

Welcome to

Networking Technologies for Cloud Computing

USTC-CYSC6402P
Instructor: Chi Zhang
Fall 2020



Today's agenda

- Introduction to Software Defined Networking
 - What is SDN? Why SDN?
 - History of SDN
 - SDN standardization
 - SDN deployment models
 - OpenFlow
 - SDN applications

OpenFlow

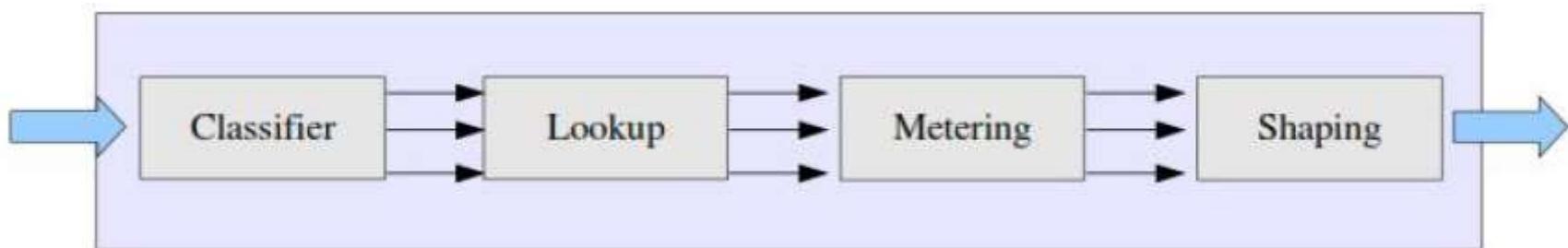
- OpenFlow overview
- Flow abstraction
- OF-switch abstraction
- Flow table pipeline
- Group table & meter table
- OpenFlow protocol
- Modes of operations

OpenFlow

- OpenFlow overview
- Flow abstraction
- OF-switch abstraction
- Flow table pipeline
- Group table & meter table
- OpenFlow protocol
- Modes of operations

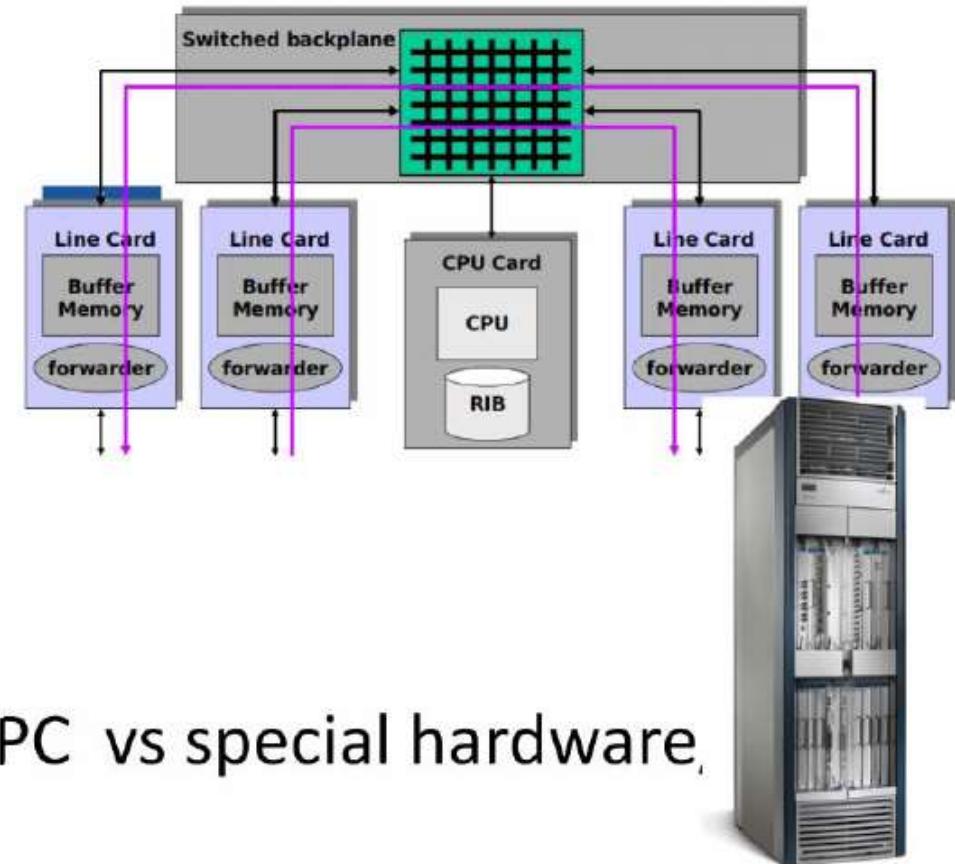
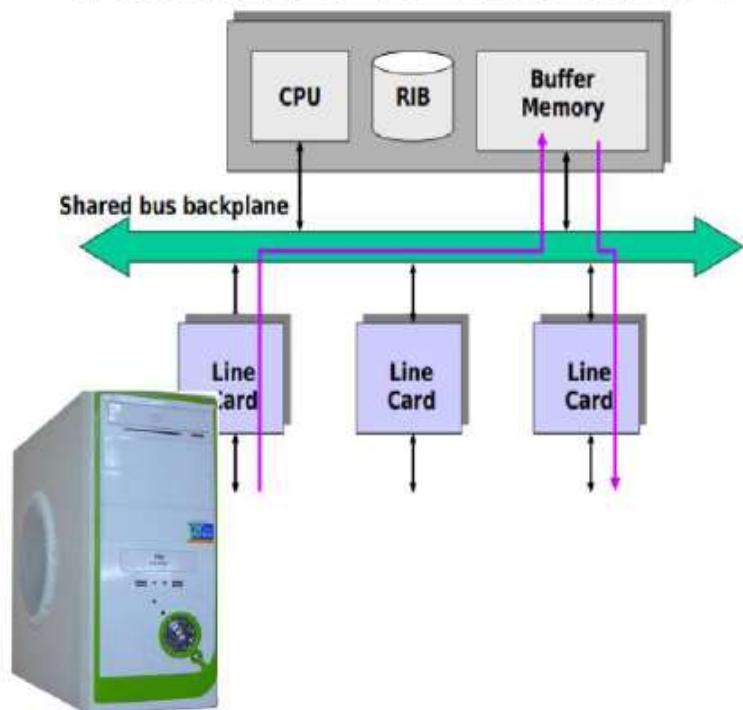
Traditional router architecture

