, is maintained at the control plane

- **Forwarding Information Base (FIB)** – The copy of the required routes maintained in interface TCAM hardware

- RIB is dynamic and maintains entire routing information, FIB is updated whenever required

RIB and FIB

The RIB

172.16.1.0	255.255.255.0	172.16.1.2	Eth0
172.16.2.0	255.255.255.0	172.16.2.2	Eth1
10.3.0.0	255.255.0.0	10.3.1.1	Eth3
10.9.0.0	255.255.0.0	10.9.1.1	Eth4

FIB at Eth0

172.16.2.0	255.255.255.0	172.16.2.2	Eth1
10.3.0.0	255.255.0.0	10.3.1.1	Eth3
10.9.0.0	255.255.0.0	10.9.1.1	Eth4

FIB at Eth1

172.16.1.0	255.255.255.0	172.16.1.2	Eth0
10.3.0.0	255.255.0.0	10.3.1.1	Eth3

RIB Feeds FIB

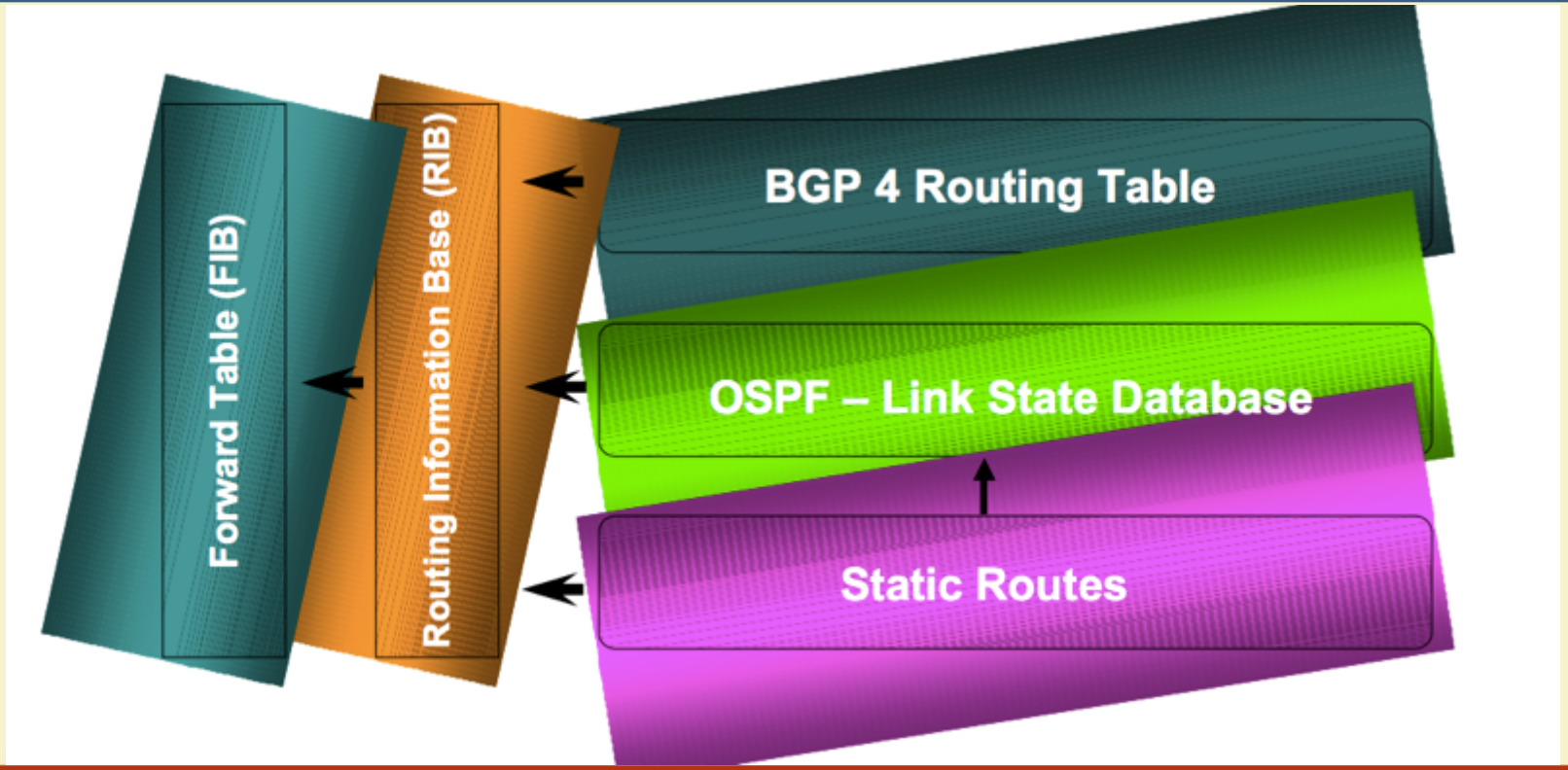
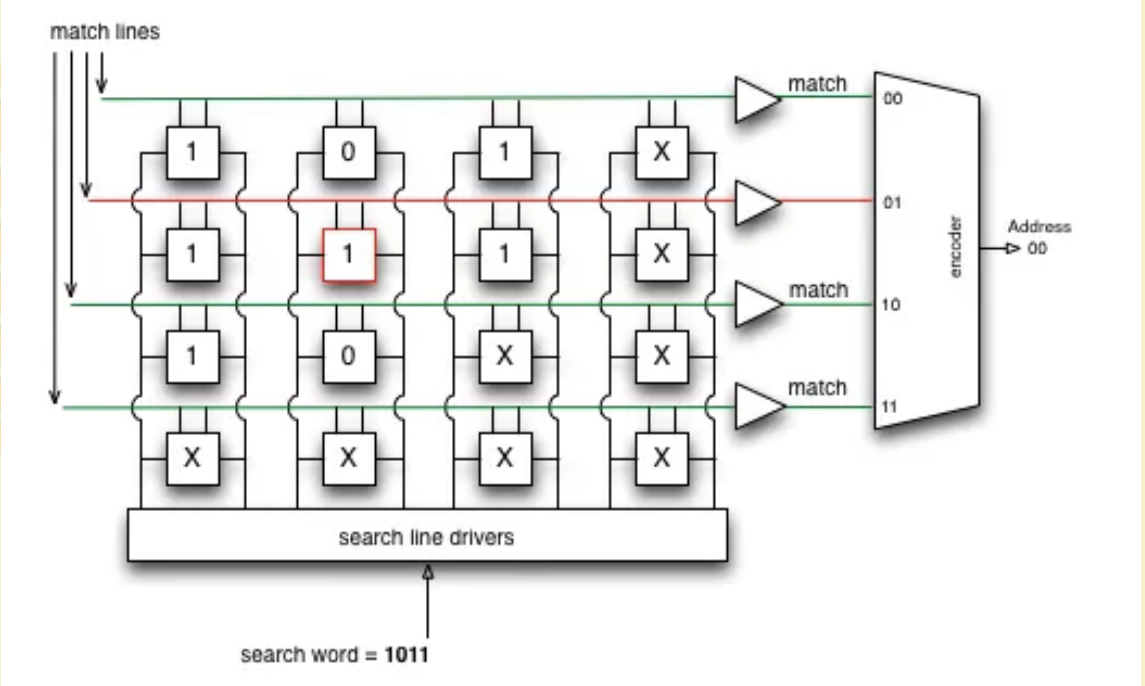


Image source: Cisco

Basic TCAM Architecture

Pfx/mask	TCAM format
101/3	101X
111/3	111X
10/2	10XX
0/0	XXXX

Image Source: http://thenetworksherpa.com/




thank you!

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

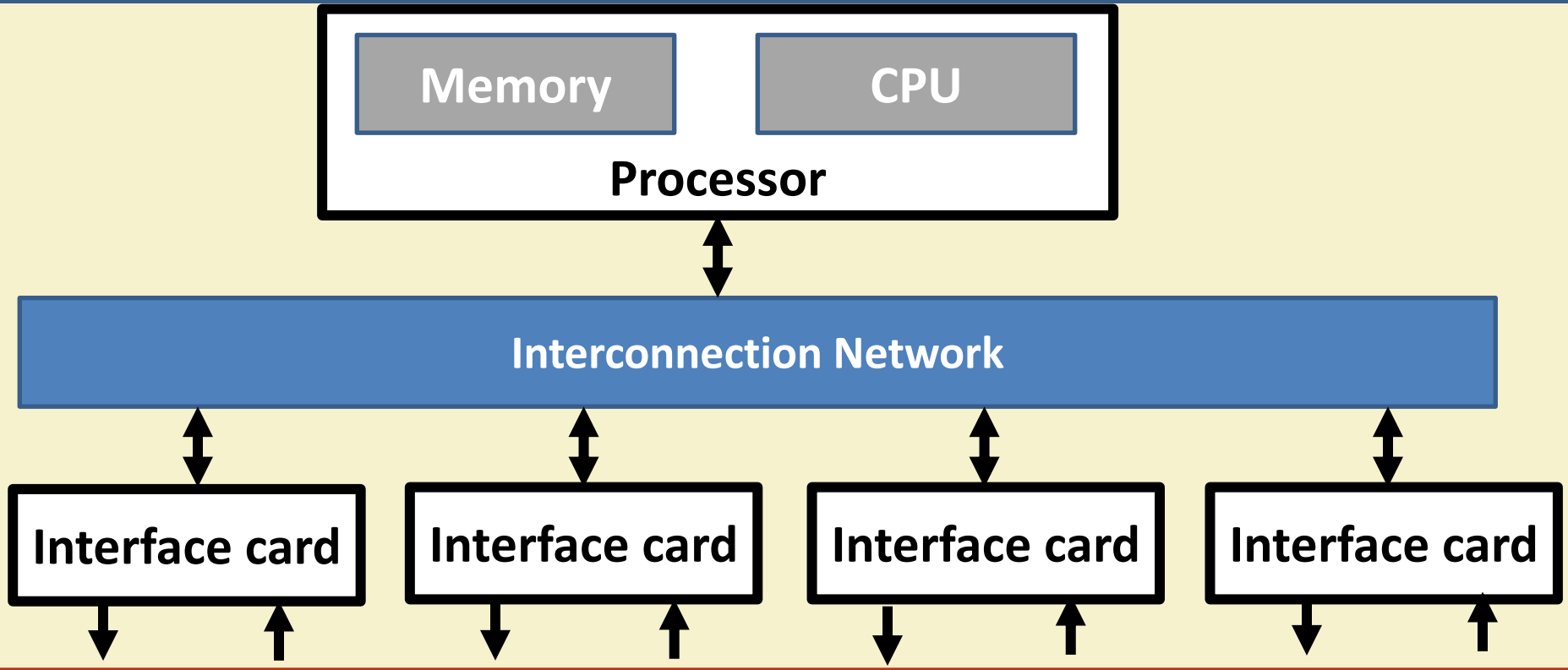
COMPUTER NETWORKS AND INTERNET PROTOCOLS

SOUMYA K GHOSH
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

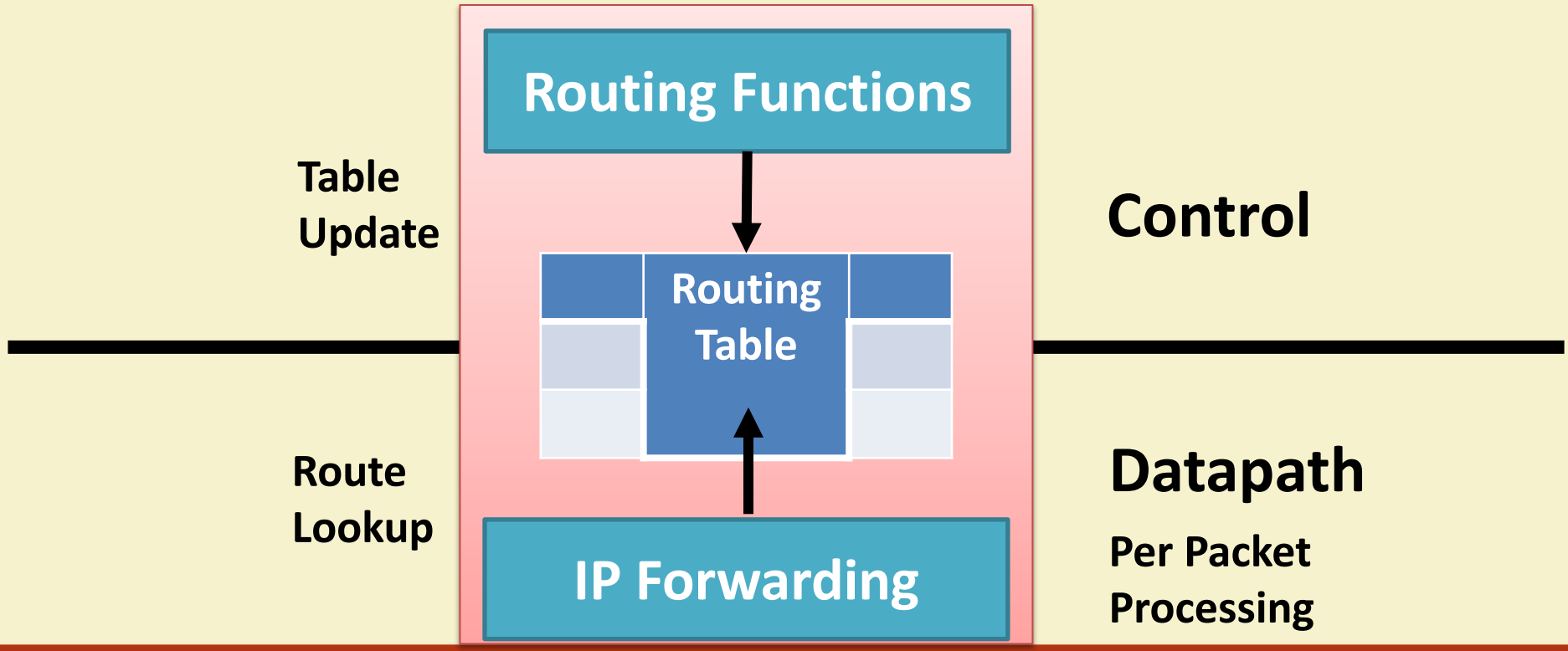
SANDIP CHAKRABORTY
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

IP Routers - Internals and Demo

Basic Architectural Components of a Router



Functional Components



Ternary Content Addressable Memory (TCAM)

- Specialized high speed memory - searches its entire content in a single clock cycle
- **Ternary** - store and query data using three different inputs - 0, 1 and X (don't care or wildcard)
- Searching is based on pattern matching, ex. 110X - match content that starts with 110*

