

Software Defined Network (SDN) with MikroTik

Mr. Narong TOHKU (CEO)

Cherry Network and Solutions Co.,Ltd.



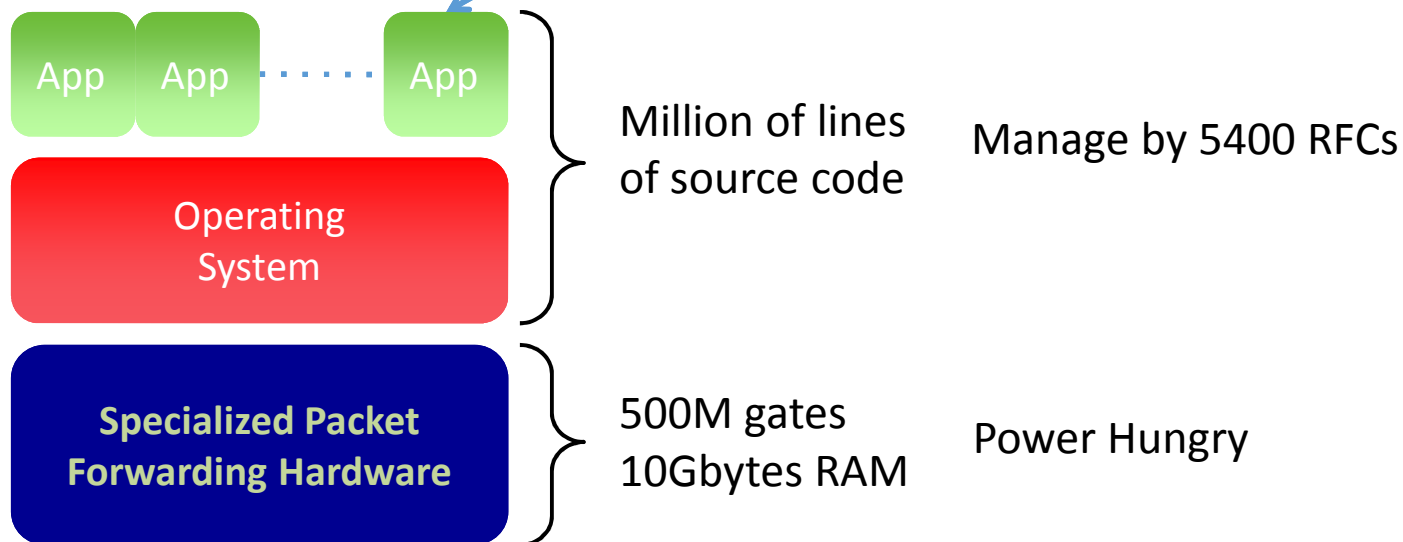
Agenda

- Introduction.
- What is Software-Defined Network?
- OpenFlow Overview.
- OpenDaylight
- OpenFlow Implementation in MikroTik

The Networking Industry (2007)



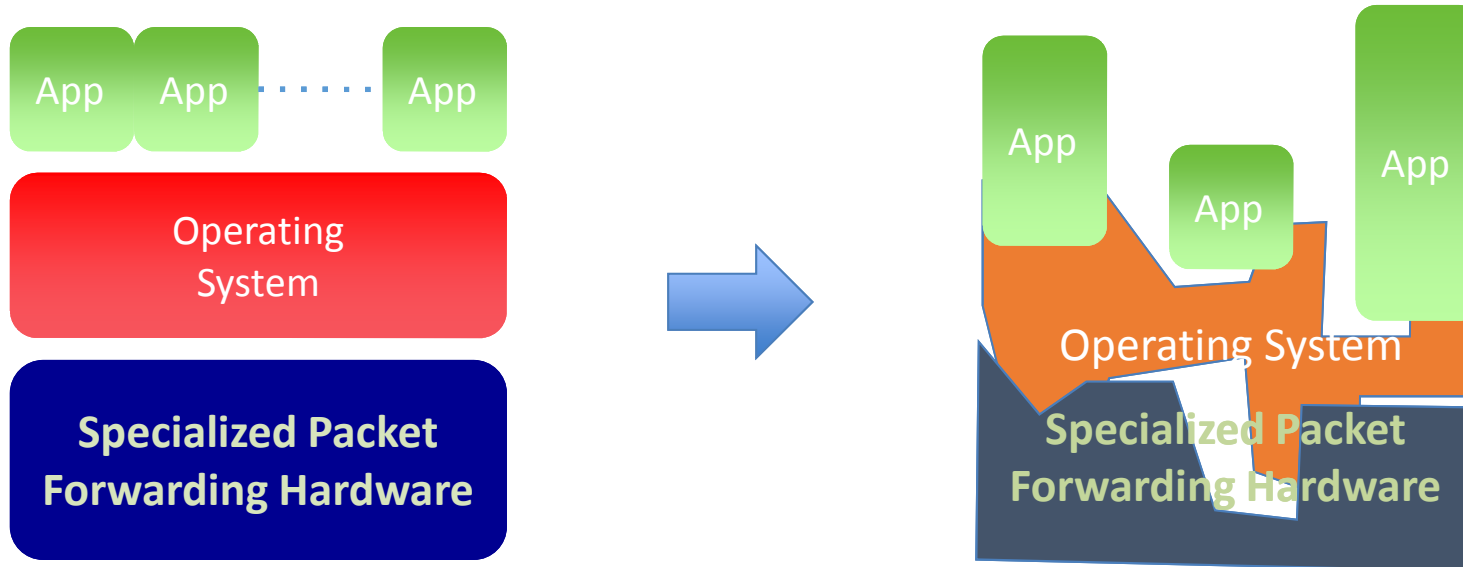
Routing, management, mobility management, access control, VPNs, ...



Many complex functions baked into the infrastructure

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

Reality...!!!! (As 2015)



Used equipment

- Software bundled with hardware.
- Vendor-specific interfaces.

er specified : Slow protocol standardization.

w people can innovate

- Equipment vendors write the code.
- Long delays to introduce new features.

Operating a network is expensive

- More than half the cost of a network.
- Yet, operator error causes most outages.



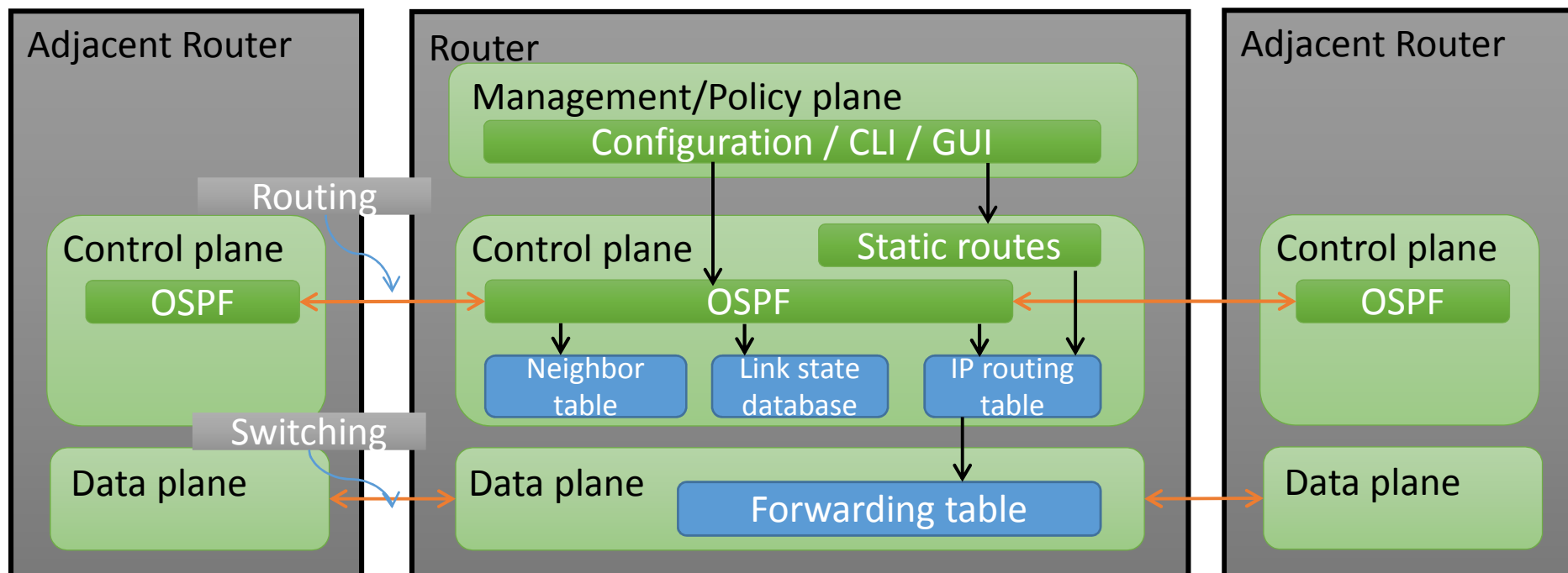
Buggy software in the equipment

- Routers with 20+ million lines of code
- Cascading failures, vulnerabilities, etc.



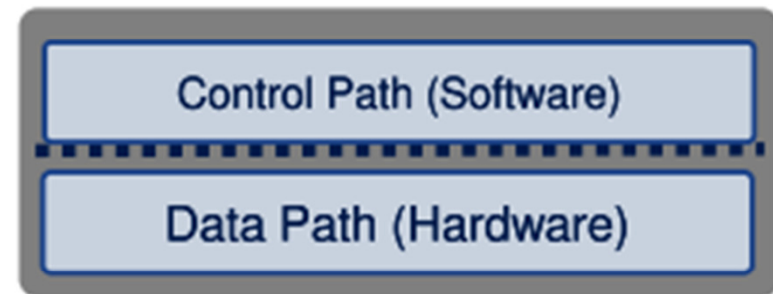
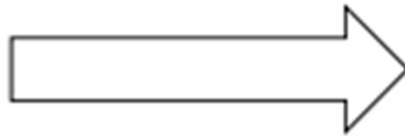
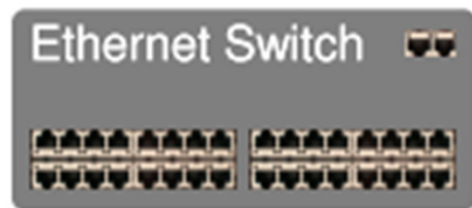
Traditional Network Router

- Router can be partitioned into **control** and **data plane**
 - Management plane/ configuration
 - Control plane / Decision: OSPF (Open Shortest Path First)
 - Data plane / Forwarding



Traditional network Router In Summary

- Typical Networking Software
 - Management plane
 - Control Plane – The brain/decision maker
 - Data Plane – Packet forwarder



Imagine IF The Network is.....!!!

Control Plane

SDN Concept
Separate Control
plane and Data
plane.