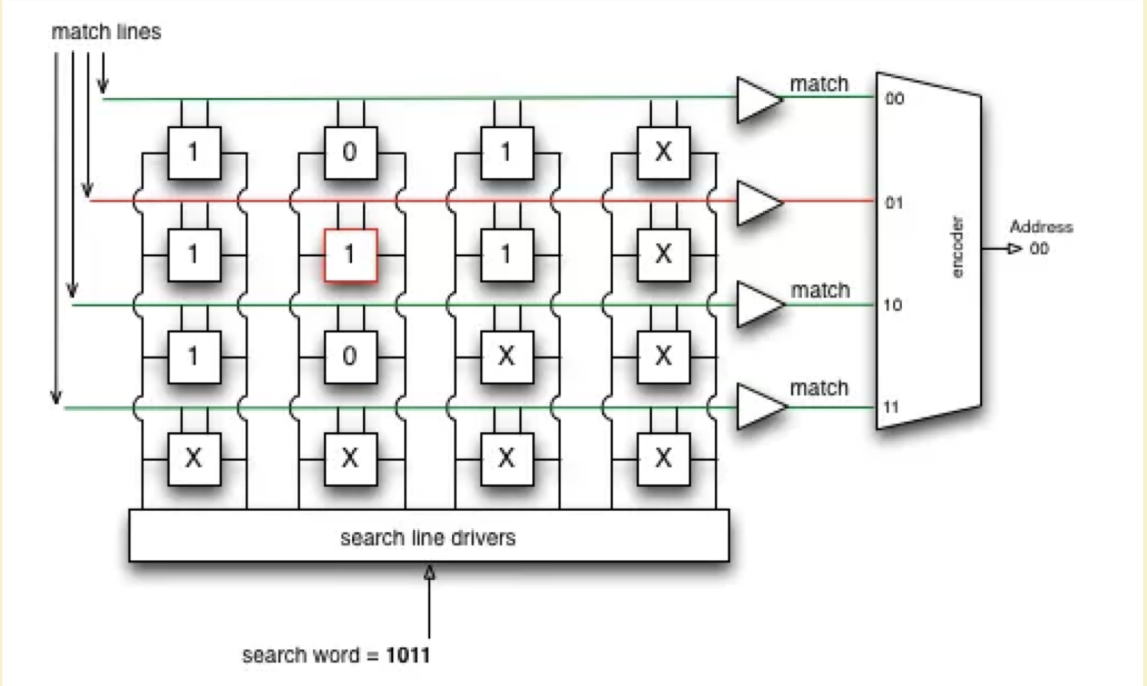
CAM vs RAM

- RAM is accessed via the address where the data is stored
- CAM can be addressed by performing a query for the content itself - retrieves the address where data is found
  - Opposite to RAM !!
- Much faster than RAM

Basic TCAM Architecture

Pfx/mask	TCAM format
101/3	101X
111/3	111X
10/2	10XX
0/0	XXXX

Image Source: http://thenetworksherpa.com/



ip and route commands

- `ip address show` : Show IP addresses
- `ip link show` : Show link information
- `ip -s address show` : Show statistics
- `route -n` : Show the IP routes
- `ip route list` : Show the IP routes
- `ip route add 172.16.2.30/24 dev eth0` : Add a new route
- `traceroute 172.16.2.30` : Trace the route of a destination

thank you!

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

COMPUTER NETWORKS AND INTERNET PROTOCOLS

SOUMYA K GHOSH
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

SANDIP CHAKRABORTY
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

Software Defined Networking

Software-defined networking (SDN) is a network framework which involves in **separating** a network's **control functions** from its **data forwarding functions**, centralizing its intelligence, and abstracting its underlying architecture from applications and services.

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

Control and Data Plane

- Traditional networking devices are proprietary
  - Vendors decide software (control plane) and hardware (data plane)
- No standardization

These two modules are "baked" in Unchangeable

What does separating control and data plane mean?

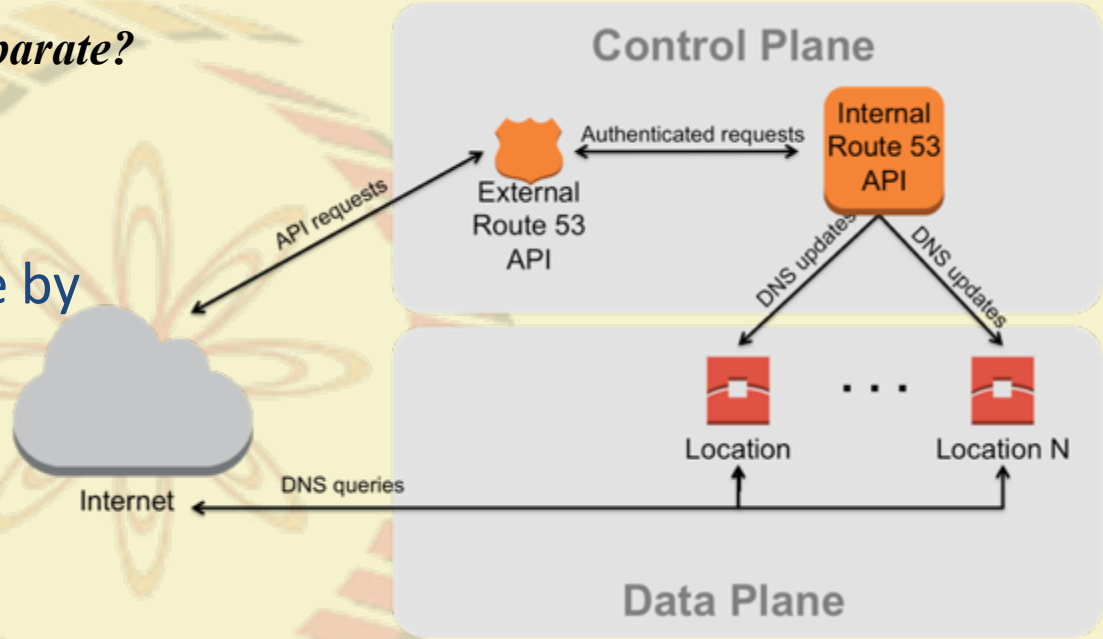
But what if they were separate?

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

What does separating control and data plane mean?

But what if they were separate?

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

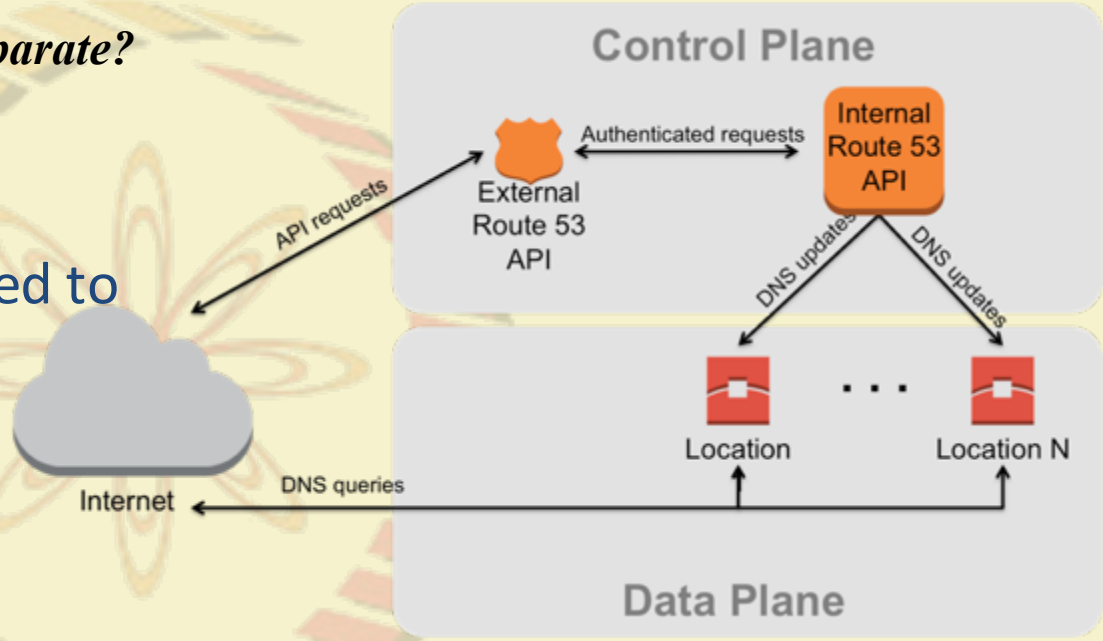


What does separating control and data plane mean?

But what if they were separate?

Advantages:

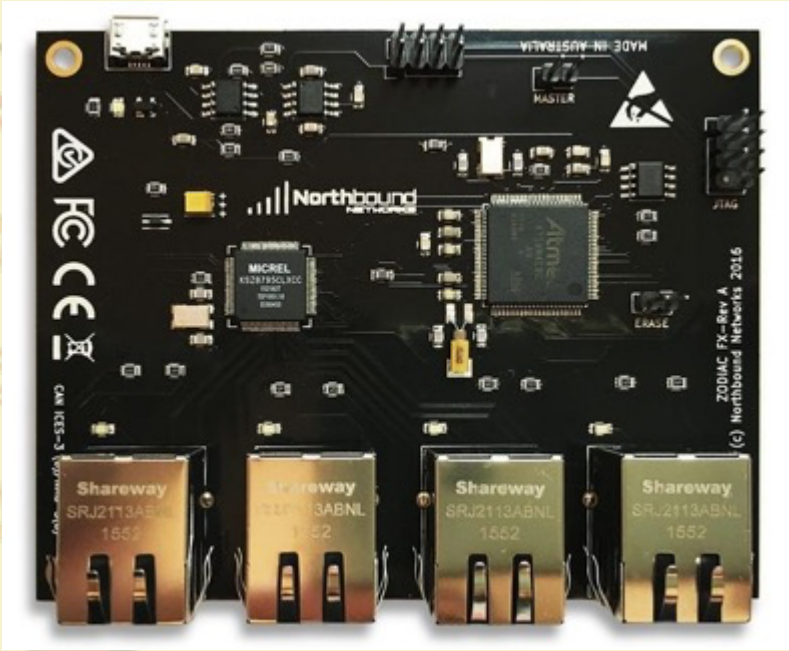
- Features are no longer limited to what the vendor provides
- Community development
- Longer life of products



How Does SDN Work

- Compared to traditional networks, a software defined network has 2 types of devices
  - Controller
  - Switches

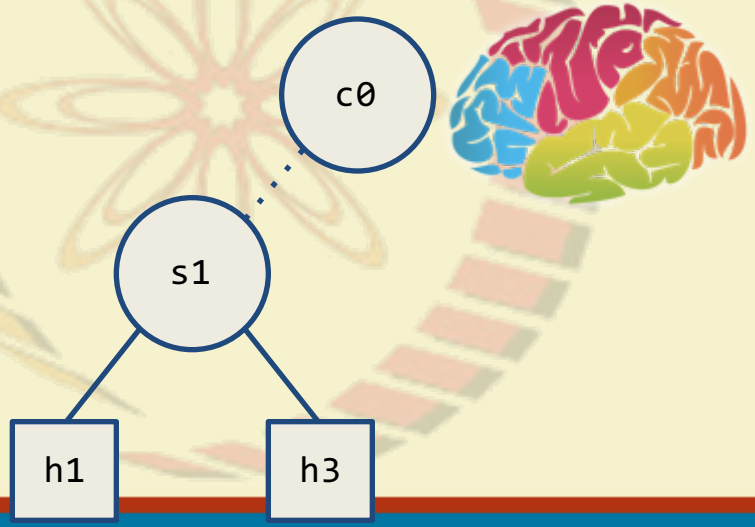
- The switches in SDN are **blind**
  - No built-in features
  - Need to be instructed by the controller



Zodiac FX - A Tiny SDN Switch

SDN Controller

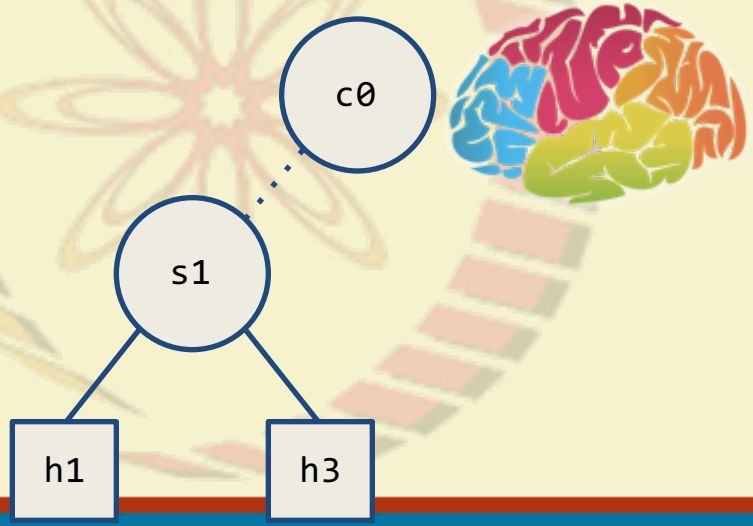
- "**Brains**" of the network
- All policies, routing logic are placed in the controller
- "**Teaches**" the switches what to do with an unknown packet



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



Forwarding Request from h1 to s1

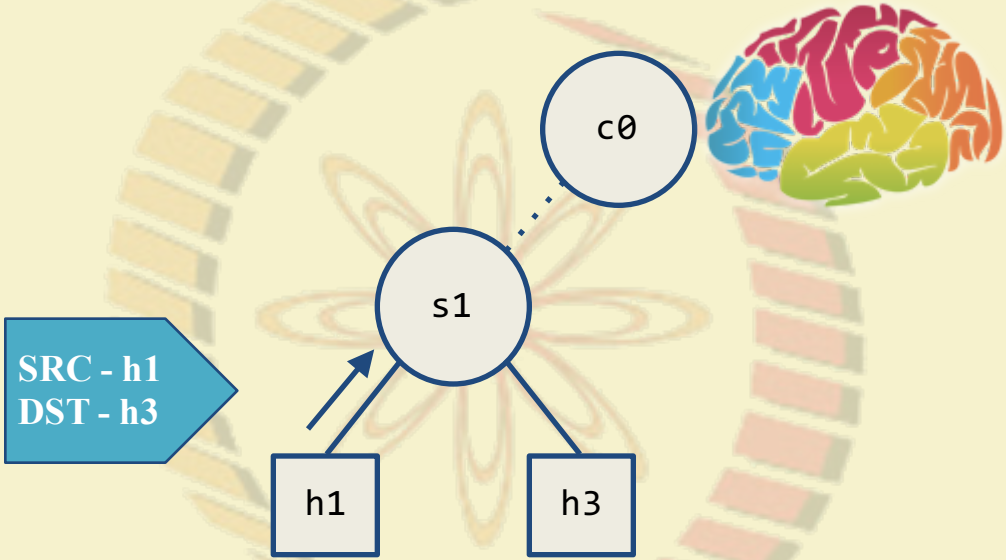
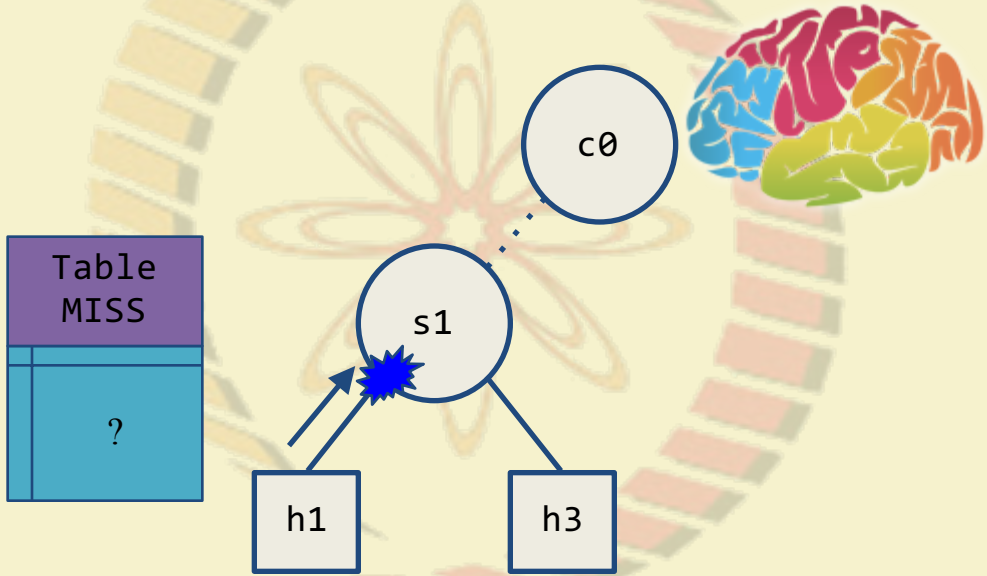