Separated

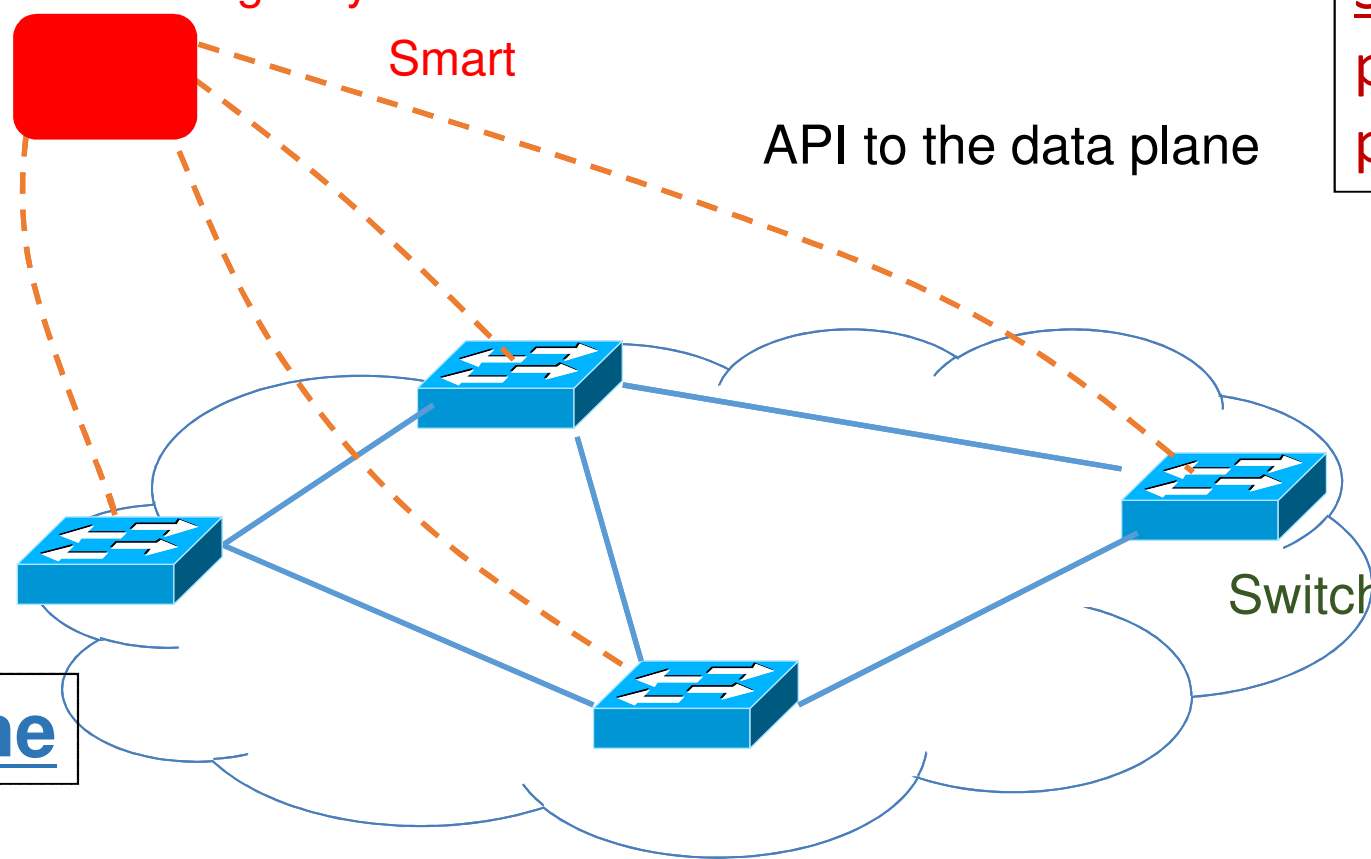
Logically-centralized control
Smart

API to the data plane

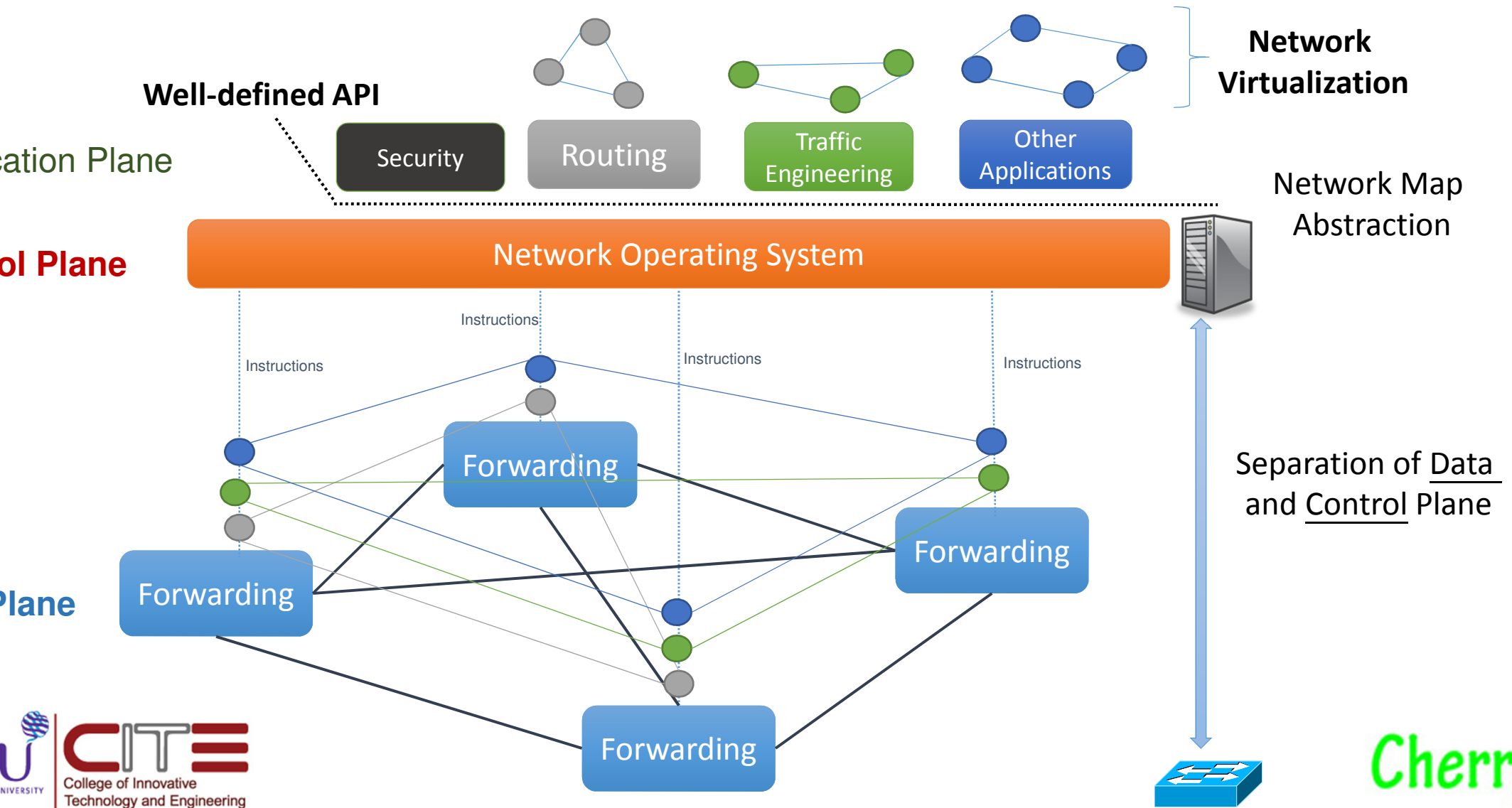
Dumb,
fast

Switches

Data Plane



Software-Defined Network with key Abstractions



SDN Basic Concept

- **Separate Control plane and Data plane entities.**
 - Network intelligence and state are logically centralized.
 - The underlying network infrastructure is abstracted from the applications.
- **Execute or run Control plane software on general purpose hardware.**
 - Decouple from specific networking hardware.
 - Use commodity servers and switches.
- **Have programmable data planes.**
 - Maintain, control and program data plane state from a central entity.
- **An architecture to control not just a networking device but an entire network.**

The Origin of SDN

Martin Casado



2006: Martin Casado, a PhD student at Stanford and team propose a clean-slate security architecture (SANE) which defines a centralized control of security (in stead of at the edge as normally done). Ethane generalizes it to all access policies.

The idea of *Software Defined Network* is originated from **OpenFlow** project (ACM SIGCOMM 2008).

2009: Stanford publishes **OpenFlow** V1.0.0 specs.

June 2009: Martin Casado co-founds Nicira.

March 2011: Open Networking Foundation is formed.

Oct 2011: First Open Networking Summit. Many Industries (Juniper, Cisco announced to incorporate.

July 2012: VMware buys Nicira for **\$1.26B**.

Lesson Learned: Imagination is the key to unlock the power of possibilities. **Cherry**

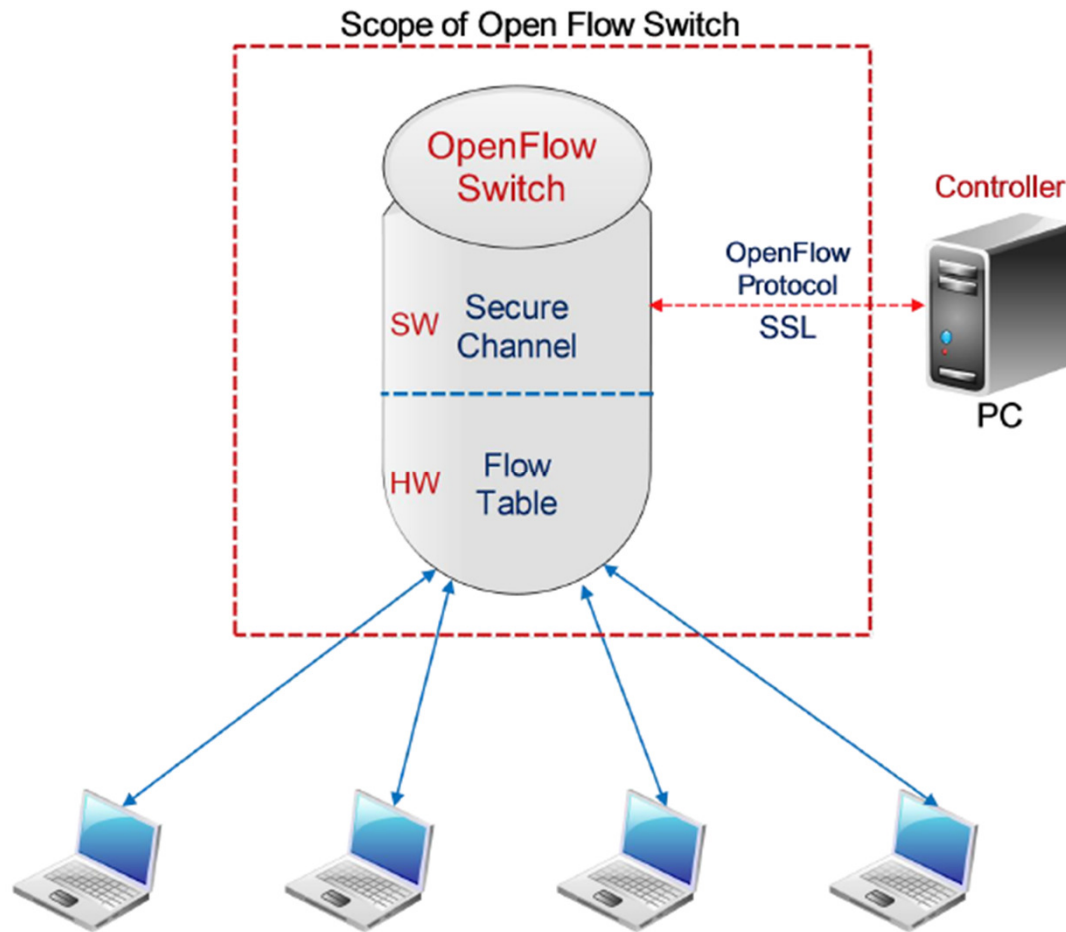
OpenFlow Overview

- OpenFlow is an open standard that enables researchers to run experimental protocols in the campus network we use every day.
- OpenFlow is added as a feature to commercial Ethernet switches, routers and **wireless access points** – and provides standardized hook to allow researchers to run experiments, without requiring to expose the internal workings of their network devices.
- OpenFlow is currently being implemented by major vendors, with OpenFlow-enabled switches now commercially available.