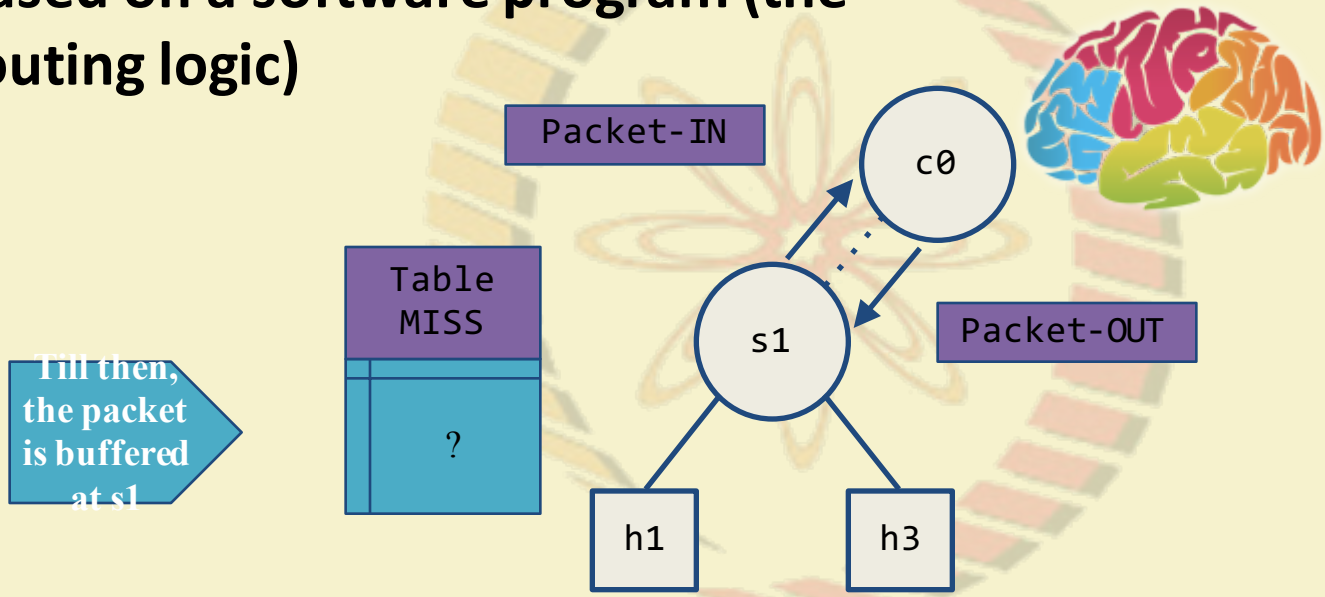
Table Miss at s1

No rule is there for h1->h2 at the beginning, so a FIB miss at S1



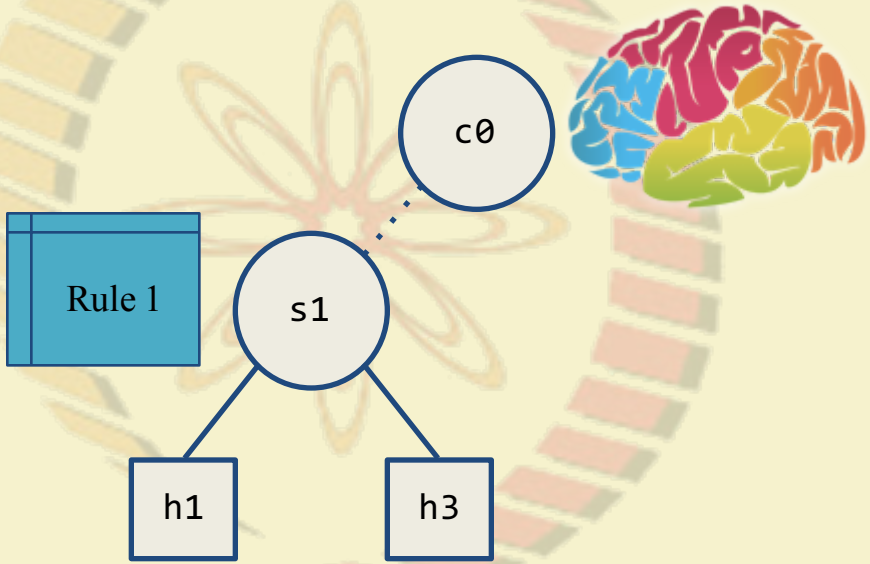
Packet-IN to Controller

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



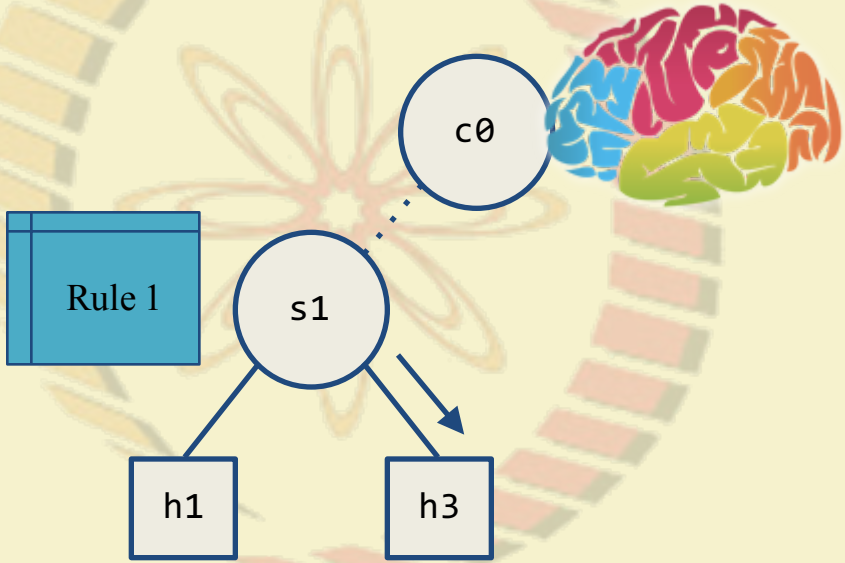
Flow Rule Set at S1…

The rule is installed at the FIB of S1



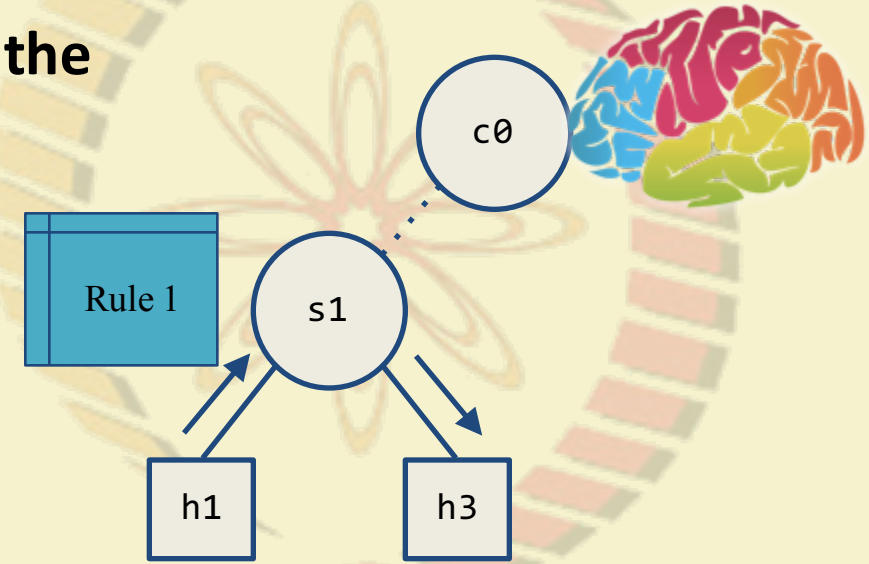
All Future Packets from h1->h3

The buffered packet is forwarded based on that rule



Buffered Packet Forwarded to h3…

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



The SDN Architecture

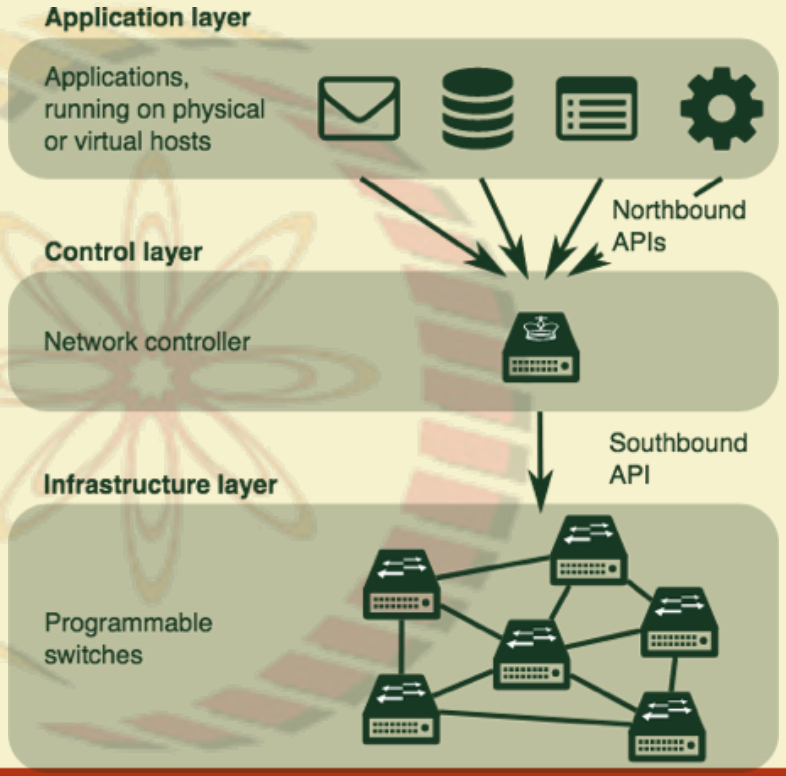


Image source: qmonnet.github.io

Traditional Network vs SDN

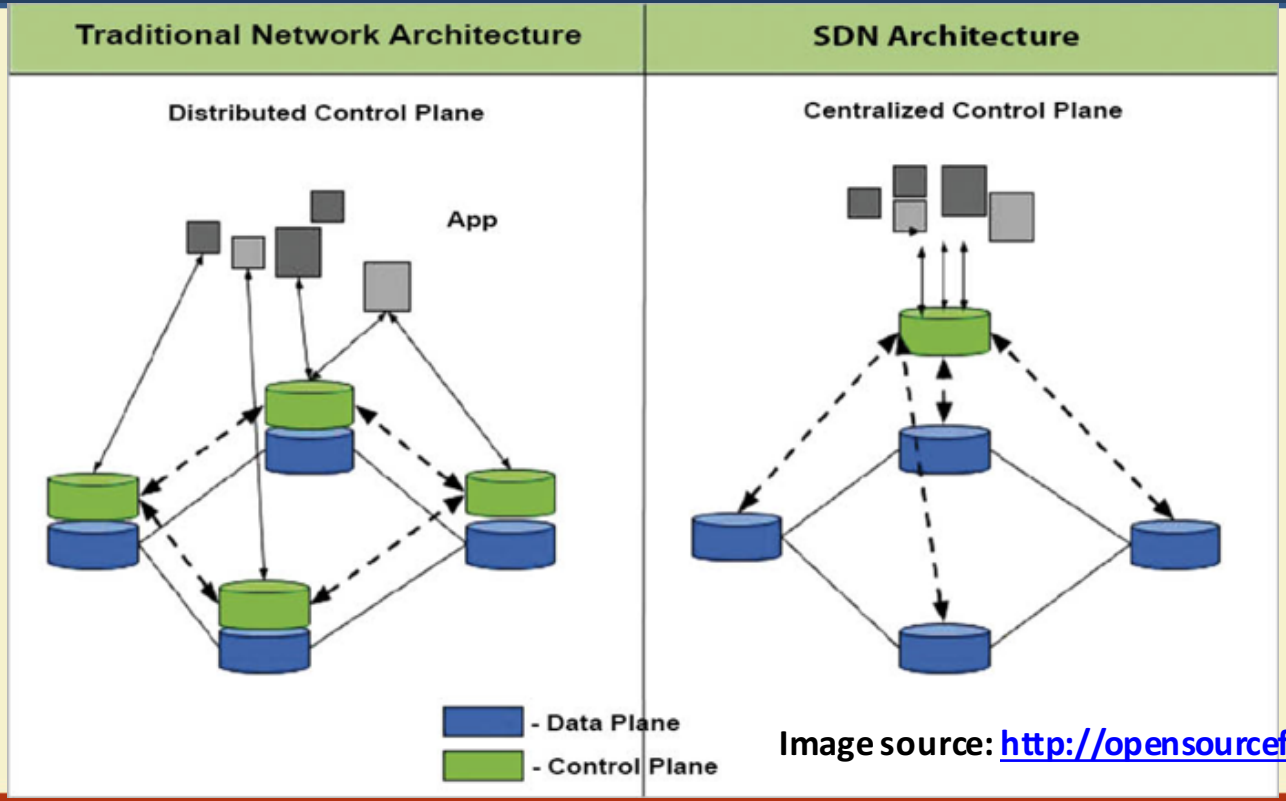


Image source: http://opensourceforu.com/


thank you!