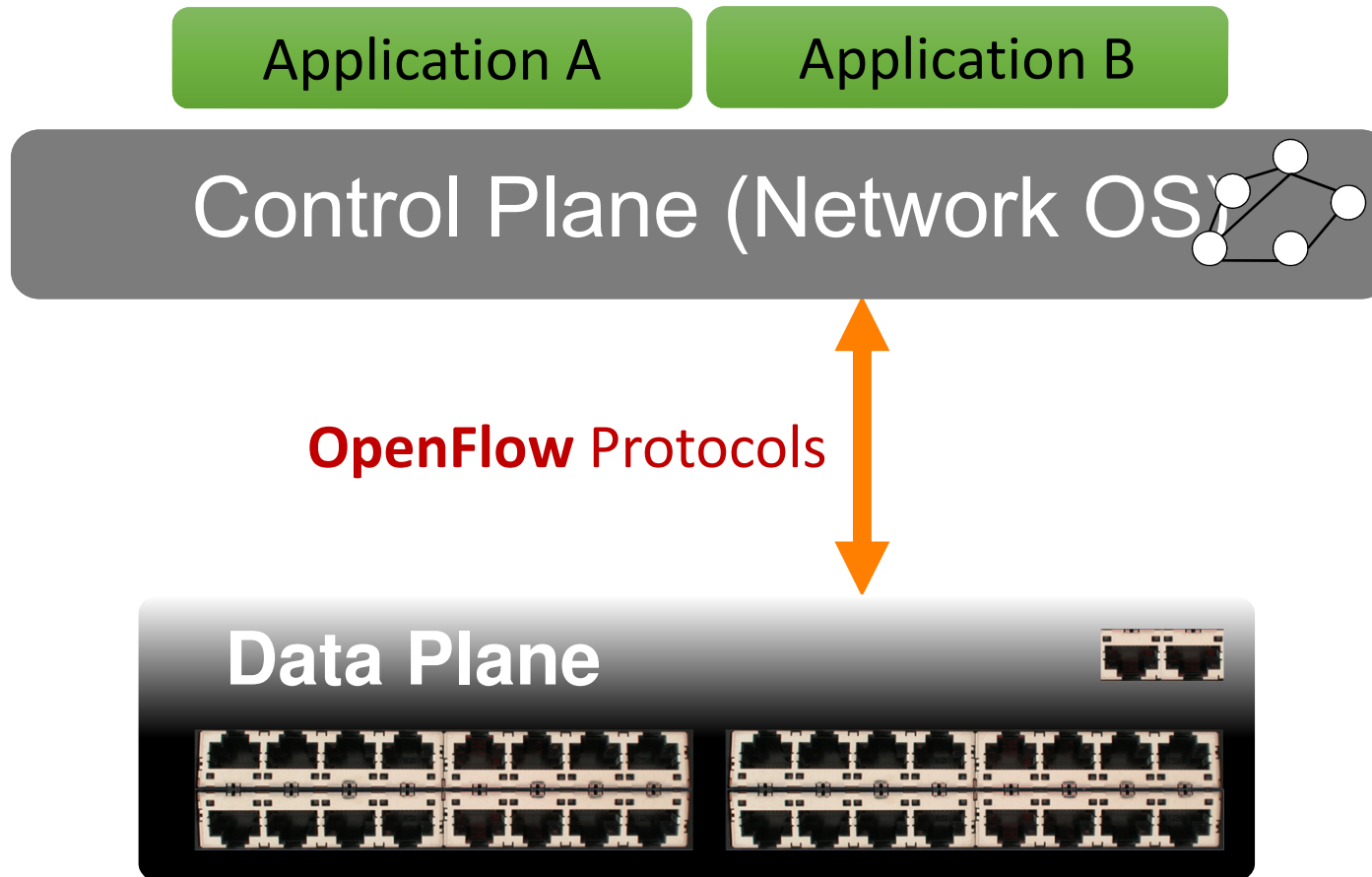
OpenFlow Overview

- What does this really mean, though?
- You start with an OF switch and a controller. The OF switch is a **switch/router/access** point that runs the **OF client**.
- Most switches have a “flow table” built from TCAM. This flow table determines which traffic moves where.

OpenFlow Overview



OpenFlow Overview



OpenFlow Overview

- OpenFlow(OF) is meant to be a means of testing new routing or forwarding methods to build these flow tables.
- It is designed to allow for layer 3 or layer 2 forwarding based on: **port, MAC, VLAN, TCP header, or IP header.**
- The way it accomplishes this is to establish a secure SSH tunnel between the OF switch and the controller.
- When a new flow start, it sends the packets to the controller.
- The controller then builds an entry into the flow table to handle the remainder of this connection.

OpenFlow Overview

- They also have the ability to segregate the OF flow table from that of the standard switch.
- The idea is you can choose a handful of ports on a device and have them participate in the OF test.
- Your controller will then update all of the OF switches in the chain with the new flow information.
- Since this is a separate flow table the device will continue to process all other traffic based on the standard flow table.

OpenFlow Overview

- So long as you can maintain the SSH connectivity to controller you can write your own proprietary mesh algorithm and have the MTKs use it.
- You could write your own algorithm that monitors all of ISP interfaces via SNMP and adjust routing on the fly according to congestion ...really the sky is the limit at this point.
- Right now there are really only two commands in the MikroTik.

Summary : What is OpenFlow?

- Allow separation of control and data planes.
- Centralization of control.
- Flow based control.
- Takes advantage routing tables in Ethernet switches and routers.
- **SDN is not OpenFlow.**
 - **SDN** is a concept of the physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices.
 - **OpenFlow** is communication interface between the control and data plane of an *SDN architecture*.
 - Allows direct access to and manipulation of the forwarding plane of network devices such as switches and routers, both physical and virtual.
 - Think of as a protocol used in switching devices and controllers interface.

How is OpenFlow related to SDN in The Nut Shell?

OpenFlow allows you to do:

Programmability

- Enable innovation/differentiation
- Accelerate new features and services introduction

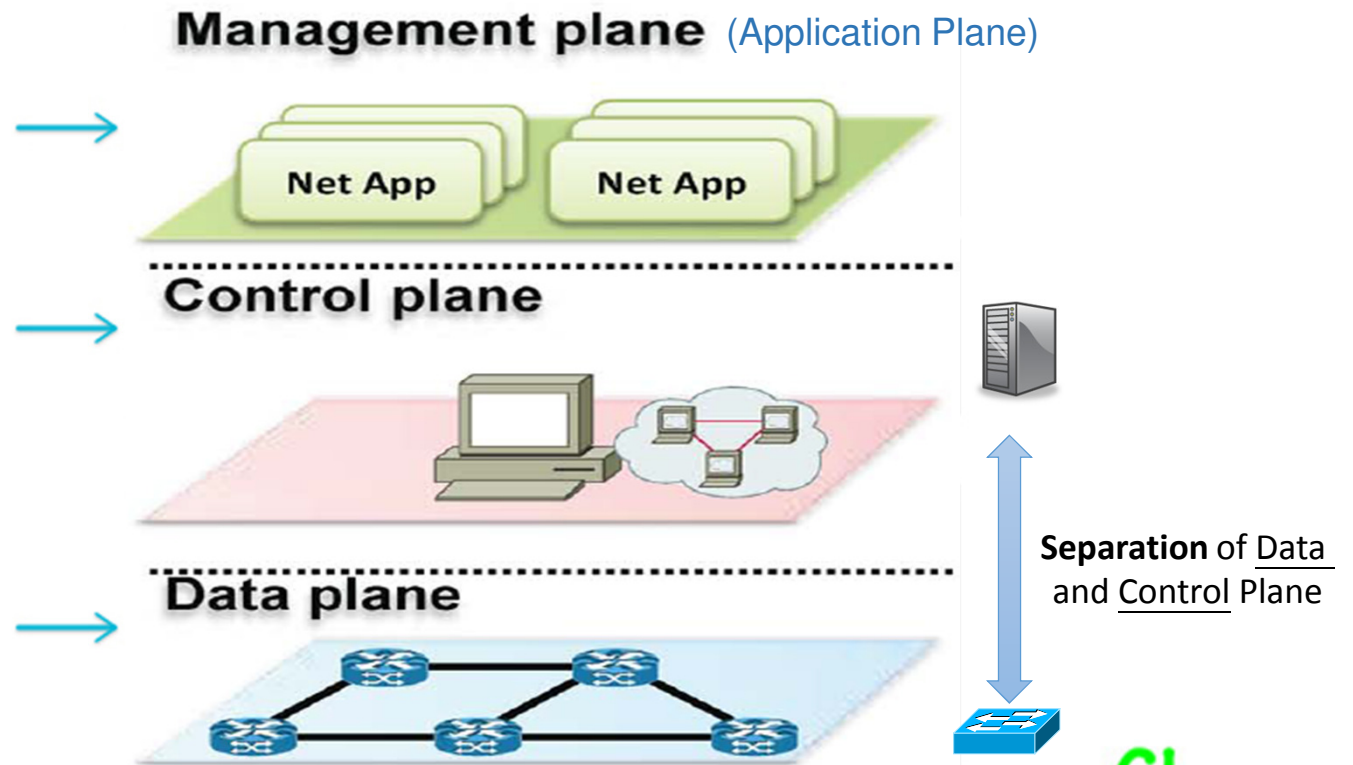
Centralized Intelligence

- Simplify provisioning
- Optimize performance
- Granular policy management

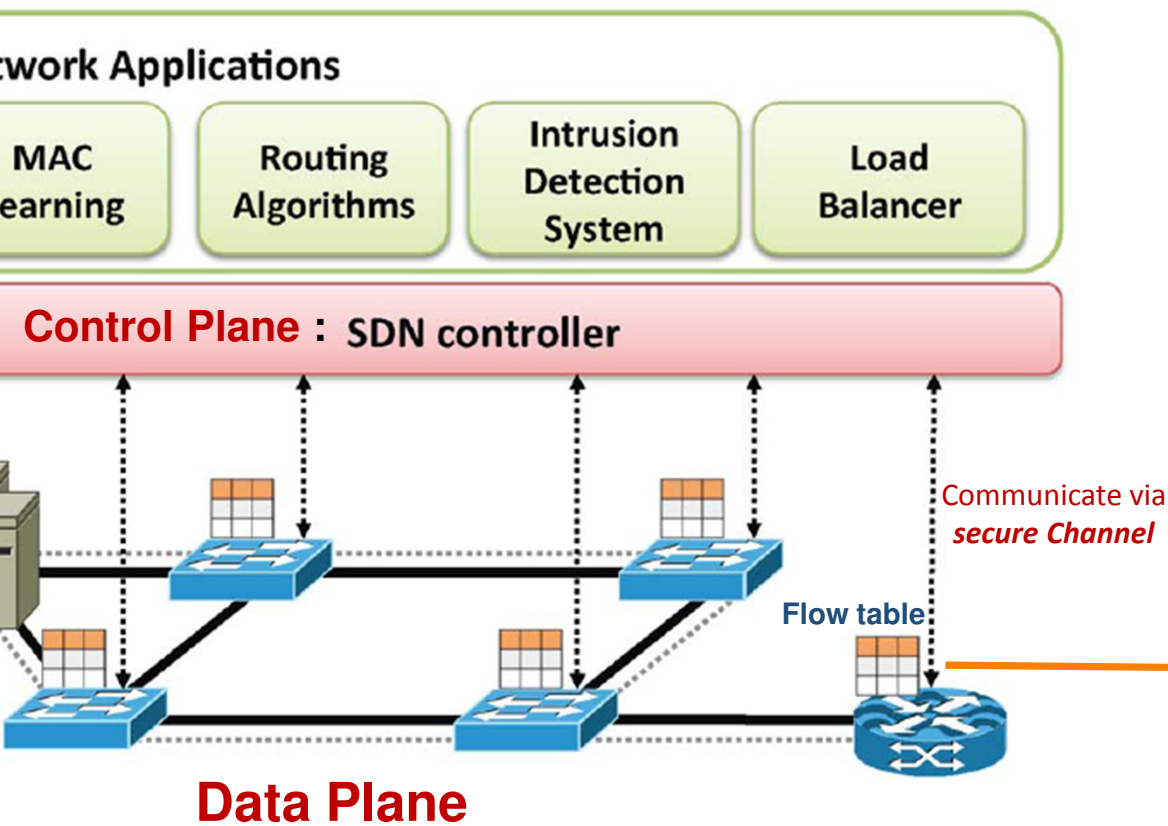
Abstraction

- Decouple:
 - Hardware & Software
 - Control plane & forwarding
 - Physical & logical config.

SDN Concept

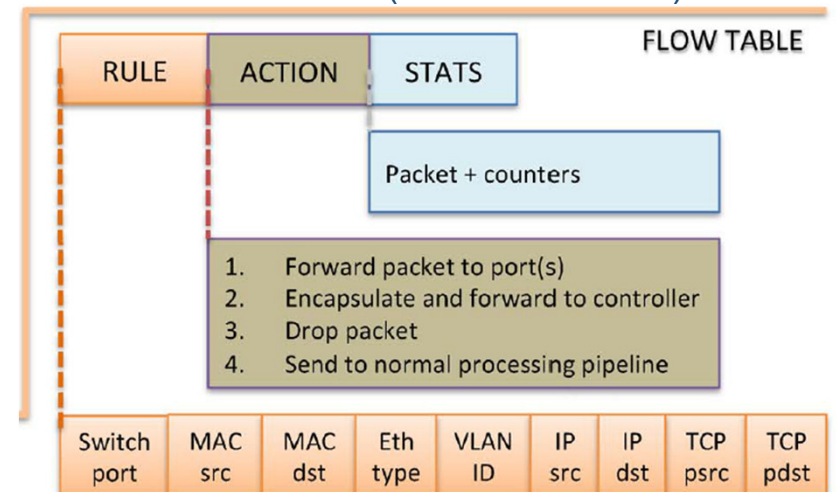


Basic OpenFlow: How Does it Work?