IIT KHARAGPUR

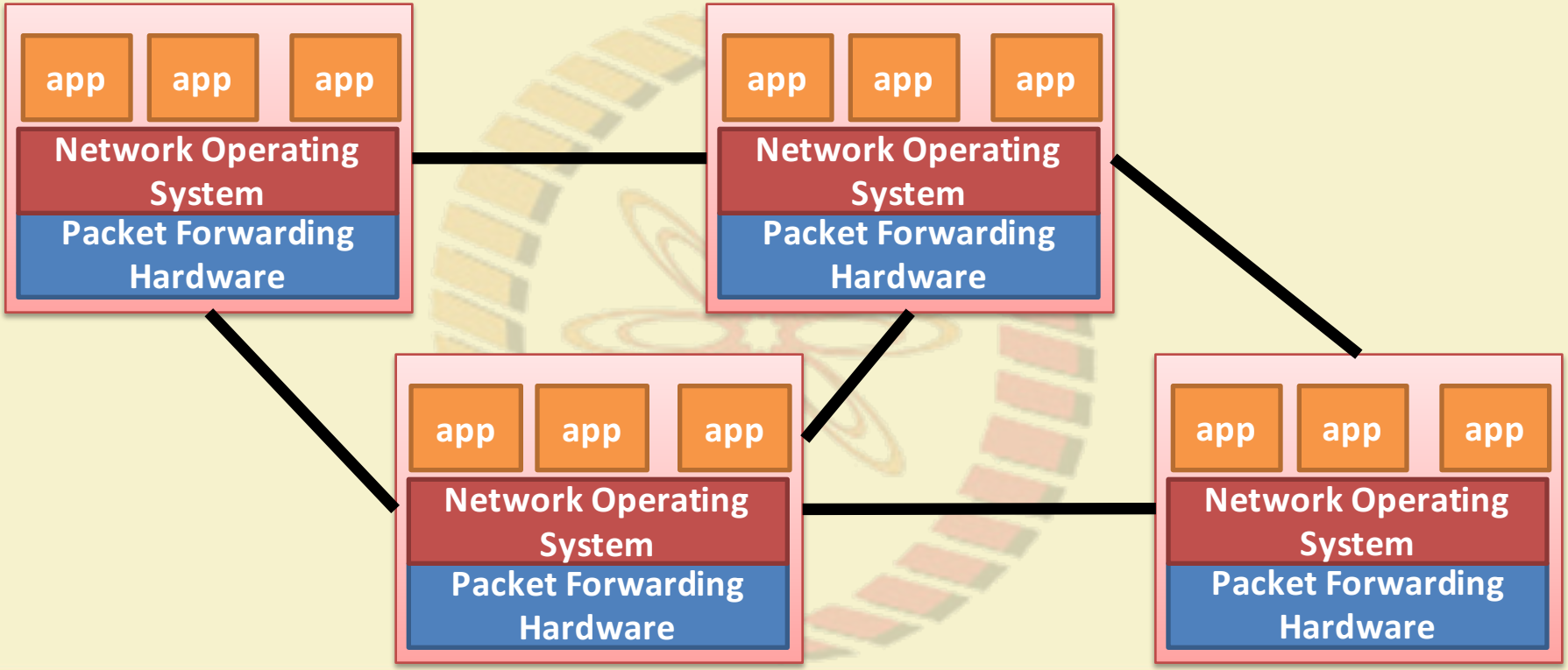
NPTEL ONLINE
CERTIFICATION COURSES

COMPUTER NETWORKS AND INTERNET PROTOCOLS

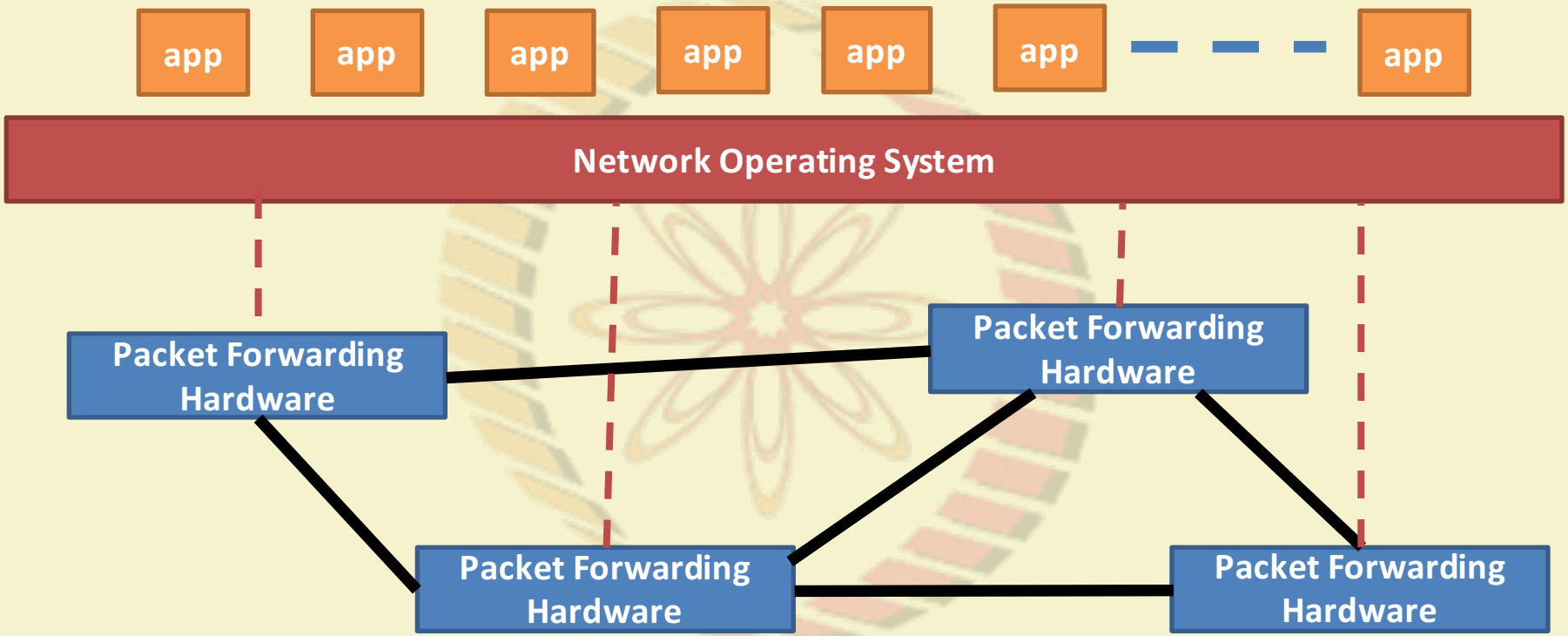
SOUMYA K GHOSH
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

SANDIP CHAKRABORTY
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

Closed Innovations in the Networking Community



SDN Based Approach to Open the Innovation



SDN Architecture

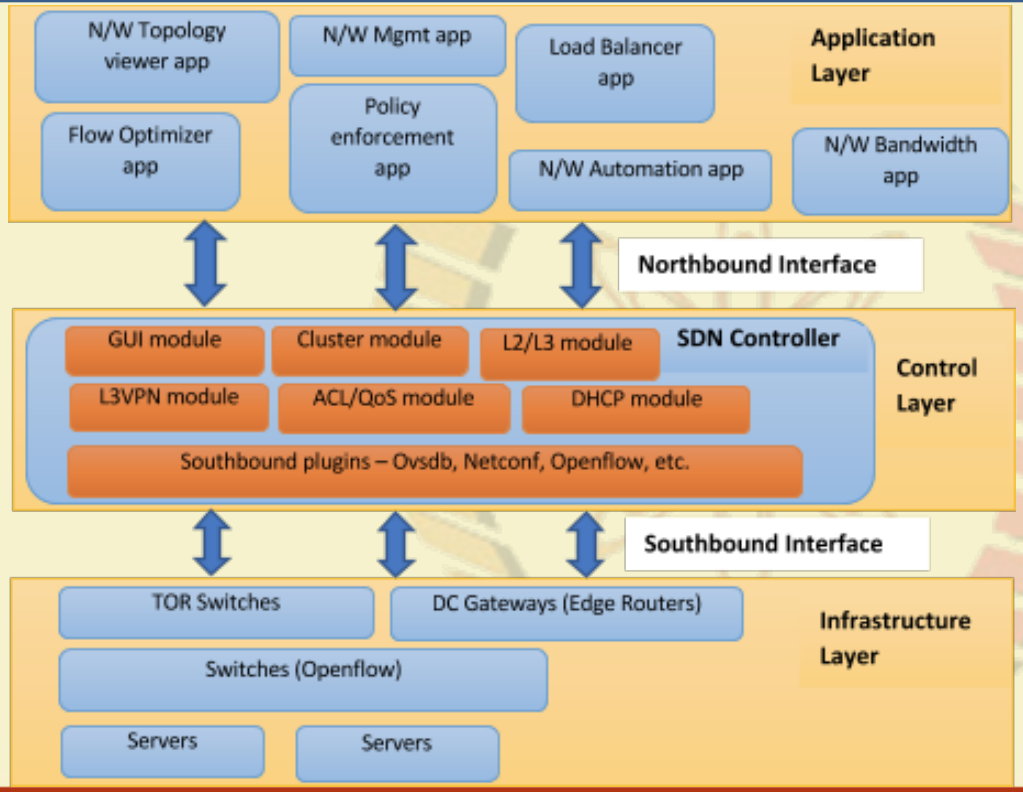
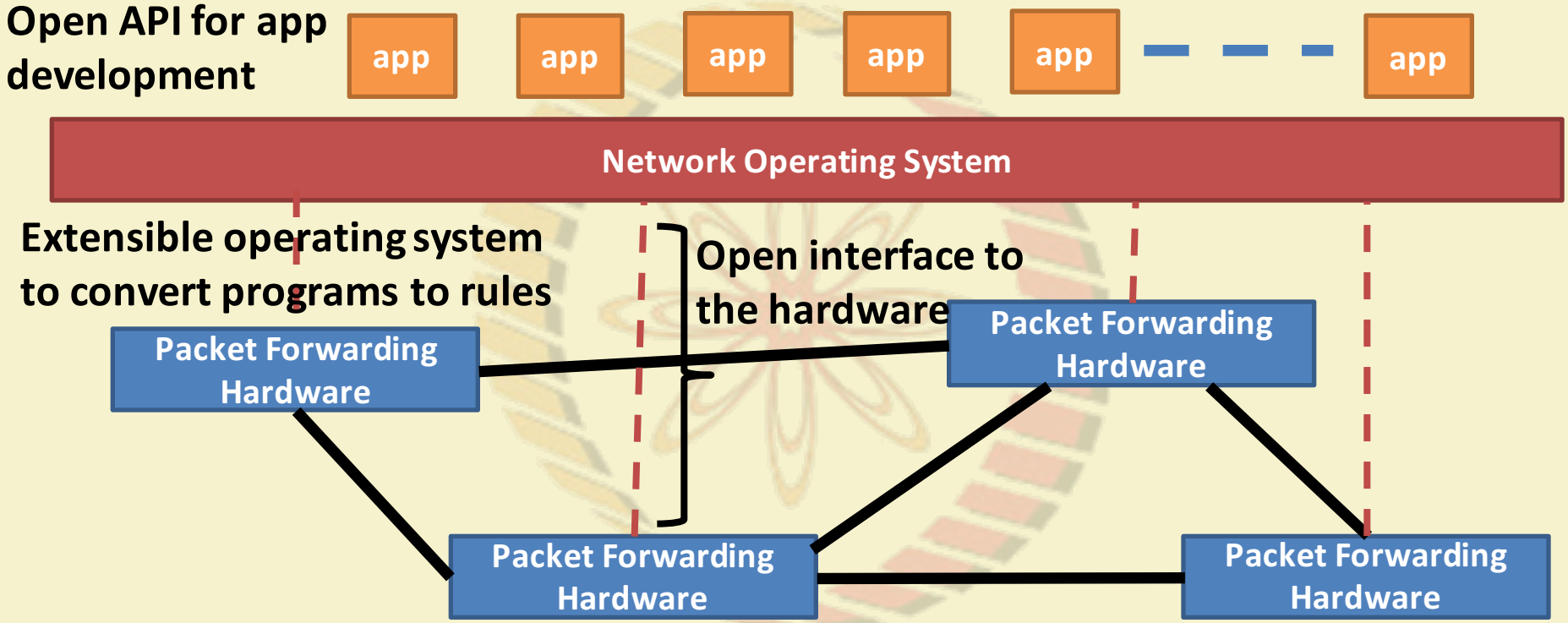


Image source:
https://www.howtoforge.com/tutorial/software-defined-networking-sdn-architecture-and-role-of-openflow/

Interfacing Between Control and Data Planes

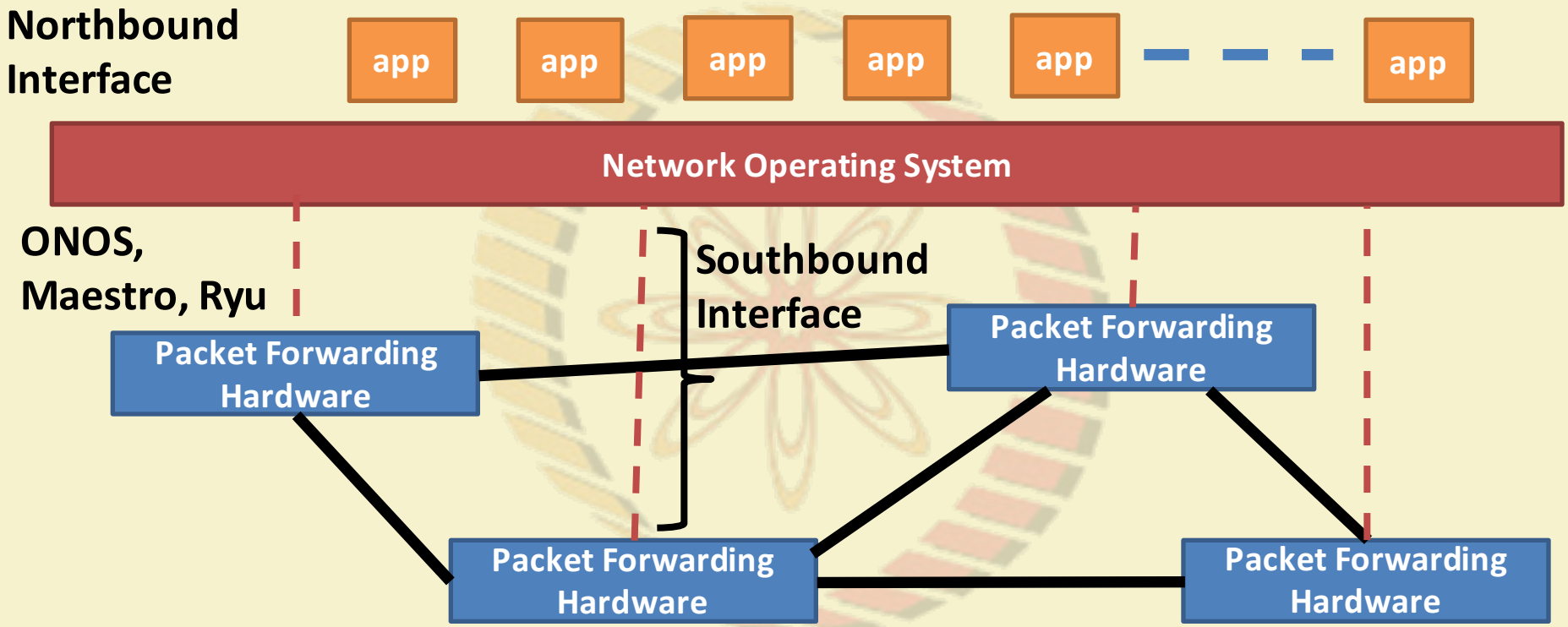


What is OpenFlow

- Protocol for controlling the forwarding behavior of Ethernet switches in a SDN

- Initially released by *Clean Slate Program* at Stanford, specifications now maintained by *Open Networking Forum*