- Controller **manages** the traffic (network flows) by **manipulating** the **flow table** at switches.
 - Instructions are stored in flow tables.
- When packet arrives at switch, match the header fields with flow entries in a flow table.
- If any entry matches, performs indicated actions and update the counters.
- If Does not match, Switch asks controller by sending a message with the packet header.

Flow Table (has 3 sections)



Match the packet header

The Actual Flow Table Looks Like

Port	Src MAC	Dst MAC	VLAN ID	Priority	EtherType	Src IP	Dst IP	Protocol IP Proto	QoS IP ToS	Src L4 Port ICMP Type	Dst L4 Port ICMP Code	Action	Counter
*	*	0A:C8:*	*	*	*	*	*	*	*	*	*	Port 1	102
*	*	*	*	*	*	*	192.168.*.*	*	*	*	*	Port 2	202
*	*	*	*	*	*	*	*	*	*	21	21	Drop	420
*	*	*	*	*	*	*	*	0x806	*	*	*	Local	444
*	*	*	*	*	*	*	*	0x1*	*	*	*	Controller	1

OpenFlow Table: Basic Actions

- **All**: To all interfaces except incoming interface.
- **Controller**: Encapsulate and send to controller.
- **Local**: send to its local networking stack.
- **Table**: Perform actions in the next flow table (table chaining or multiple table instructions).
- **In_port**: Send back to input port.
- **Normal**: Forward using traditional Ethernet.
- **Flood**: Send along minimum spanning tree except the incoming interface.

OpenFlow Table: Basic Stats

Per Table	Per Flow	Per Port	Per Queue
Active Entries	Received Packets	Received Packets	Transmit Packets
Packet Lookups	Received Bytes	Transmitted Packets	Transmit Bytes
Packet Matches	Duration (Secs)	Received Bytes	Transmit overrun errors
	Duration (nanosecs)	Transmitted Bytes	
		Receive Drops	
		Transmit Drops	
		Receive Errors	
		Transmit Errors	
		Receive Frame Alignment Errors	
		Receive Overrun errors	
		Receive CRC Errors	
		Collisions	

- Provide counter for incoming flows or packets.
- Information on counter can be retrieved to control plane.
- Can be used to monitor network traffic.

Additional Feature to Rules and Stats

OpenFlow Version	Match fields	Statistics	# Matches		# Instructions		# Actions		# Ports	
			Req	Opt	Req	Opt	Req	Opt	Req	Opt
v 1.0	Ingress Port	Per table statistics	18	2	1	0	2	11	6	2
	Ethernet: src, dst, type, VLAN	Per flow statistics								
	IPv4: src, dst, proto, ToS	Per port statistics								
	TCP/UDP: src port, dst port	Per queue statistics								
v 1.1	Metadata, SCTP, VLAN tagging	Group statistics	23	2	0	0	3	28	5	3
	MPLS: label, traffic class	Action bucket statistics								
v 1.2	OpenFlow Extensible Match (OXM)		14	18	2	3	2	49	5	3
	IPv6: src, dst, flow label, ICMPv6									
v 1.3	PBB, IPv6 Extension Headers	Per-flow meter	14	26	2	4	2	56	5	3
		Per-flow meter band								
v 1.4	—	—	14	27	2	4	2	57	5	3
		Optical port properties								

Basic OpenFlow Recap

SDN Concept:

Management plane (Application Plane)



Control plane



Data plane



OpenFlow:

- Support different applications: routing, load balancers, monitoring, security, etc.
- Programmable: Modify and interact with the network model in control Plane.

- Global view of the entire network (the network model).
- Centralized per flow based control.
- Distributed system that creates a consistent, up-to-date network view (real time)
 - Runs on servers (controllers) in the network.
- Uses an open protocol to:
 - Get state information **from** switch.
 - Give control directives **to** switch.

Data and Control plane communicate via **secure Channel**

- Packet forwarding according to instruction stored in flow Tables.
- Provide statistic on network traffic to controller.
- Hardware: (Dump) Switches.

OpenFlow: More Details

SDN Concept

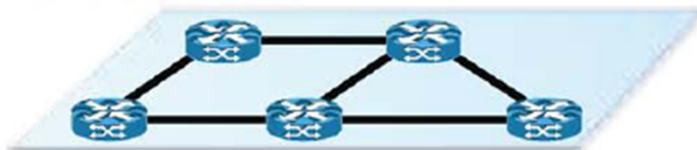
Management plane (Application Plane)



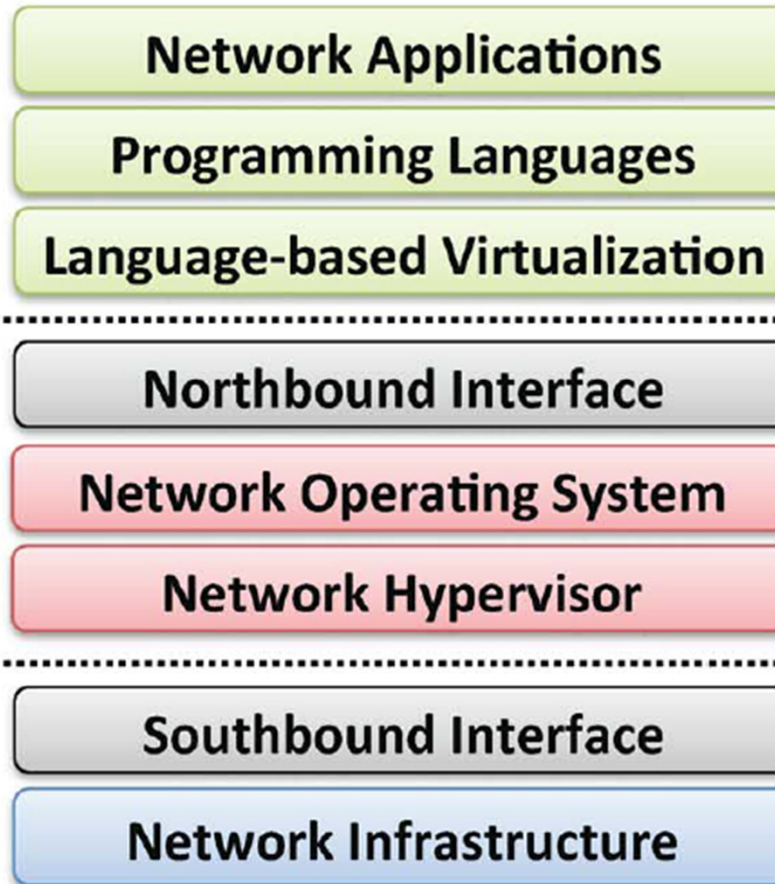
Control plane



Data plane



Different layers in OpenFlow



Discussed

Routing, load balancers, security, etc.

Make decisions and instructions

Firmware handling instructions from control plane (e.g Open Vswitch) via flow tables.

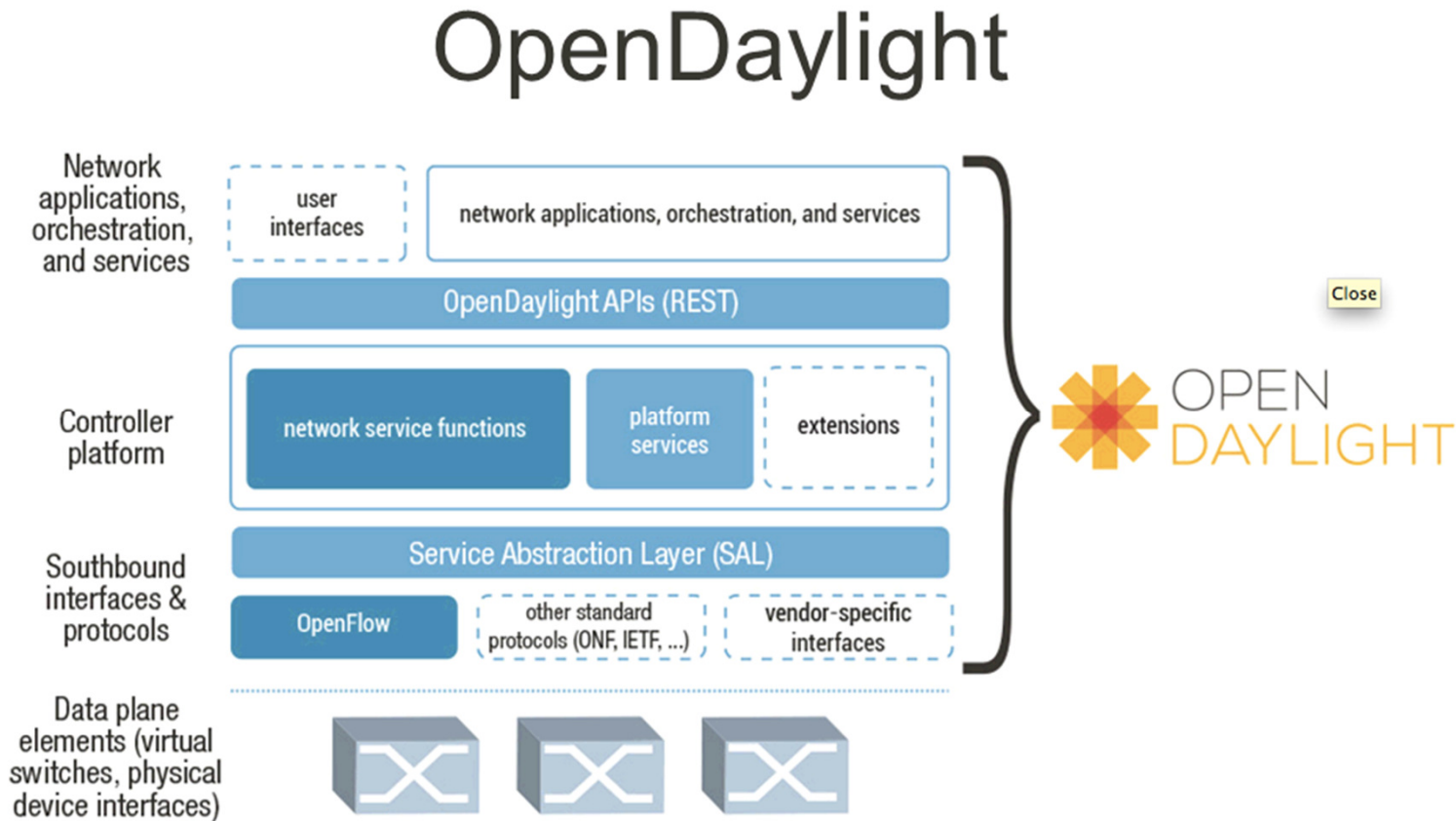
Hardware (switches)



OpenDaylight (Control Plane)

OpenDaylight (ODL) is a modular open platform for customizing and automating networks of any size and scale. The OpenDaylight Project arose out of the SDN movement, with a clear focus on network programmability. It was designed from the outset as a foundation for commercial solutions that address a variety of use cases in existing network environments.

OpenDaylight (Control Plane)



OpenDaylight Controller : Installation

OpenDaylight Controller runs in a JVM. Being a Java application, it can potentially run on any operating system that supports Java. However, for best results we recommend the following:

- A recent Linux distribution
- Java Virtual Machine 1.7

Note: The following instructions assumes:

- You are installing OpenDaylight Controller on your local Linux machine
- You will use the Mininet VM to create a virtual network

OpenDaylight Controller has not been tested in other environments.