SDN Messaging Interface



SDN Architecture

Programming API in your preferred programming language

ONOS/ Maestro/ Ryu/ OpenDayLight

OpenFlow - open interfacing with hardware

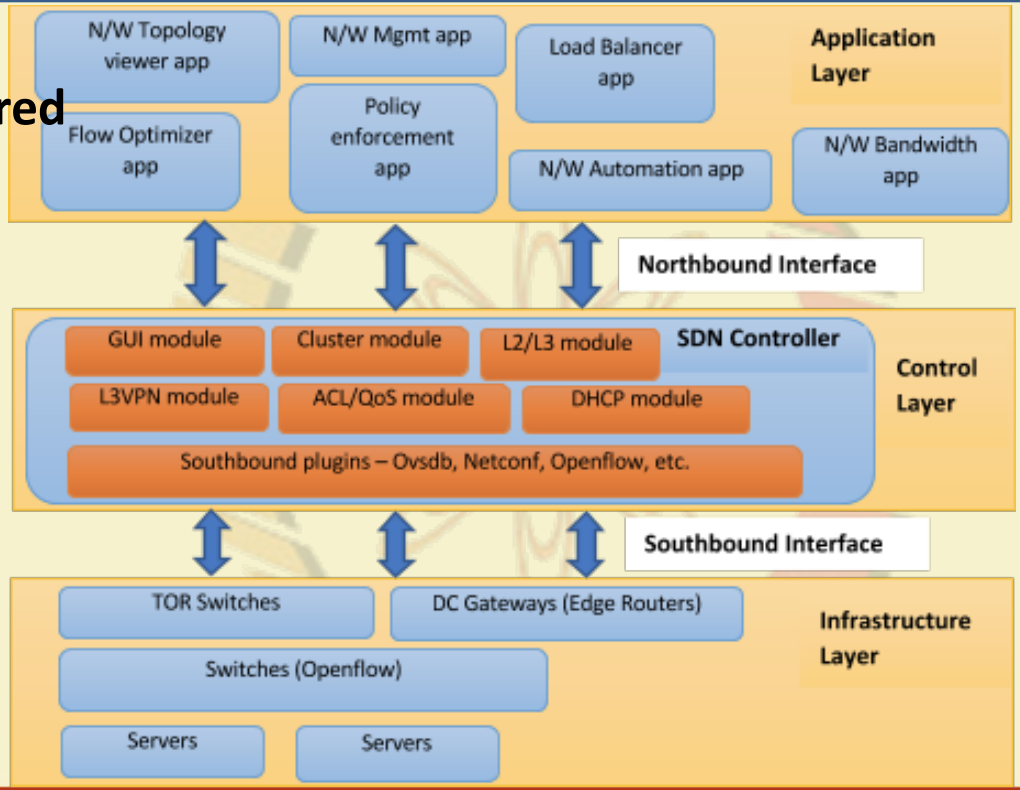


Image source: https://www.howtoforge.com/tutorial/software-defined-networking-sdn-architecture-and-role-of-openflow/

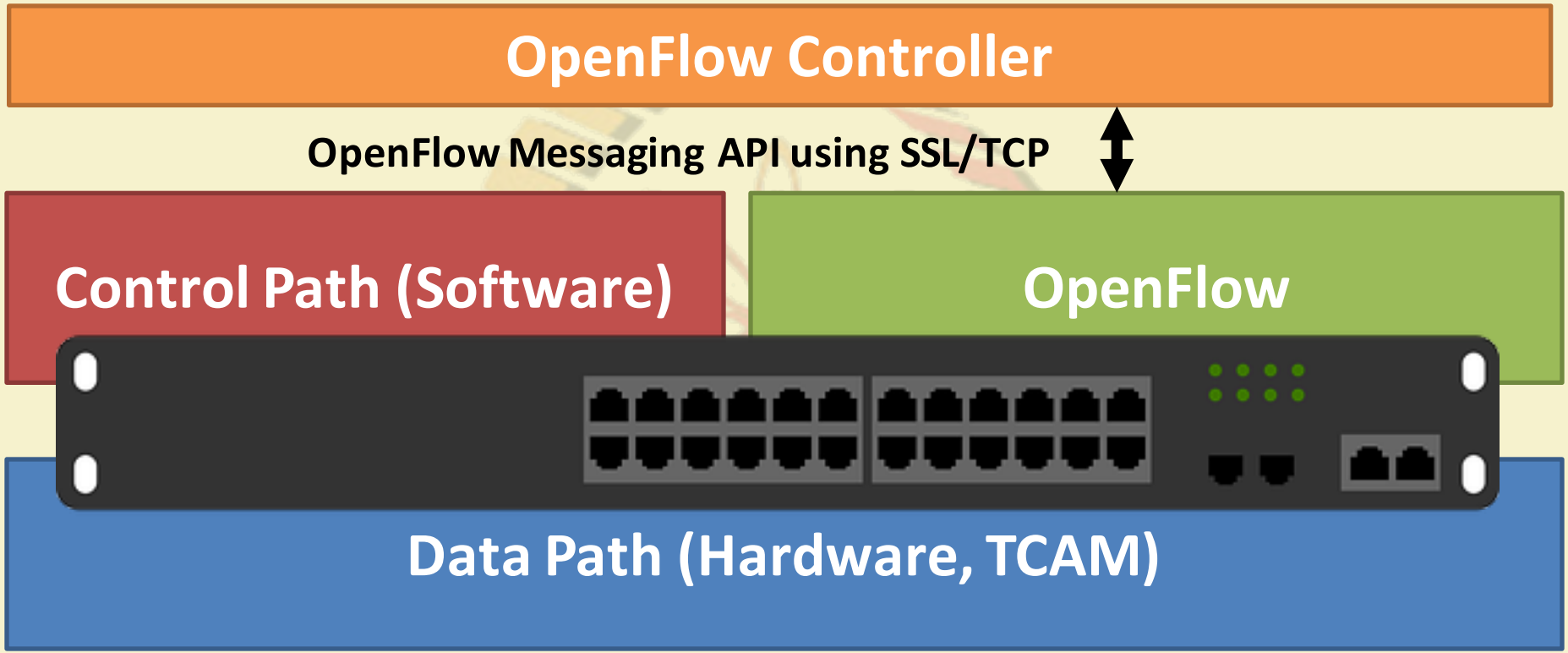
How OpenFlow Works

How OpenFlow Works

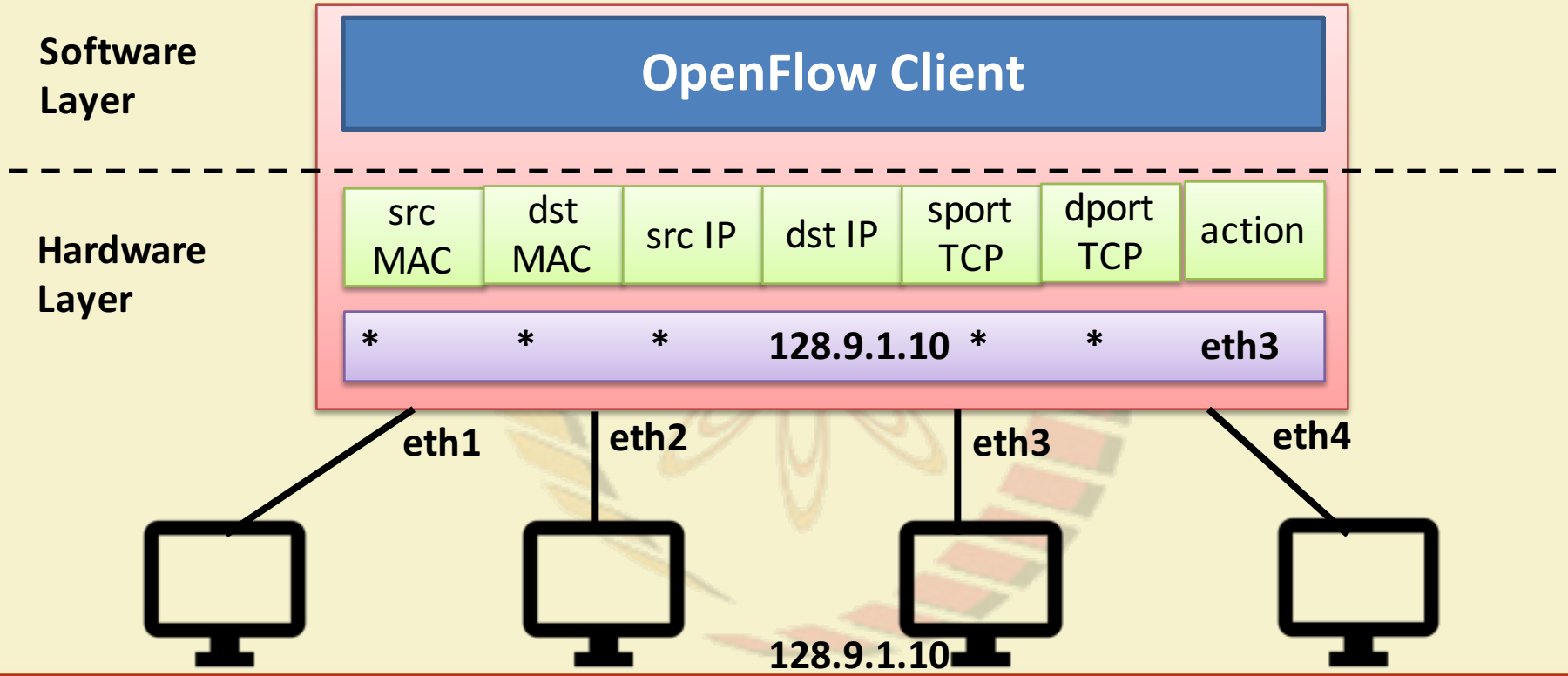
Control Path (Software)

Data Path (Hardware, TCAM)