Installing OpenDaylight Controller

- Download and extract the OpenDaylight Controller zip file
- Prerequisite: JVM 1.7+ (JAVA_HOME should be set to environment)

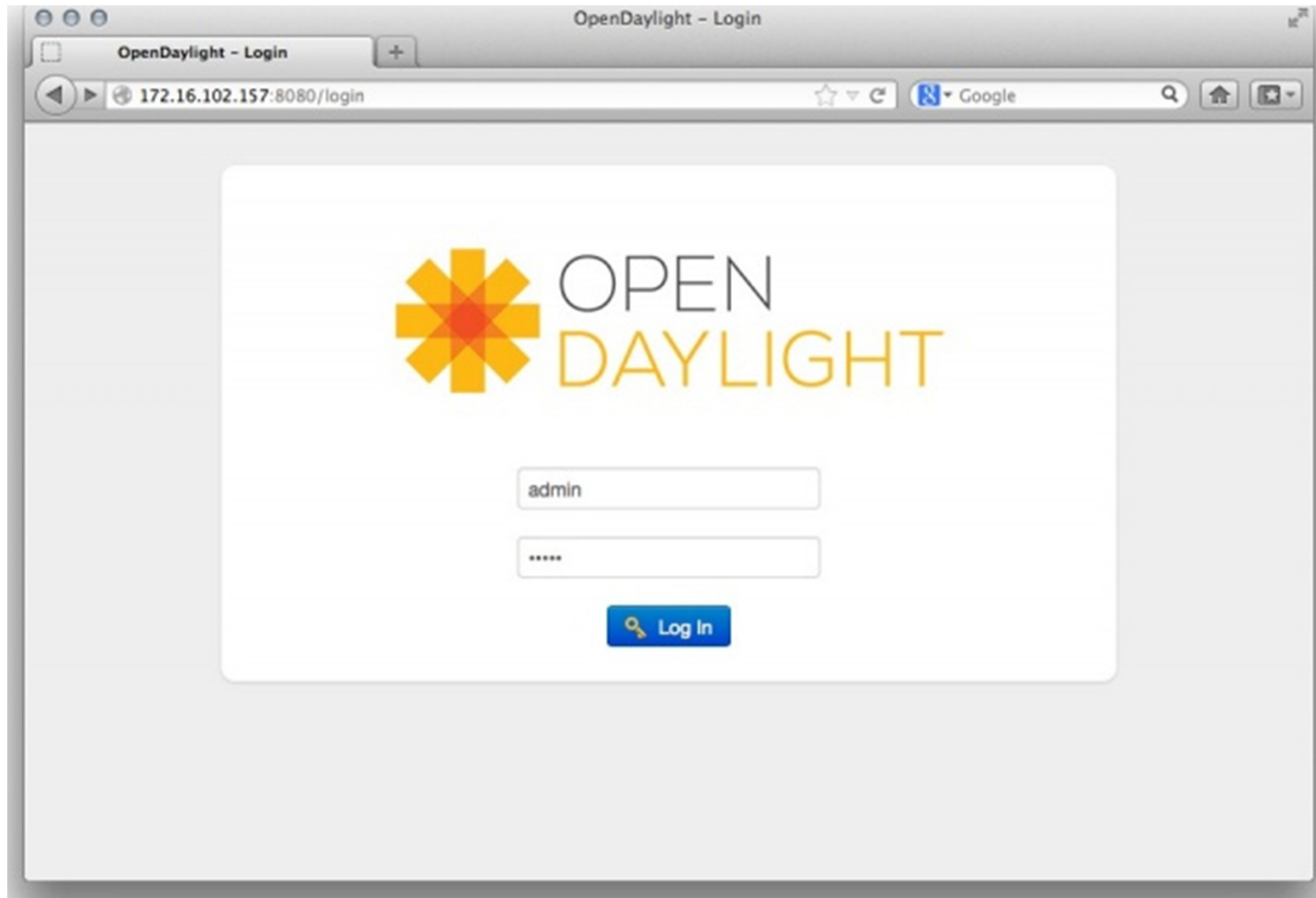
Download and Run the controller

- The source code of the OpenDaylight Controller is in a directory called **opendaylight**. In this directory (the corresponding directory in the git repository is usually the `distribution/opendaylight/target/distribution.opendaylight-0.1.6-SNAPSHOT-osgipackage/opendaylight`), you will see the following files:
 - **run.sh** — launches the OpenDaylight Controller on Linux/Unix systems
 - **run.bat** — launches OpenDaylight Controller on Windows systems
 - **version.properties** — indicates the build version
 - **configuration** — basic initialization files (internal to OpenDaylight Controller)
 - **lib** — Java libraries
 - **plugins** — OpenDaylight Controller's OSGi plugins

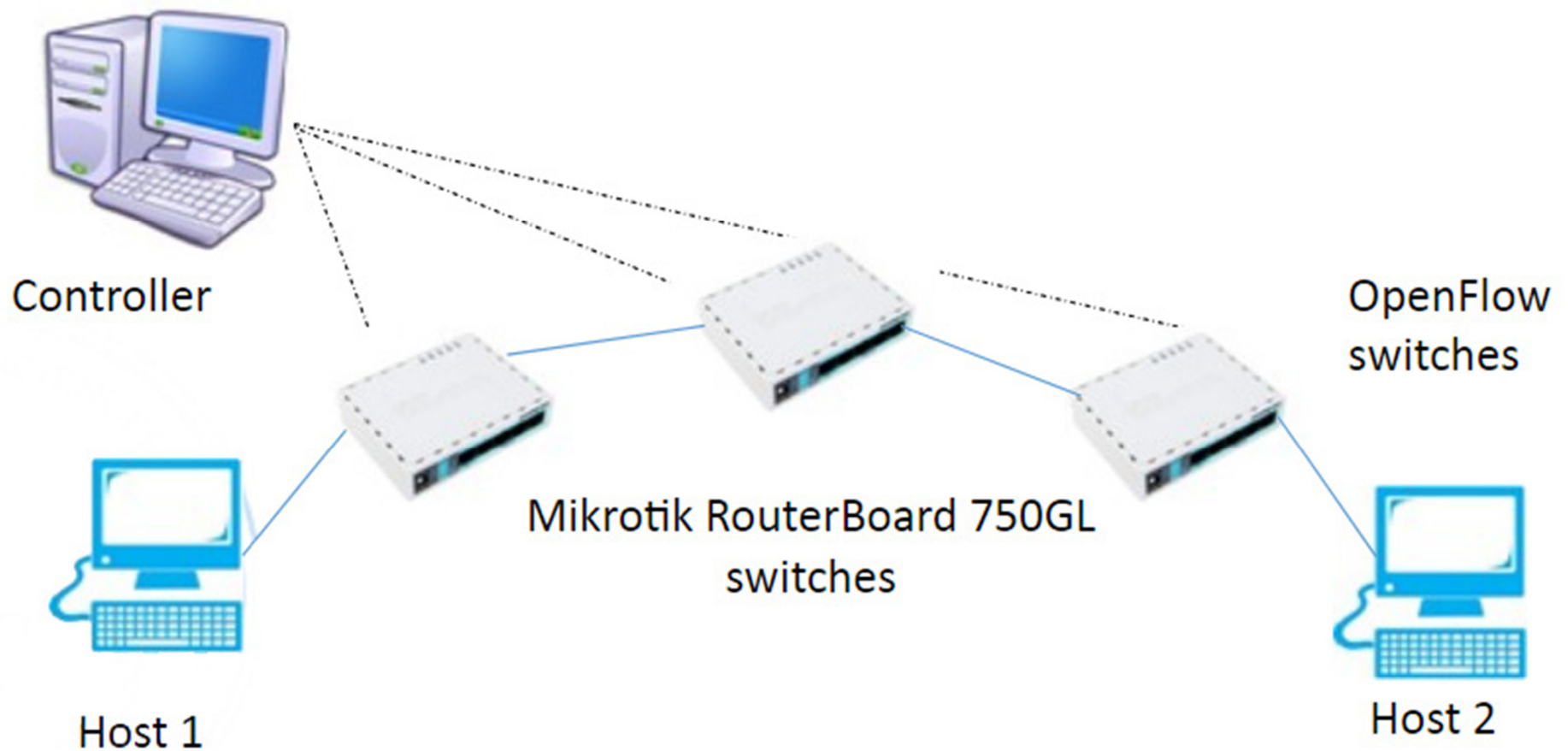
Download and Run the controller

- Enter **./run.sh** or **./run.bat** with administrator privileges to launch the OpenDaylight Controller application.
- Navigate to **http://controller-ip:8080** to open the web interface, then use the following credentials to log in:
 - User: **admin**
 - Password: **admin**
- Once logged in, please ensure to change the default admin password. To do so, navigate to the **admin** menu in the top right corner of the web interface. Select **Users** and then select the admin user. A pop-up will then present the option to **Change Password**.

OpenDaylight (Control Plane)



OpenFlow Implementation in MikroTik



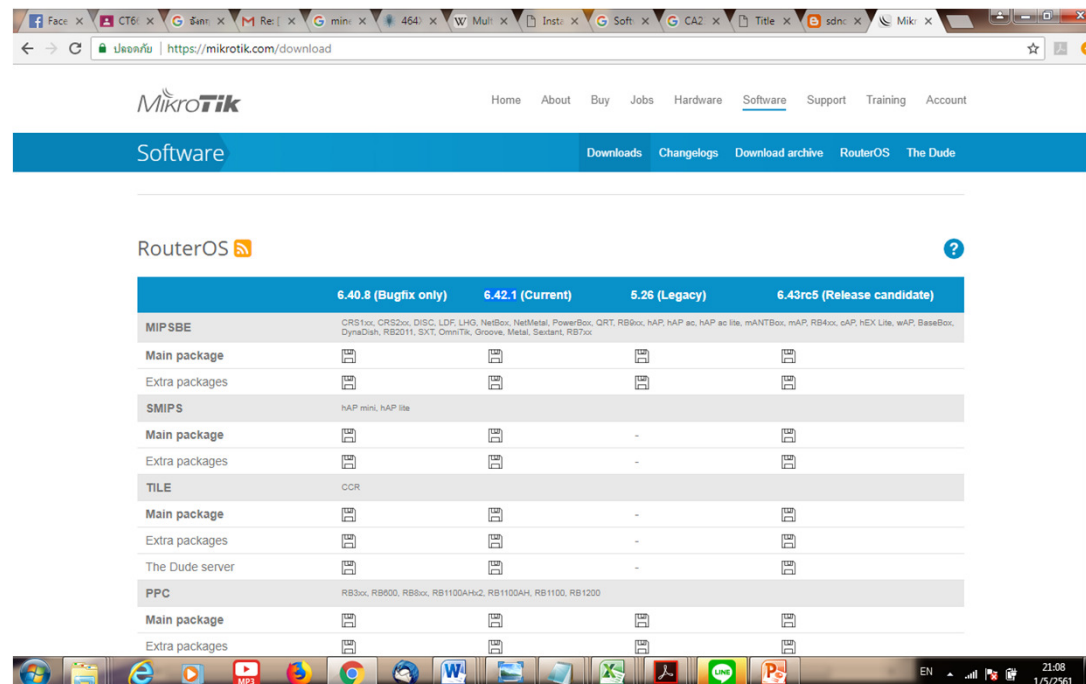
Open Flow Implementation in MikroTik

- Currently RouterOS implements OpenFlow version 1.0.0 required features.
- Support for newer, versions, optional features and switchings hardware acceleration are to be added.
- Current implementation should be considered **experimental-NOT production ready** and is available for evaluation purposes.
- OpenFlow support is available as standalone openflow package.



Installation of Openflow on RouterOS (i.e RouterBoard 750GL)

- RouterOS 6.42.1 version has the compatibility for the openflow. So first step is to upgrade your RouterOS to latest version (i.e currently latest version is 6.42.1).



The screenshot shows the Mikrotik RouterOS download page. The browser address bar displays 'https://mikrotik.com/download'. The page features a navigation menu with 'Software' selected. Below the navigation, there is a table of RouterOS versions and their corresponding packages.

	6.40.8 (Bugfix only)	6.42.1 (Current)	5.26 (Legacy)	6.43rc5 (Release candidate)
MIPSBE <small>CRS1xx, CRS2xx, DISC, LOF, LHG, NetBox, NetMetal, PowerBox, QRT, RB0xx, hAP, hAP ac, hAP ac lite, mANTBox, mAP, RB4xx, cAP, HEX Lite, wAP, BaseBox, DynaDish, RB2011, SXT, OmniTik, Groove, Metal, Gardant, RB7xx</small>				
Main package	[Download]	[Download]	[Download]	[Download]
Extra packages	[Download]	[Download]	[Download]	[Download]
SMIPS <small>hAP mini, hAP lite</small>				
Main package	[Download]	[Download]	-	[Download]
Extra packages	[Download]	[Download]	-	[Download]
TILE <small>CCR</small>				
Main package	[Download]	[Download]	-	[Download]
Extra packages	[Download]	[Download]	-	[Download]
The Dude server	[Download]	[Download]	-	[Download]
PPC <small>RB3xx, RB900, RB8xx, RB1100AH2, RB1100AH, RB1100, RB1200</small>				
Main package	[Download]	[Download]	[Download]	[Download]
Extra packages	[Download]	[Download]	[Download]	[Download]

Upgrade process :

- First step - visit <http://www.mikrotik.com/download> and head to the download page, there choose the type of system you have the RouterOS installed on
- Download the the upgrade package and the All packages (In All packages folder you will find the openflow package), it will include all the functionality of RouterOS:

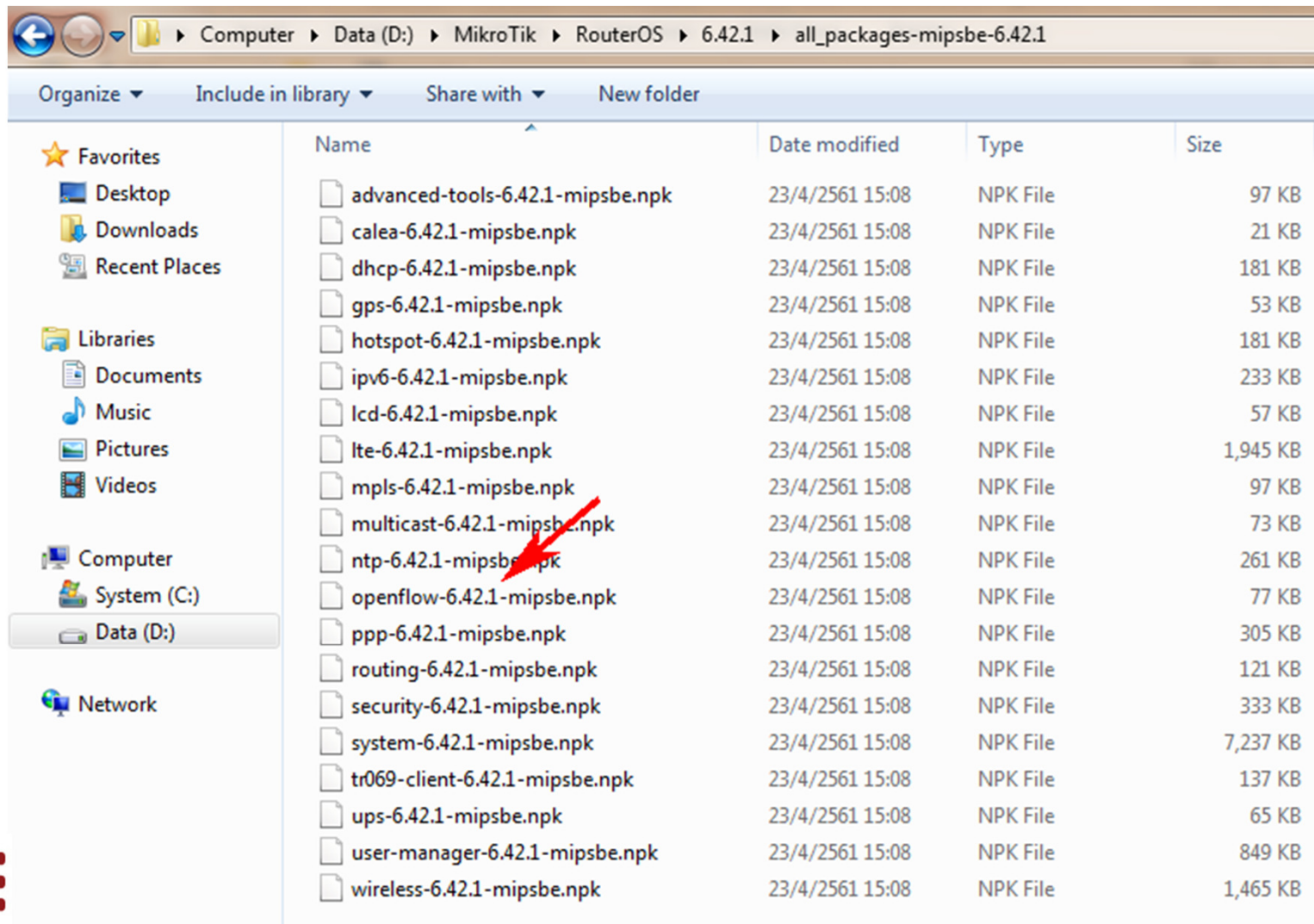
Upgrade process :



Software

	6.40.8 (Bugfix only)	6.42.1 (Current)	5.26 (Legacy)	6.43rc5 (Release candidate)
MIPSBE	CRS1xx, CRS2xx, DISC, LDF, LHG, NetBox, NetMetal, PowerBox, QRT, RB9xx, hAP, hAP ac, hAP ac lite, mANTBox, mAP, RB4xx, cAP, hEX Lite, wAP, BaseBox, DynaDish, RB2011, SXT, OmniTik, Groove, Metal, Sextant, RB7xx			
Main package				
Extra packages				
SMIPS	hAP mini, hAP lite			
Main package			-	
Extra packages			-	
TILE	CCR			
Main package			-	
Extra packages			-	
The Dude server			-	

Upgrade process :



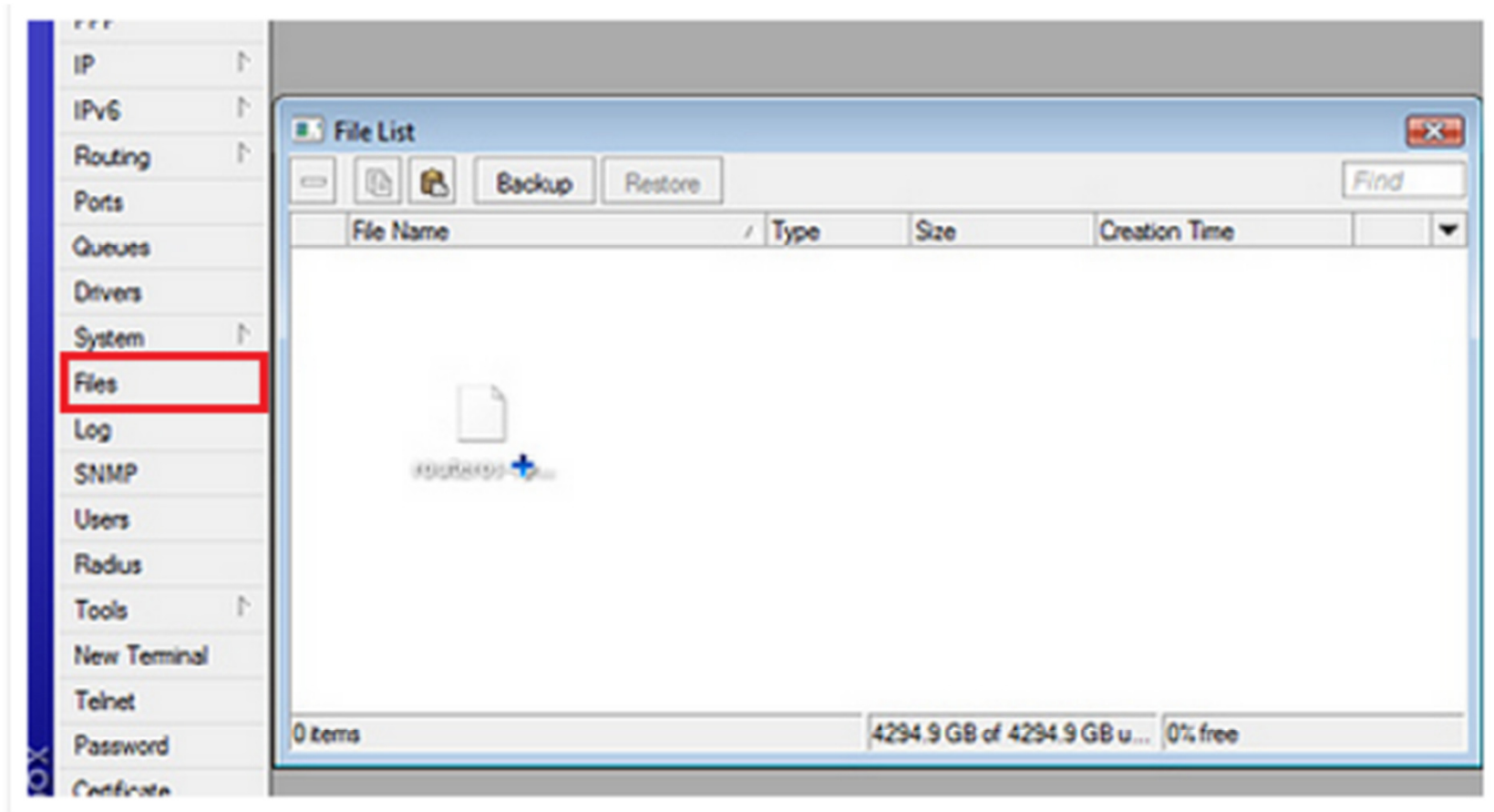
The screenshot shows a Windows File Explorer window with the address bar set to "Computer > Data (D:) > MikroTik > RouterOS > 6.42.1 > all_packages-mipsbe-6.42.1". The left sidebar shows the "Data (D:)" drive selected. The main pane displays a list of files with columns for Name, Date modified, Type, and Size. A red arrow points to the file "ntp-6.42.1-mipsbe.npk".

Name	Date modified	Type	Size
advanced-tools-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	97 KB
calea-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	21 KB
dhcp-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	181 KB
gps-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	53 KB
hotspot-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	181 KB
ipv6-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	233 KB
lcd-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	57 KB
lte-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	1,945 KB
mpls-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	97 KB
multicast-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	73 KB
ntp-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	261 KB
openflow-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	77 KB
ppp-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	305 KB
routing-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	121 KB
security-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	333 KB
system-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	7,237 KB
tr069-client-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	137 KB
ups-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	65 KB
user-manager-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	849 KB
wireless-6.42.1-mipsbe.npk	23/4/2561 15:08	NPK File	1,465 KB

Upgrade process :

- Now install **Winbox** from the <http://www.mikrotik.com/download>.
- Open **winbox** and login into the router using (IP: 192.168.88.1 username=admin password=blank)
- Select the **downloaded file** (upgrade package file) with your mouse, and drag it to the **Files** menu. If there are some files already present, make sure to put the package in the root menu, not inside the hotspot folder.

Upgrade process :



Upgrade process :

- After it finishes - **REBOOT** and that's all! The New version number will be seen in the Winbox Title and in the Packages menu.
- To reboot the router click on the **New Terminal** on the left and type **system reboot**.
- After the reboot, your router will be up to date, you can check it in this menu by typing **/system package print** in the **New Terminal** window.
- If your router did not upgrade correctly, make sure you check the **log** by typing **/log print without-paging**.
- At this point your RouterBoard is now up gradated to the latest version and ready to install the **openflow** on it.

OpenFlow Installation:

- First extract the **All packages** folder that we had already downloaded. This folder contain the **openflow package**.
- Now again log in to router using **winbox** and go to the **Files** and drag there the **openflow package**.
- Now again **reboot** the router and to see the installed packages by typing **/system package print**. You will see that **openflow package** is listed there.

OpenFlow Implementation

- Due to this care must be taken to not disable access to the device when configuring OpenFlow.
- Currently only unencrypted TCP is available as the communications channel between RouterOS OpenFlow switch and controller .

OpenFlow Implementation

- RouterOS supports multiple OpenFlow switch instances, each of which can have separate controller(s) and sets of ports.
- Switch instance can be added using command:
[admin@MikroTik]>/openflow add name=ofswitch1 controllers=192.168.2.254

OpenFlow

Switches Flows Ports

+ - ✓ ✗ ⌵ Find

Name	Datapath ID	Passive Port	Controllers
ofswitch1	1/D4:CA:6D:57:0F:34	192.168.2.254	

OpenFlow Switch <ofswitch1>

Name: ofswitch1 OK

Datapath ID: 1/D4:CA:6D:57:0F:34 Cancel

Passive Port: ▼ Apply

Controllers: 192.168.2.254 ⬆️ Disable

Copy

Remove

enabled

1 item (1 selected)

OpenFlow Implementation

The screenshot shows a web-based interface for managing OpenFlow. The main window has tabs for 'Switches', 'Flows', and 'Ports'. The 'Ports' tab is active, displaying a table with columns for 'Interface', 'Switch', and 'Port ...'. A single entry is visible: 'ether3' on 'ofswitch1'. A modal dialog titled 'OpenFlow Port <ether3>' is open, showing configuration options for the selected port. The dialog includes fields for 'Interface' (ether3), 'Switch' (ofswitch1), 'Port Number' (1), and statistics for Tx and Rx (Bytes, Packets, Drops). Buttons for 'OK', 'Cancel', 'Apply', 'Disable', 'Copy', and 'Remove' are also present.