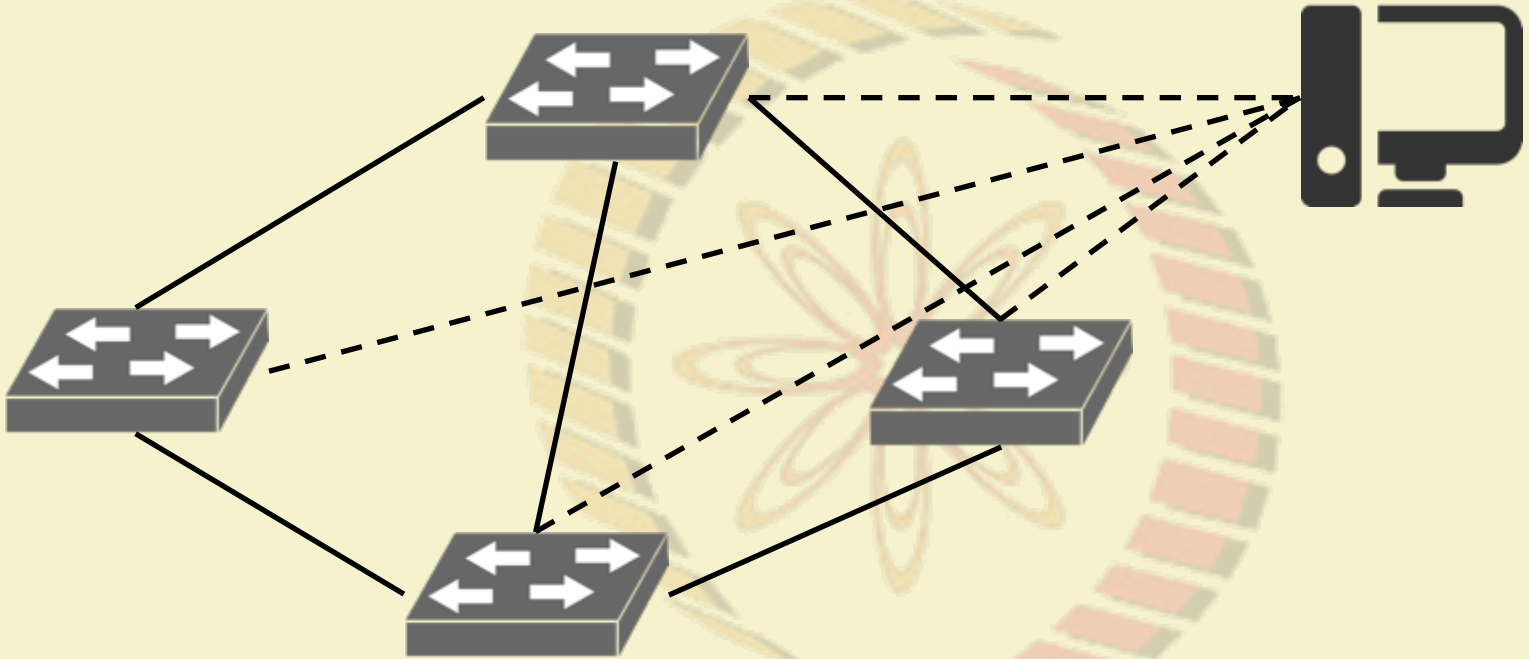
How OpenFlow Works



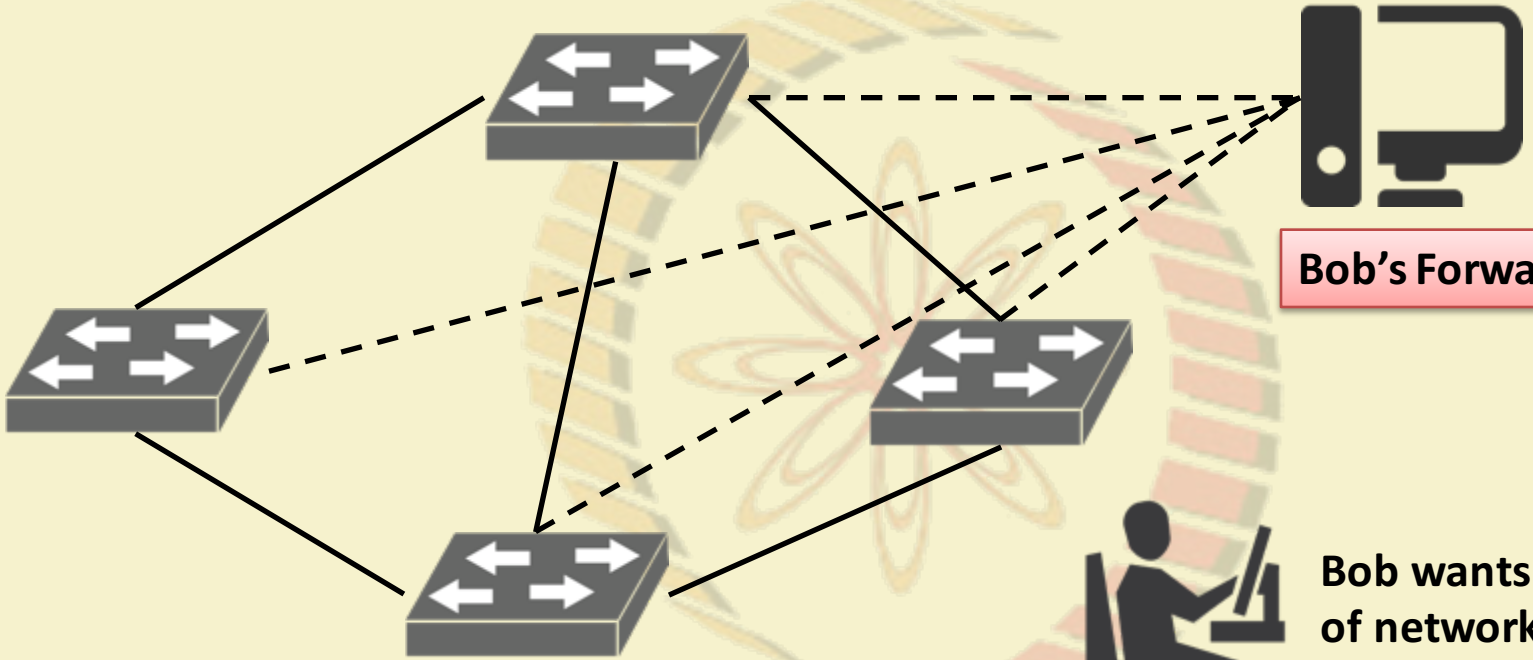
OpenFlow Example



Power of OpenFlow



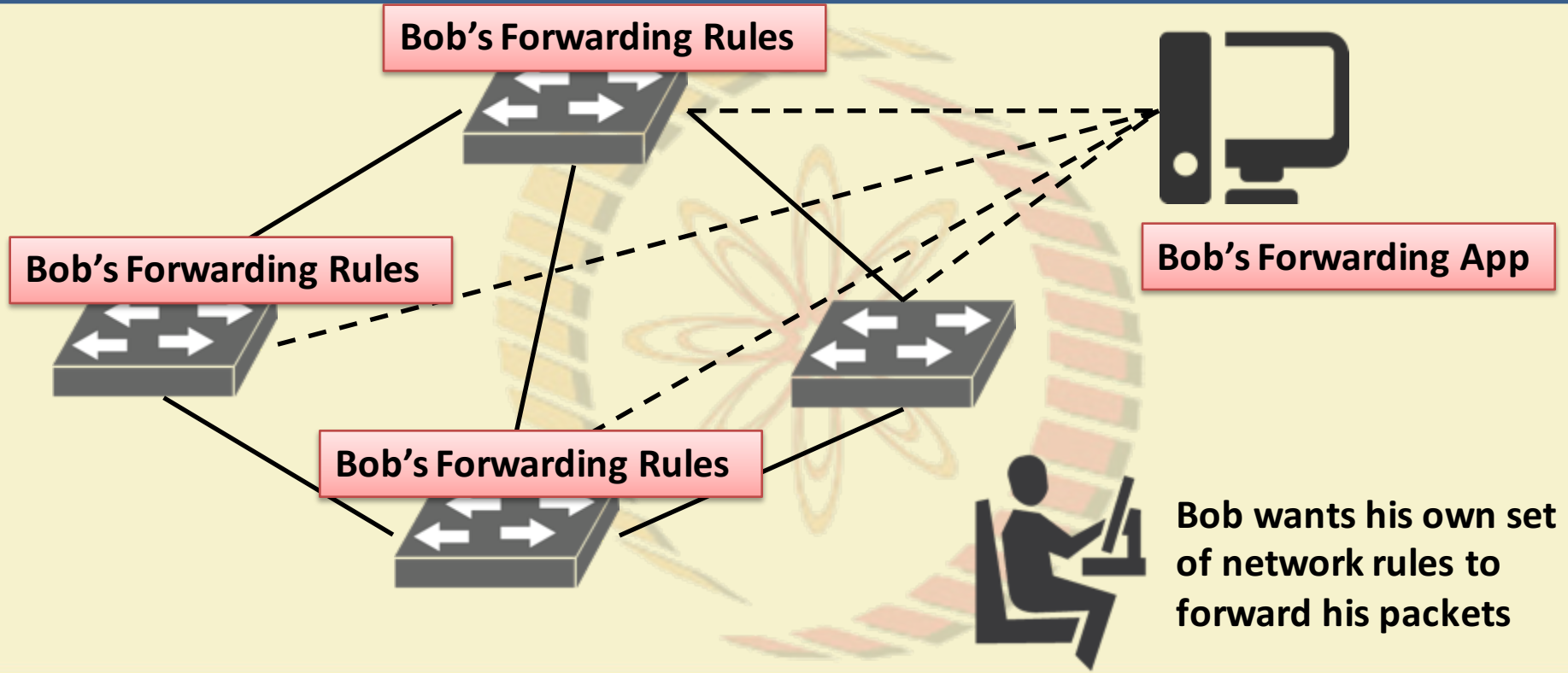
Power of OpenFlow



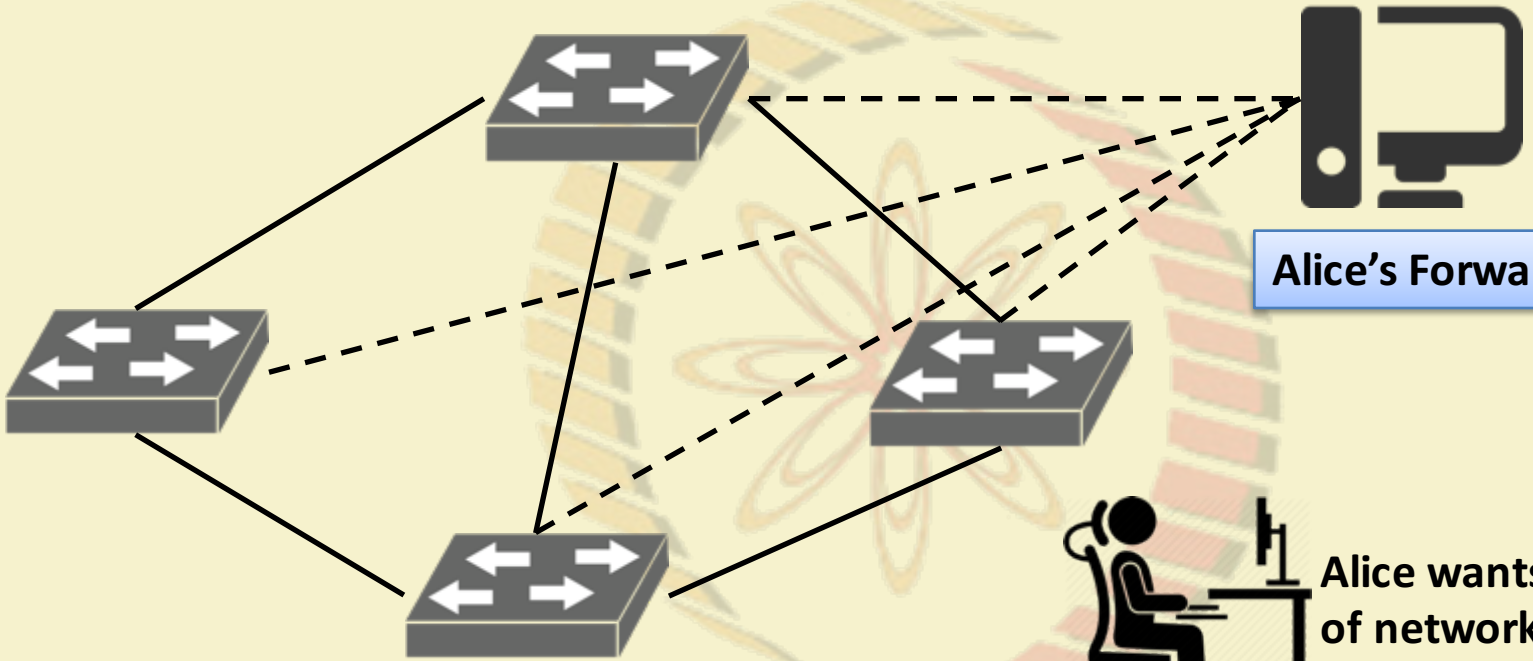
Bob's Forwarding App

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

Power of OpenFlow



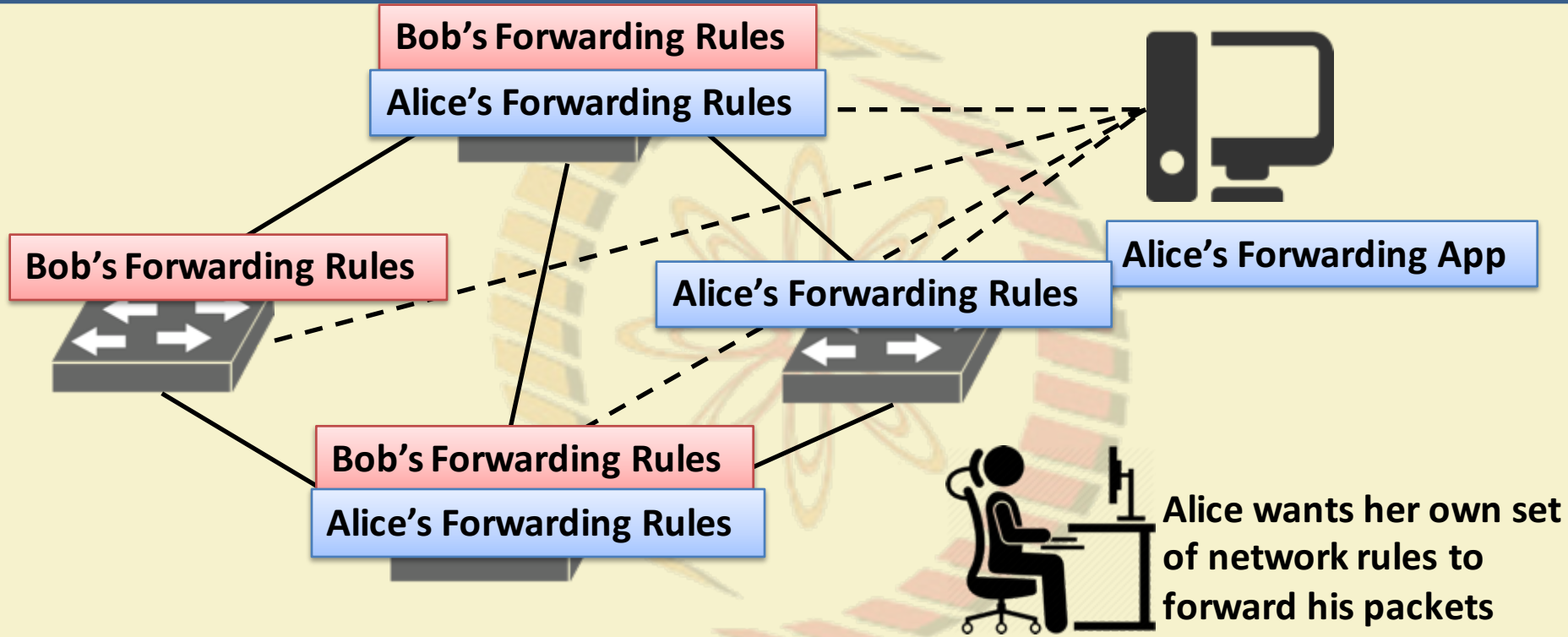
Power of OpenFlow



Alice's Forwarding App

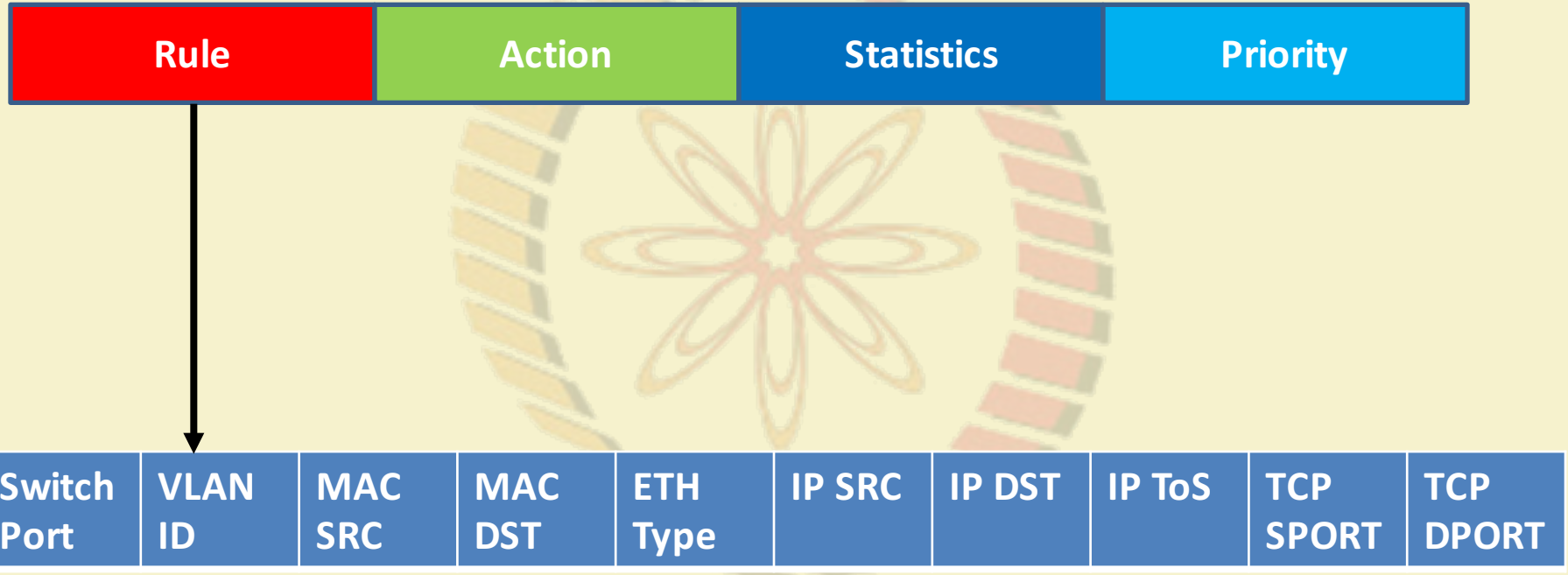
Alice wants her own set of network rules to forward his packets

Power of OpenFlow

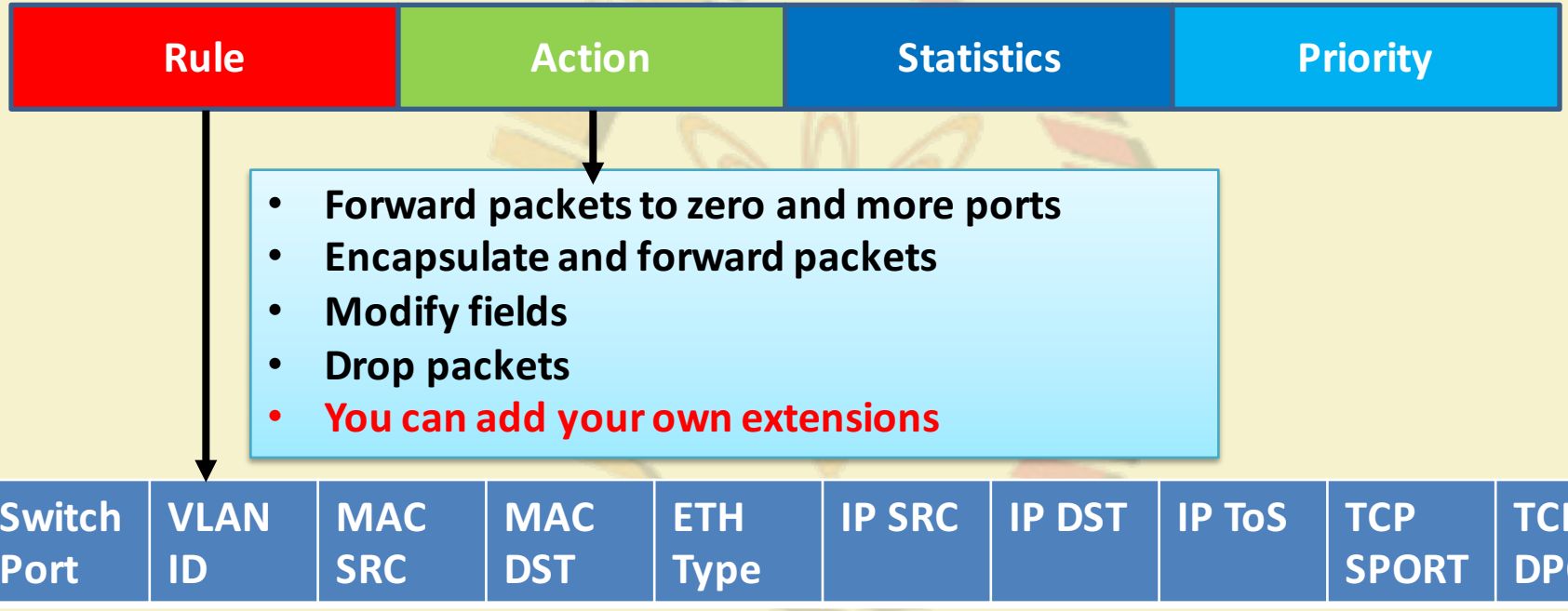


OpenFlow Flow Table

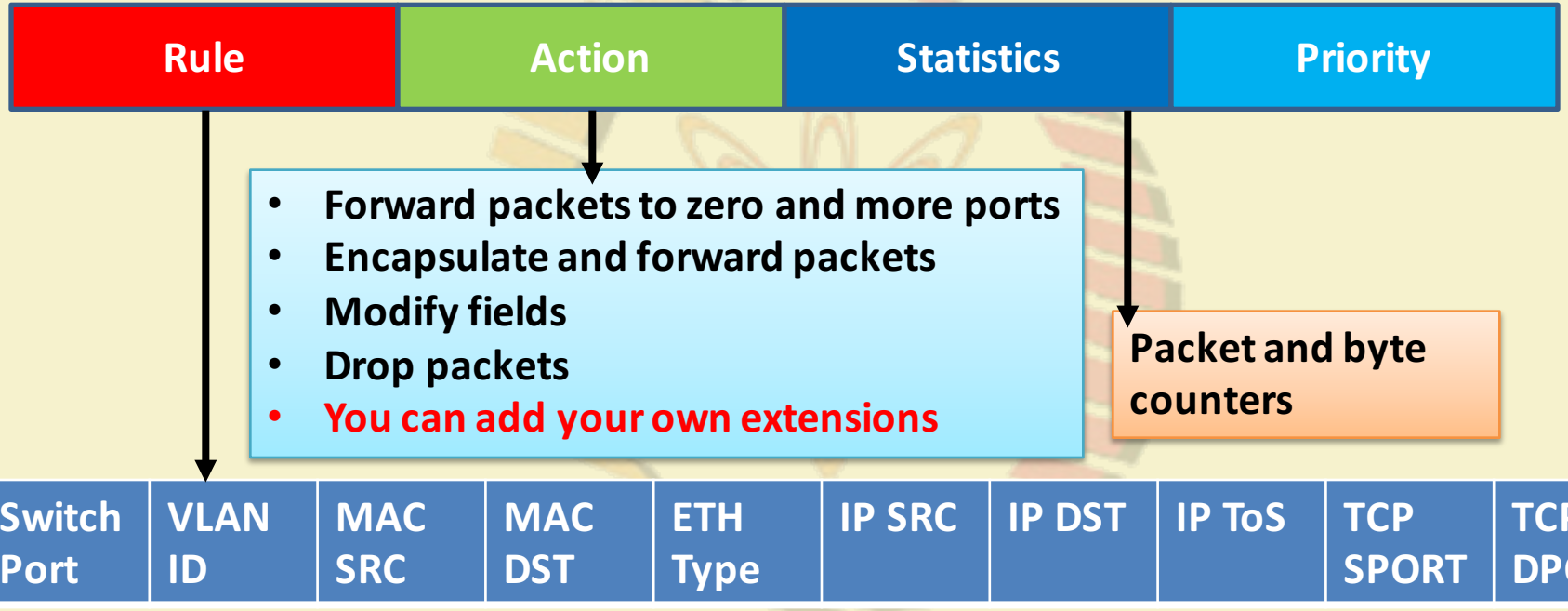
OpenFlow Flow Table



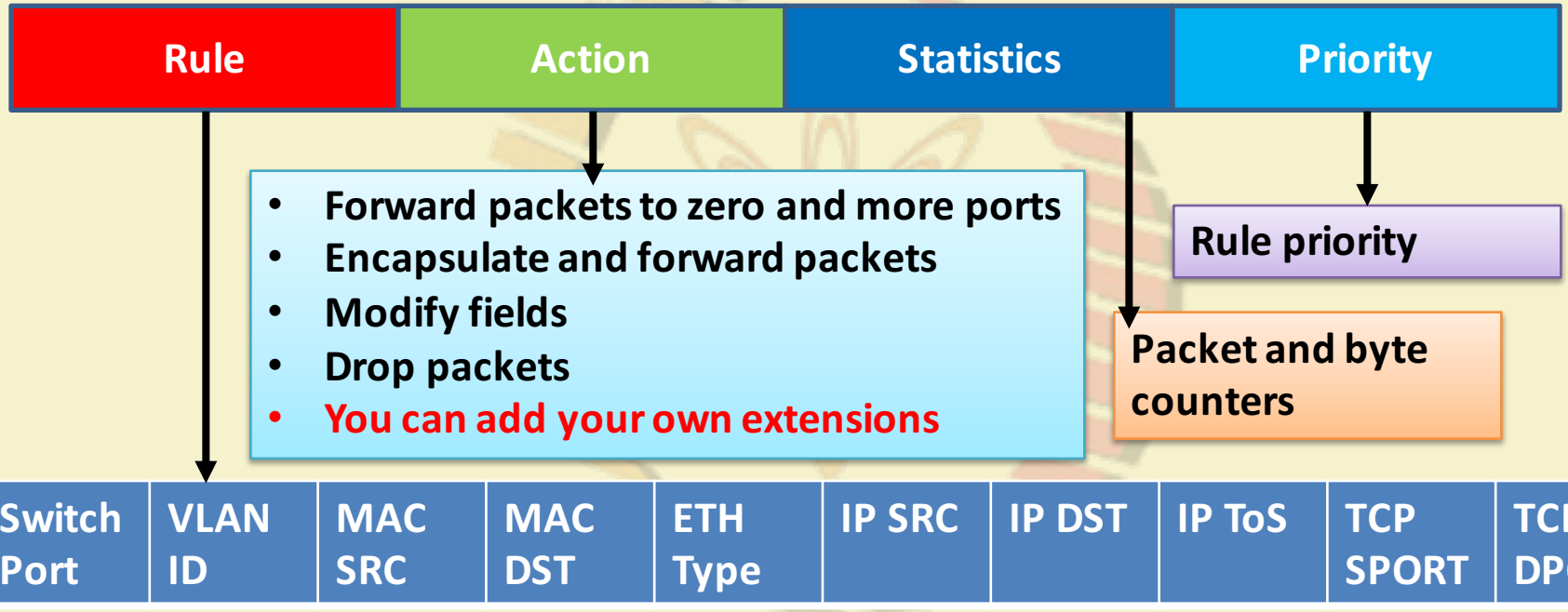
OpenFlow Flow Table



OpenFlow Flow Table



OpenFlow Flow Table



Examples of OpenFlow Flow Tables

- **Switching**

Switch Port	VLAN ID	MAC SRC	MAC DST	ETH Type	IP SRC	IP DST	IP ToS	TCP SPORT	TCP DPORT	Action
*	*	*	12:3F:.	*	*	*	*	*	*	eth2

- **Firewall**

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

Examples of OpenFlow Flow Tables

- **Forwarding**

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

- **Flow Switching**

Switch Port	VLAN ID	MAC SRC	MAC DST	ETH Type	IP SRC	IP DST	IP ToS	TCP SPORT	TCP DPORT	Action
*	*	00:1F:...	14:B2:..	0800	202.1.*.*	212.19.*.*	*	80	8080	eth2

Examples of OpenFlow Flow Tables

- **Source Routing**

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

OpenFlow Messages

- Once TCP connection is made, *OpenFlow HELLO* message is exchanged between the controller and the switch
  - Negotiate OpenFlow version
  - Agree on highest version supported.
- Controller sends *OpenFlow Feature Request* message
  - Get the datapath ID of the switch
  - Determine what features switch supports
- Based on the application program, OpenFlow messages for switch configurations (flow entries) are sent

OpenFlow Messages

- To check connection aliveness, *echo request* and *echo reply* OpenFlow messages are sent from either controller or switch.
- To group flow entries, groups are configured by the controller through group messages that can be stored into group tables inside switches
- To get statistics details from the switch, OpenFlow messages like *flow stats, port stats, queue stats, group stats, table stats,* etc. can be sent from controller.
- Asynchronous OpenFlow messages like *flow rule removal from a switch*, *configuration apply fail error from switch, port up/down status from switch*, etc. can be sent from switch to update controller.

Packet Processing

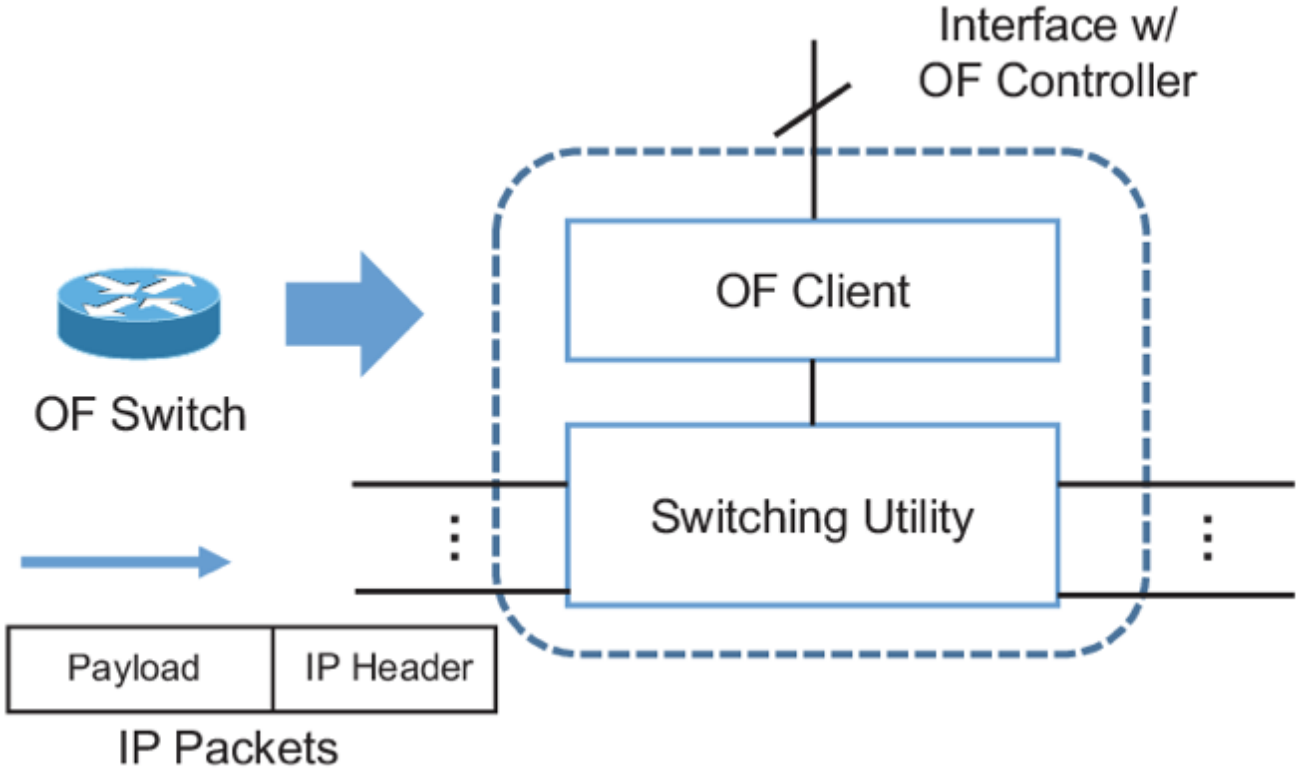


Image Source:
https://www.researchgate.net/figure/Structure-of-the-OpenFlow-switch_fig3_284725082

Pointers to Explore

- Open Networking Foundations: https://www.opennetworking.org/
- OpenFlow specification (version 1.5.1) https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf
- ONOS: https://onosproject.org/
- Ryu SDN Framework: https://osrg.github.io/ryu/


thank you!