Interface	Switch	Port ...	Tx Bytes	Rx Bytes	Tx Packets	Rx Packets	Tx Drops	Rx Drops
ether3	ofswitch1		0 B	1563 B	0 B	15 B	0 B	0 B

OpenFlow Implementation

```
[admin@MikroTik ]> /openflow add name=ofswitch1 controllers=192.168.2.254
```

Switch ports can be added using command:

```
[admin@MikroTik]> / openflow port add switch=ofswitch1 interface=ether3
```

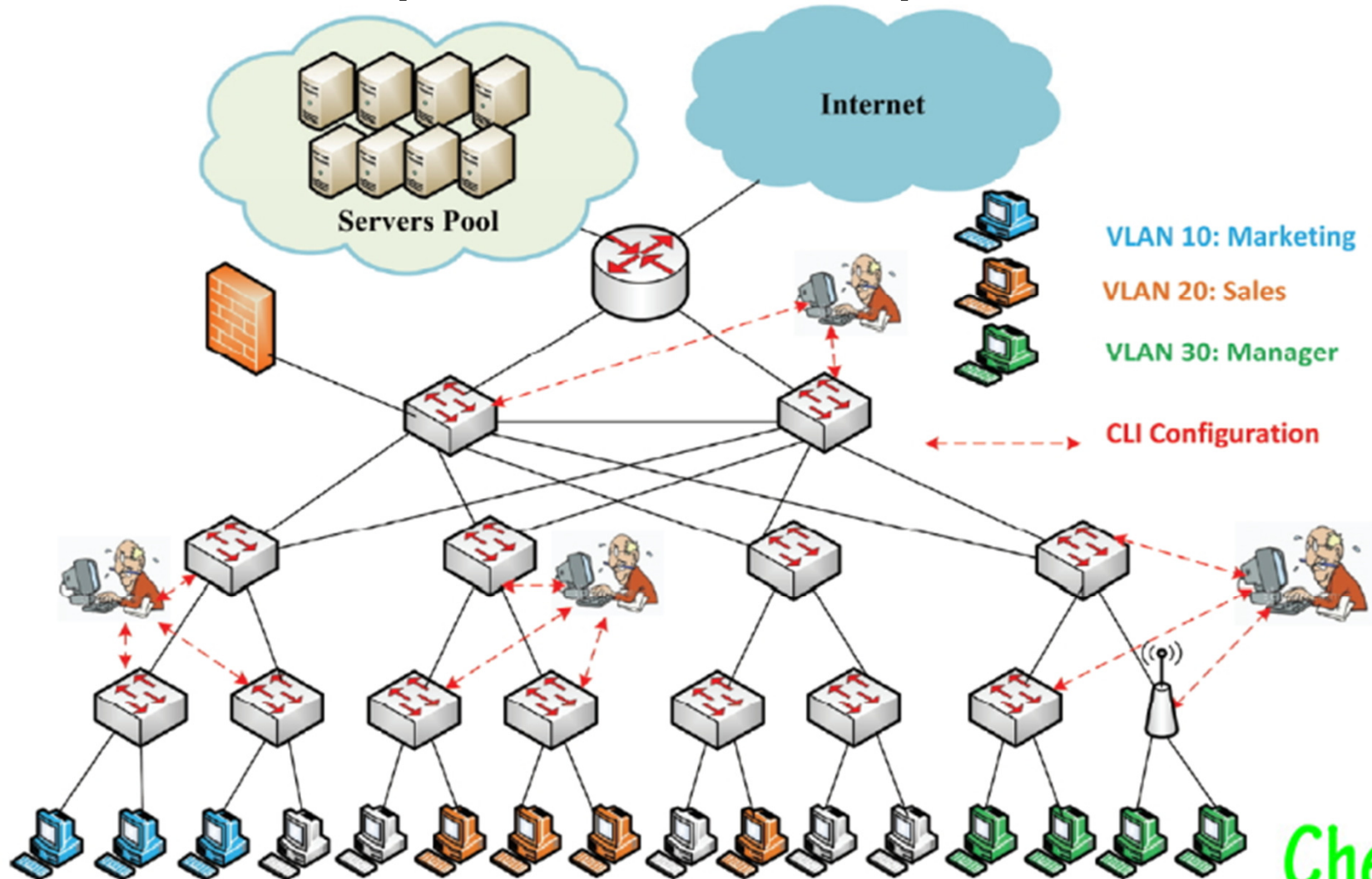
What can be achieved when using OpenFlow on RouterOS?

- **Speed and automation:** Programmatic APIs and logically centralized control streamline operations compared to manual order entry and NMS-based provisioning.
- **Operator innovation:** A standardized information model and open northbound APIs allow network operators to develop their own service models and constraintbased connection routing algorithms to differentiate their services

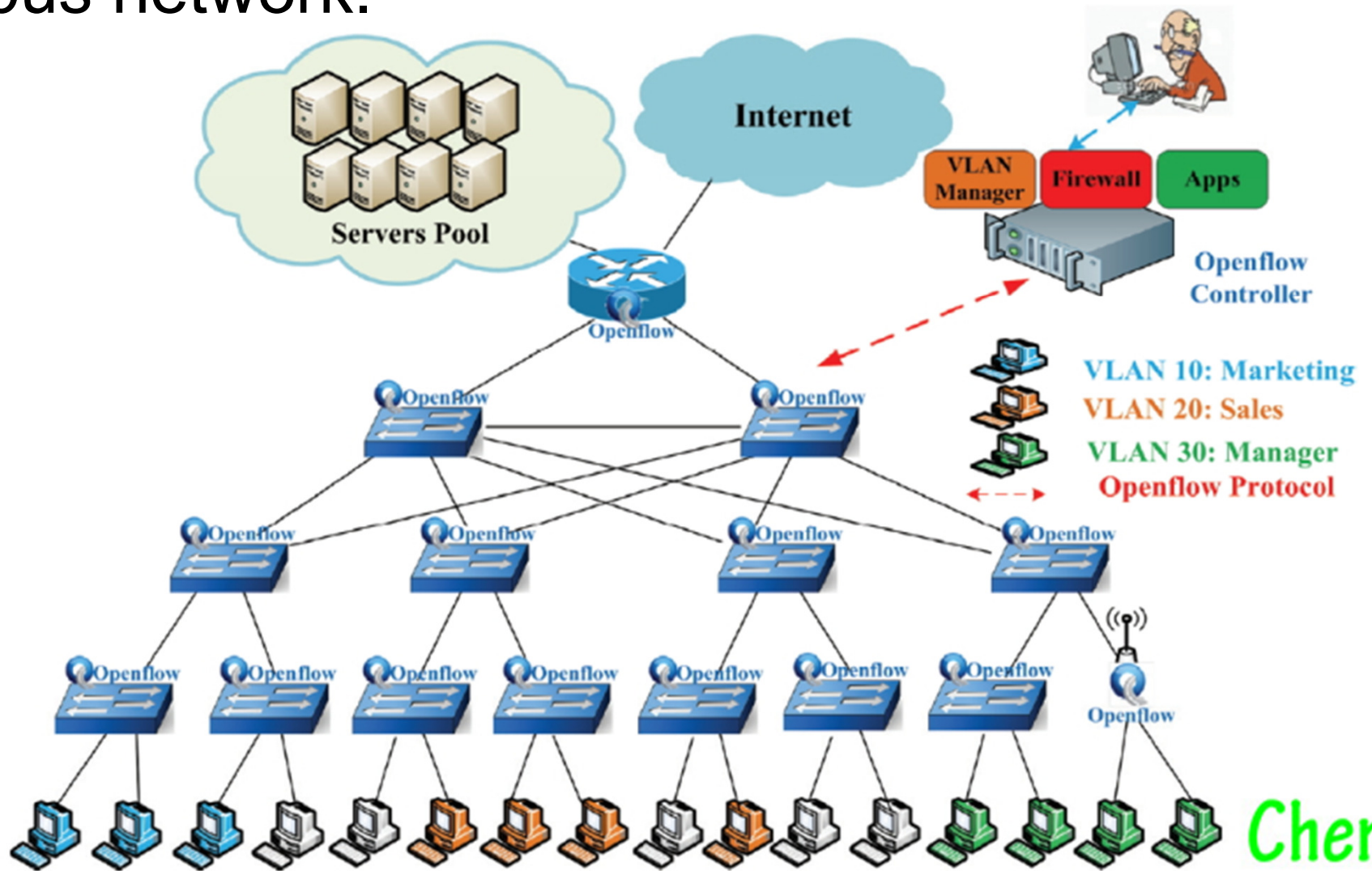
What can be achieved when using OpenFlow on RouterOS?

- **Efficiency:** Using OpenFlow for conjoint control of packet, circuit, and photonic transport resources enables multi-layer path computation and optimization to intelligently aggregate and express traffic to minimize stranded capacity
- **Multi-vendor networking:** Standard OpenFlow enables common connection provisioning commands across any compliant NE, allowing the choice of different vendors for different layers and/or geographical or operational domains.

Virtual Local Area Network (VLAN) management in the traditional enterprise and campus network.



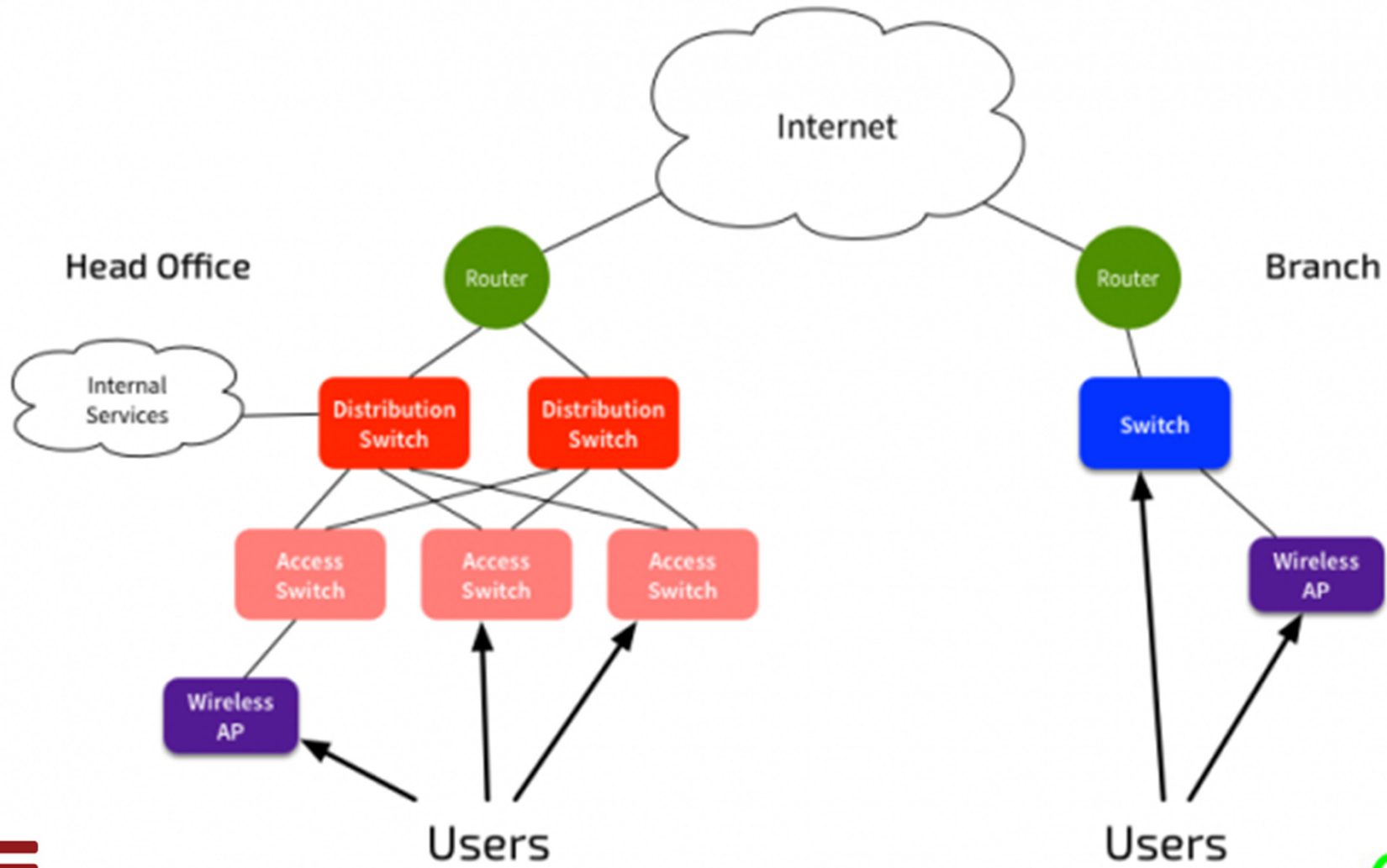
Virtual Local Area Network (VLAN) management in Software Defined Networking (SDN)-based enterprise and campus network.