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

COMPUTER NETWORKS AND INTERNET PROTOCOLS

SOUMYA K GHOSH
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

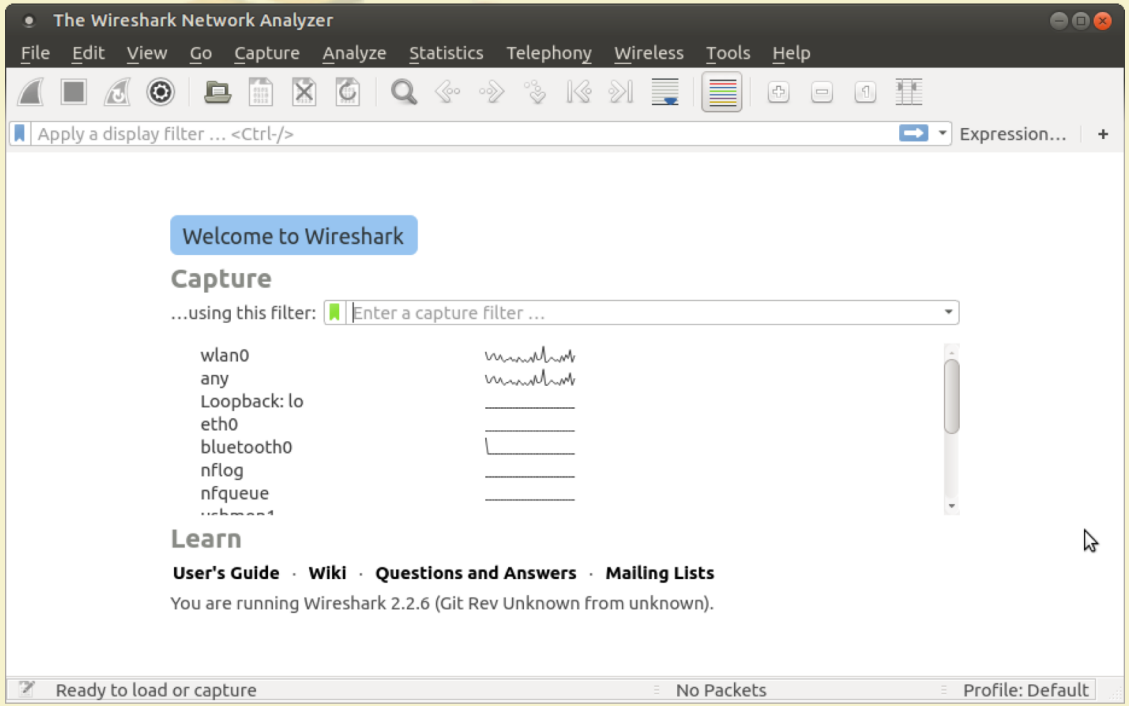
SANDIP CHAKRABORTY
COMPUTER SCIENCE AND ENGINEERING,
IIT KHARAGPUR

Socket programming

- UDP client
- UDP server
- TCP client
- TCP server
- TCP server with fork
- TCP server with select

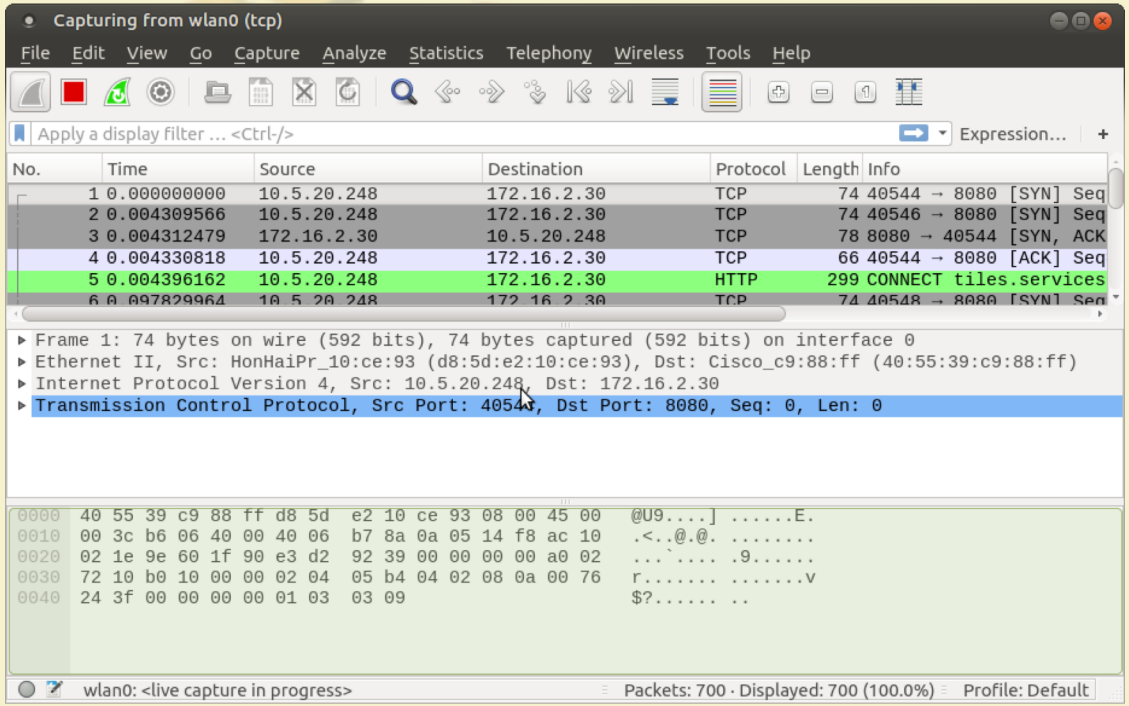
Traffic analysis with wireshark

Wireshark home.



Traffic analysis with wireshark

Wireshark capture mode



Computer Network Learning

- Best way to learn computer network is experimenting on existing network
  - May not be available for everyone
  - Have limited access
  - Expensive to setup

Computer Network Learning

- Best way to learn computer network is experimenting on existing network
  - May not be available for everyone
  - Have limited access
  - Expensive to setup

- Emulated network in a computer
  - Independent of existing network
  - Can be setup as required

Network in a Computer

- Part of computer networks in physical network

Routers

Switches

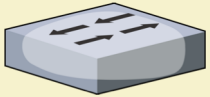
Hosts/servers

Link

Network in a Computer

- We can emulate these components using Mininet

Virtual namespace for legacy network,
Open vSwitch for Software driven network



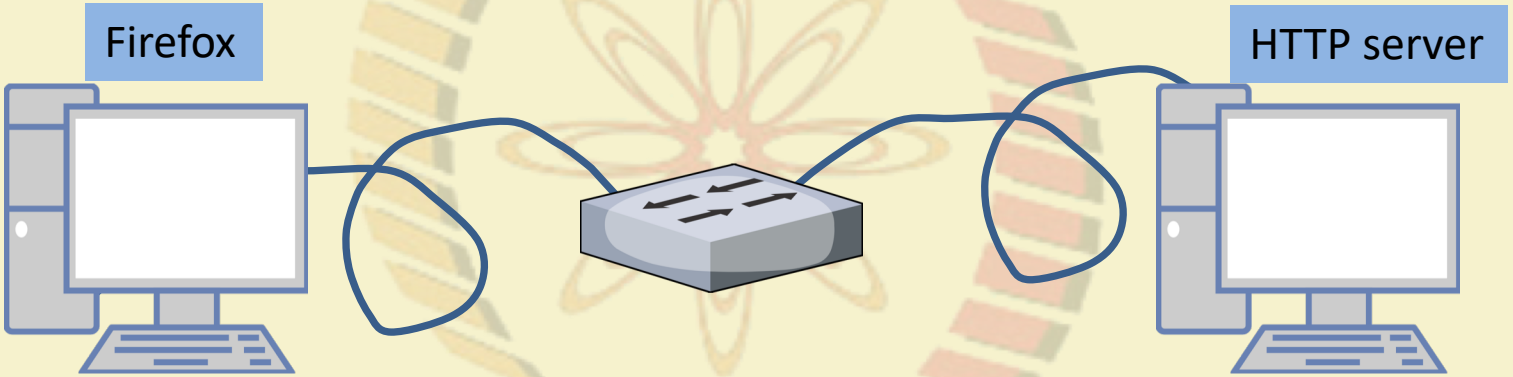
Open vSwitch

Virtual namespace

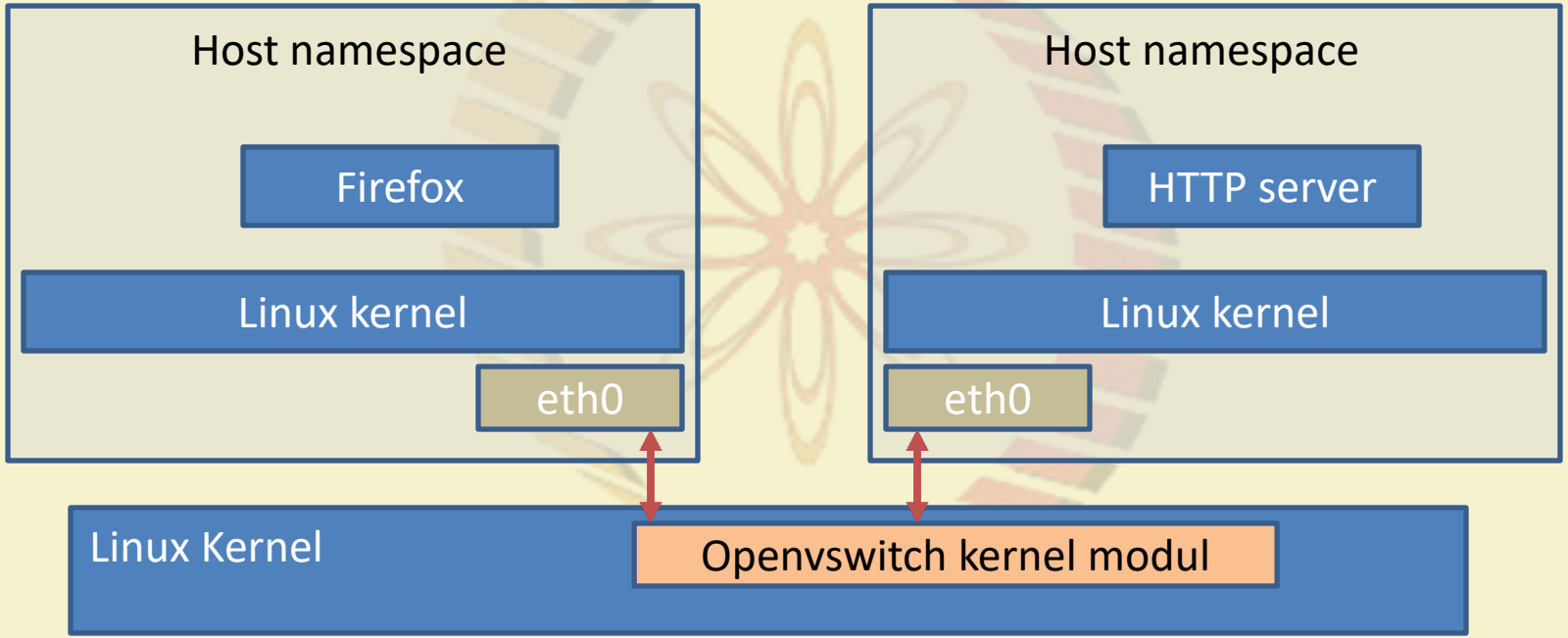
Virtual link

Network in a computer

- Simple computer network (Physical)

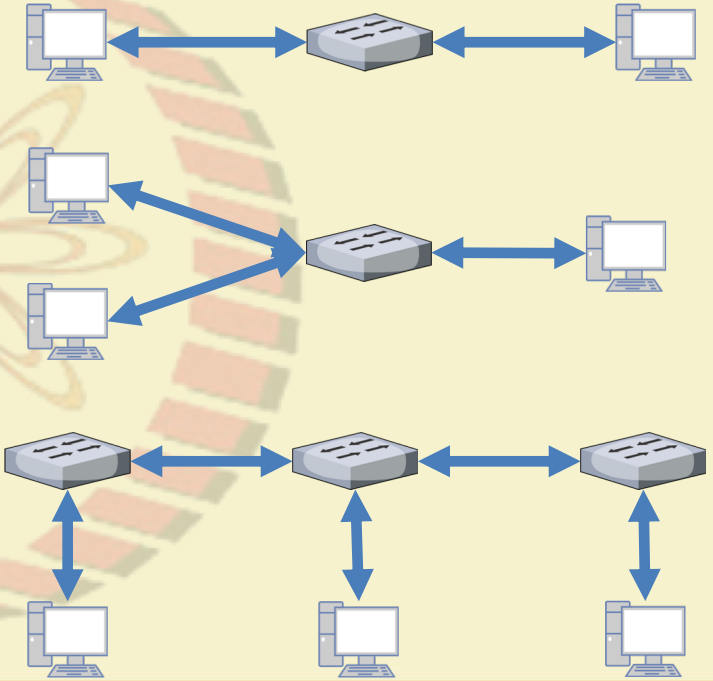


Network in a computer

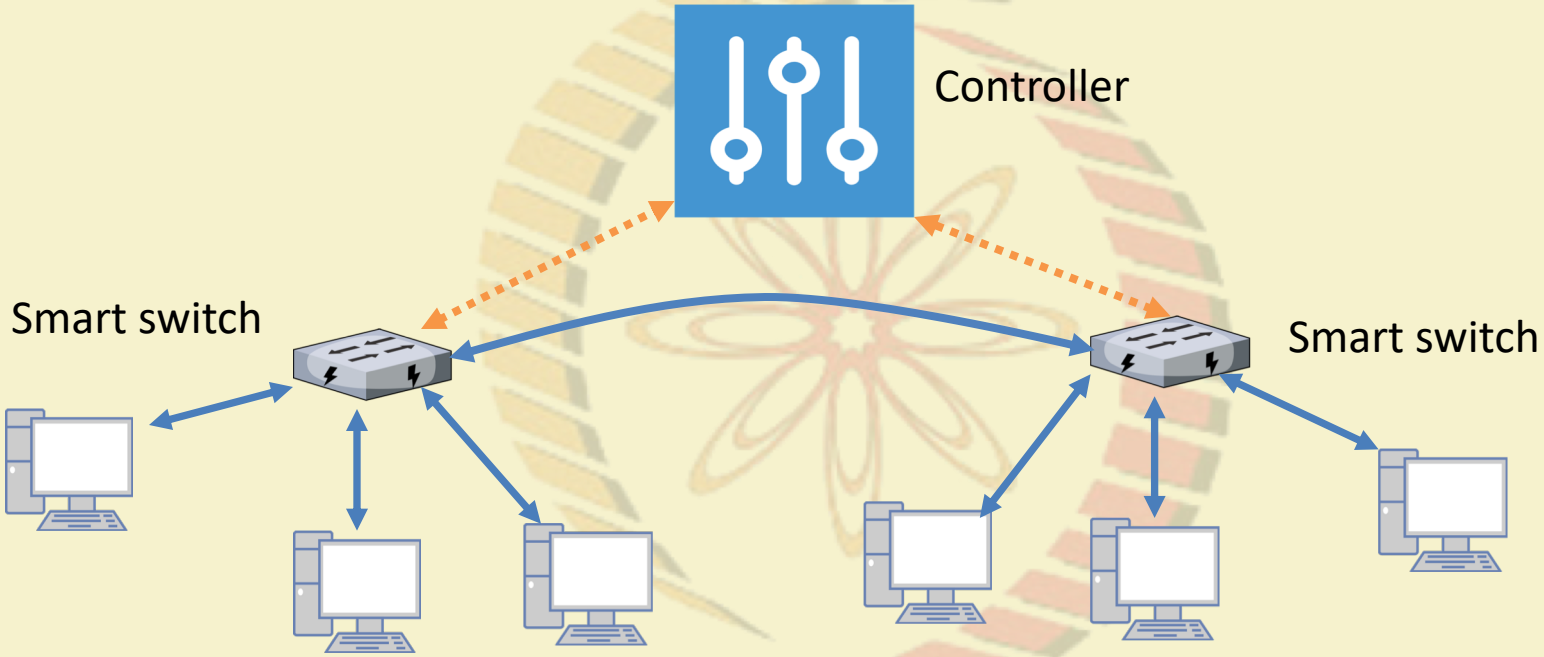


Network in a computer

- Use Mininet to create the network
  - `$ mn --topo single,2`
  - `$ mn --topo single,3`
  - `$ mn --topo linear,3`



Network with controller in a computer



– $ mn --topo linear,2,3 --controller=remote


thank you!