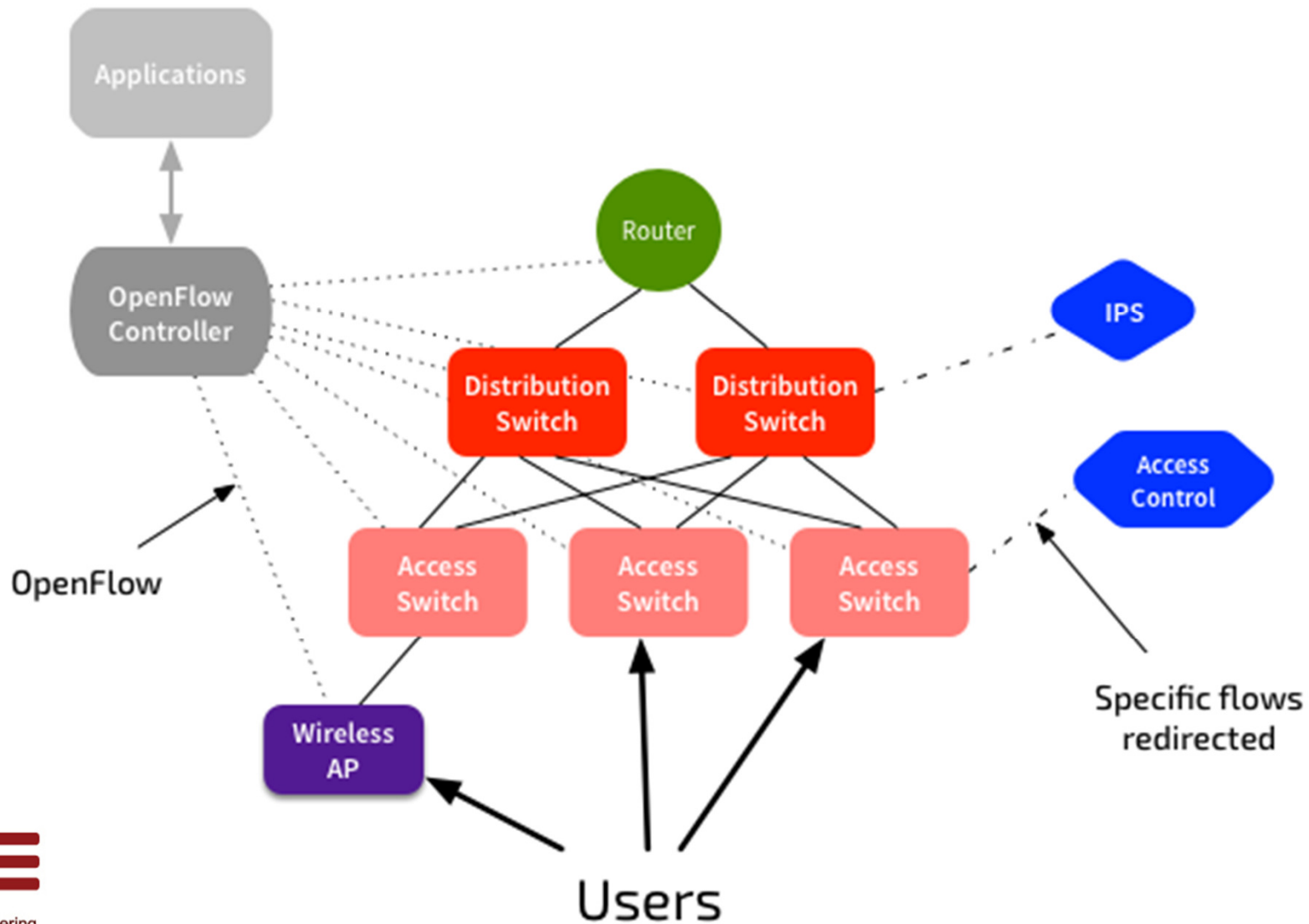
SDN for Enterprise

- **Campus/WAN**
- **Data Centre**

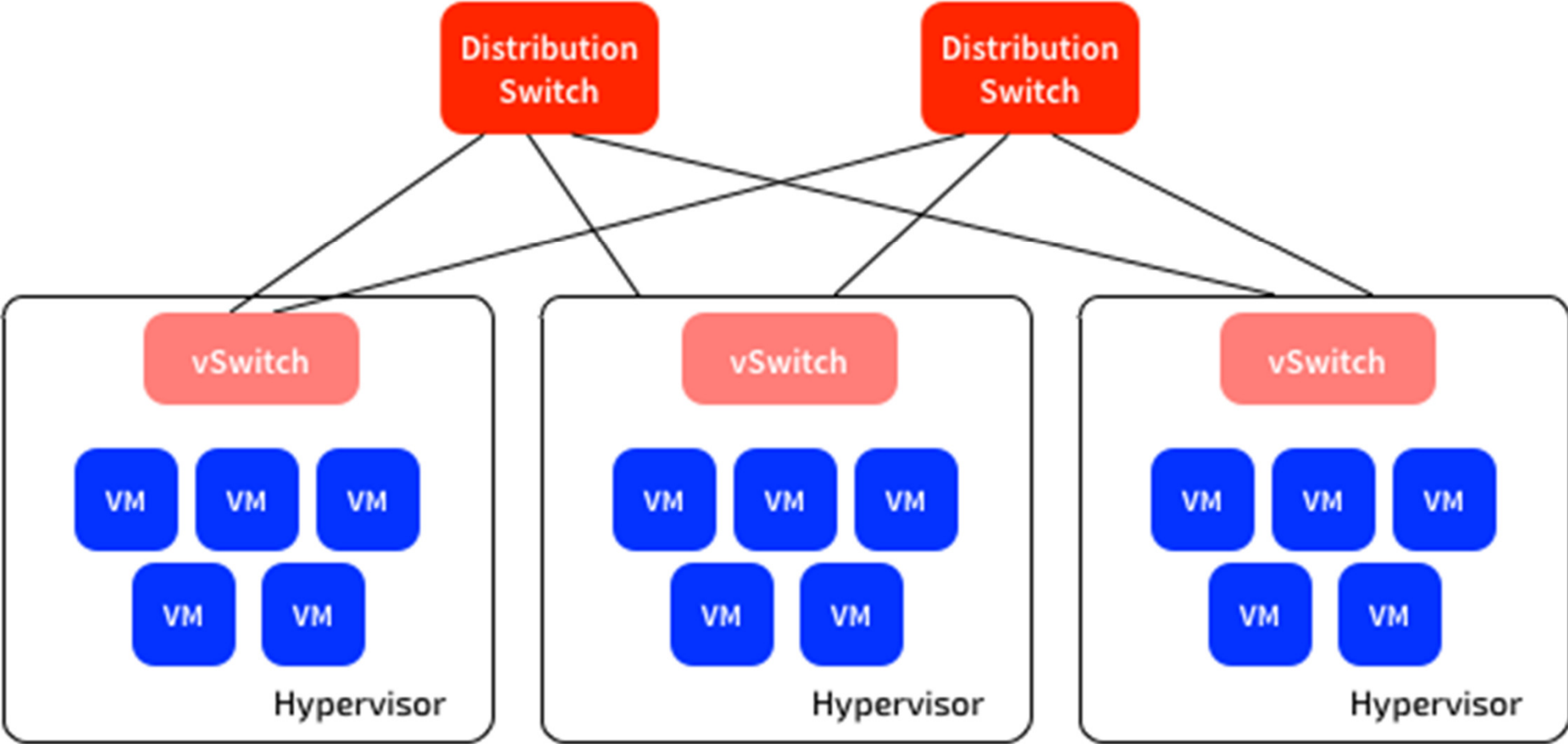
Campus/WAN (Traditional)



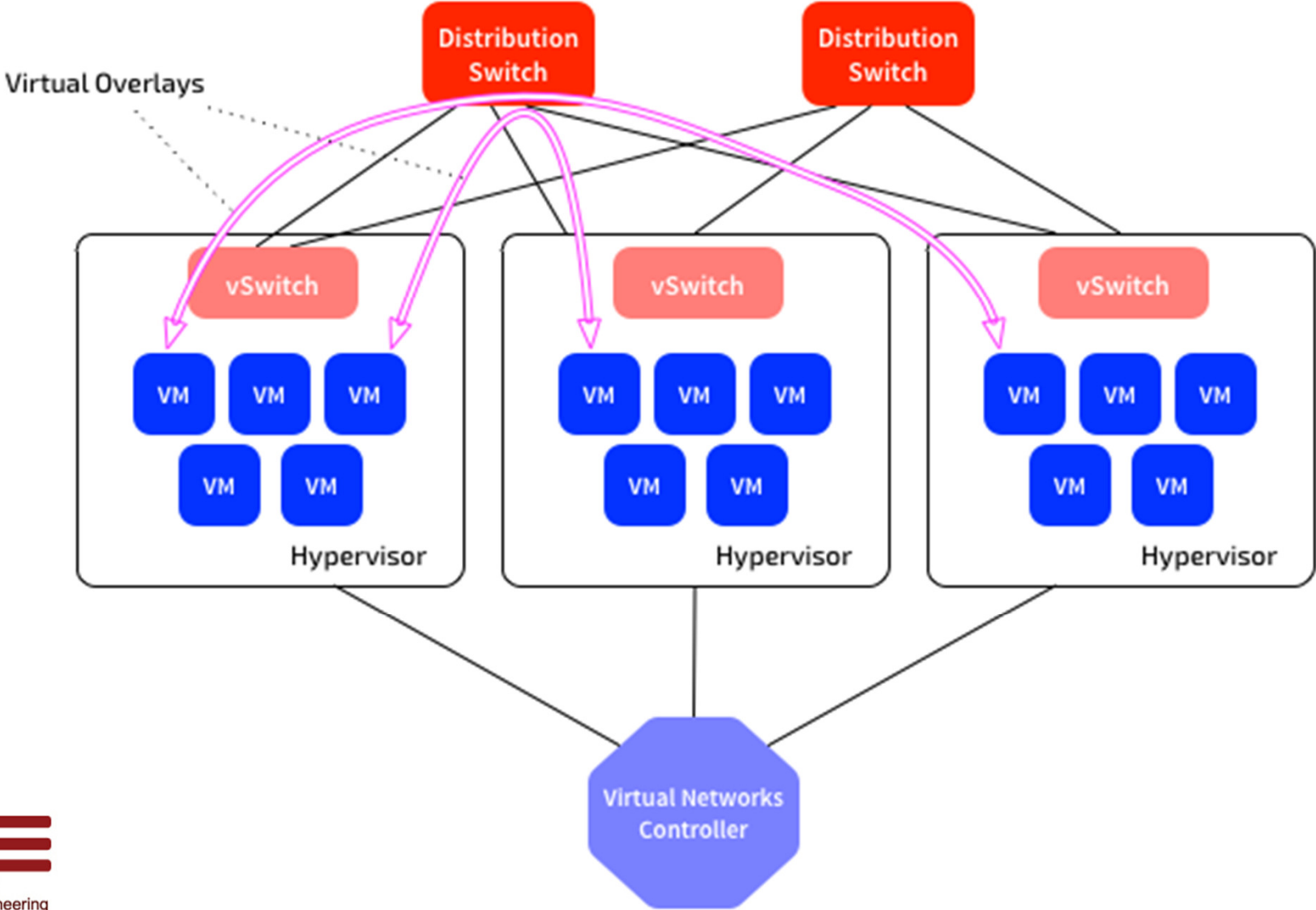
Campus/WAN (SDN)



Data Centre (Traditional)



Data Centre (SDN)



Conclusion

- Software Defined Networks(SDN) are a new networking paradigm in which there is separation of the forwarding and control planes. The control is migrated to a separate entity called the SDN Controller, Leaving the forwarding plane with bare-metal and virtual switches (switches devoid of control logic) to carry out packet forwarding. SDN allows external applications to program the network via an Application Programmable Interface(API).The most popular SDN protocol is OpenFlow. OpenvSwitch(OvS) is the most widely used OpenFlow switch.
- This Presentation is SDN concept
- We can use MikroTik for Data plane (OpenFlow)
- If you want to known in detail and Laboratory follow next presentation

Question????

- Thank you

Narong TOHKU

Mobile : 0824507266

Line id : lung.toh.1967

E-mail : narong.tohku@gmail.com