- What does a router do?
 - Packet Forwarding (MPLS, IPv4, IPv6, Tunneling etc.)
 - Packet Filtering (access-list)
 - Packet Classification
 - Metering/Policing/Shaping
 - Compute Route: Build routing and forwarding



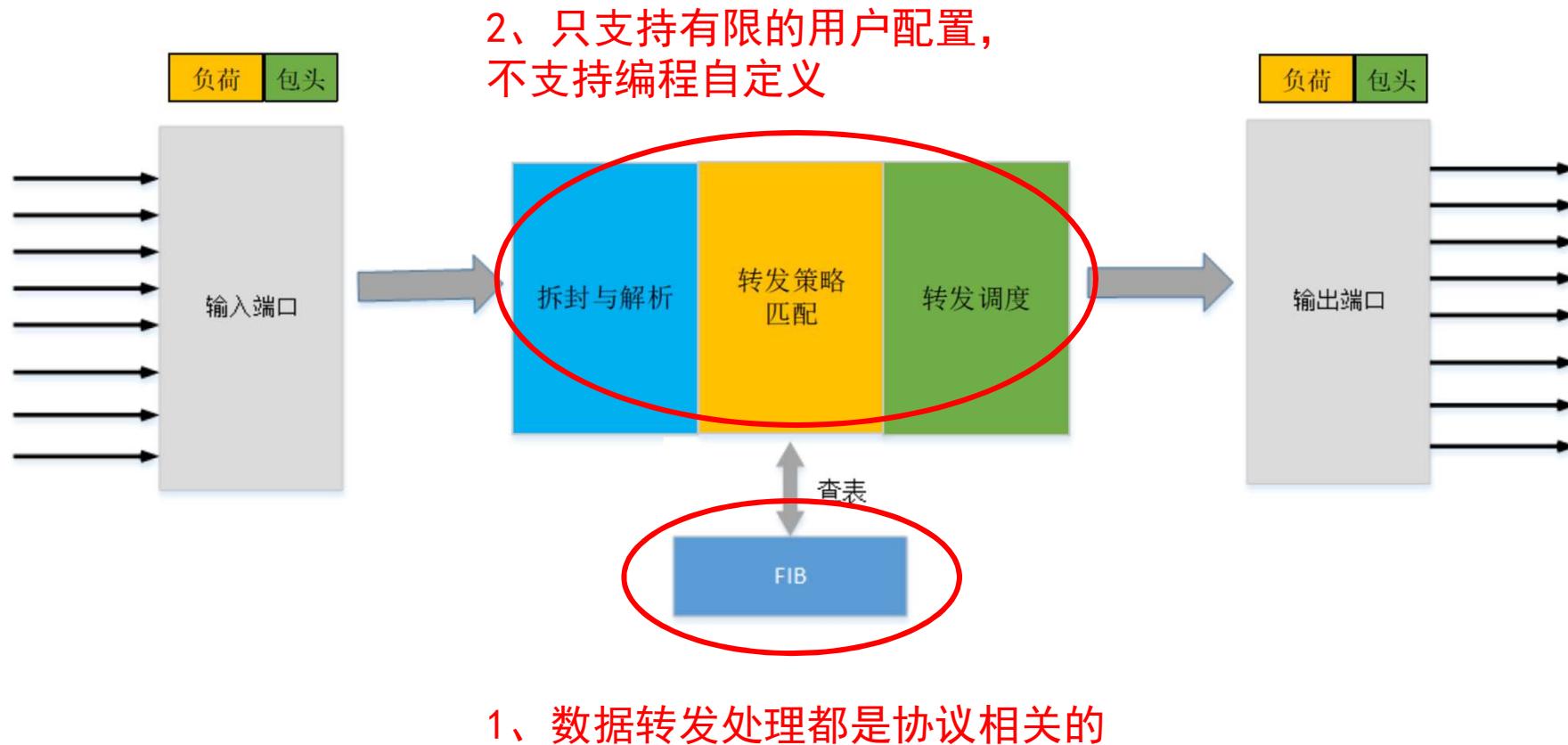
Traditional router architecture

Software vs Hardware Router



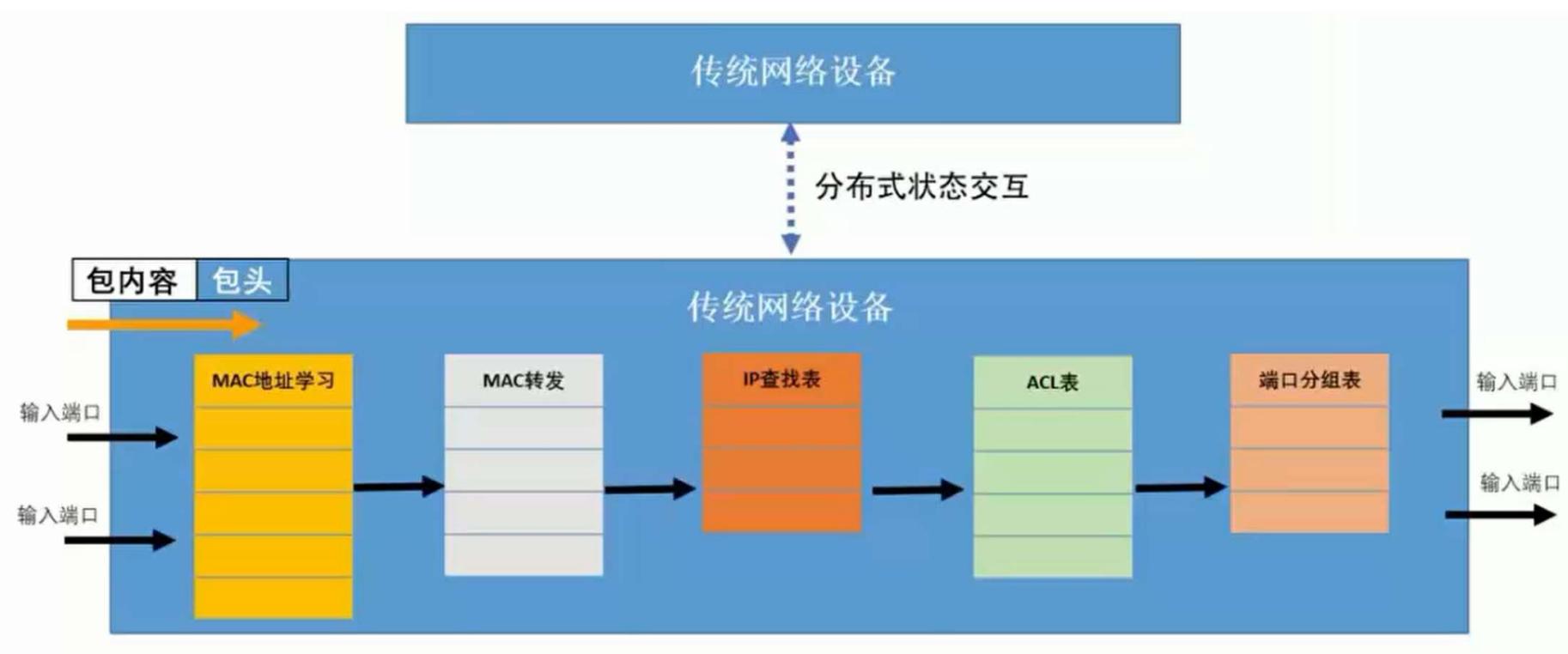
Regular commodity server/PC vs special hardware

Traditional data plane



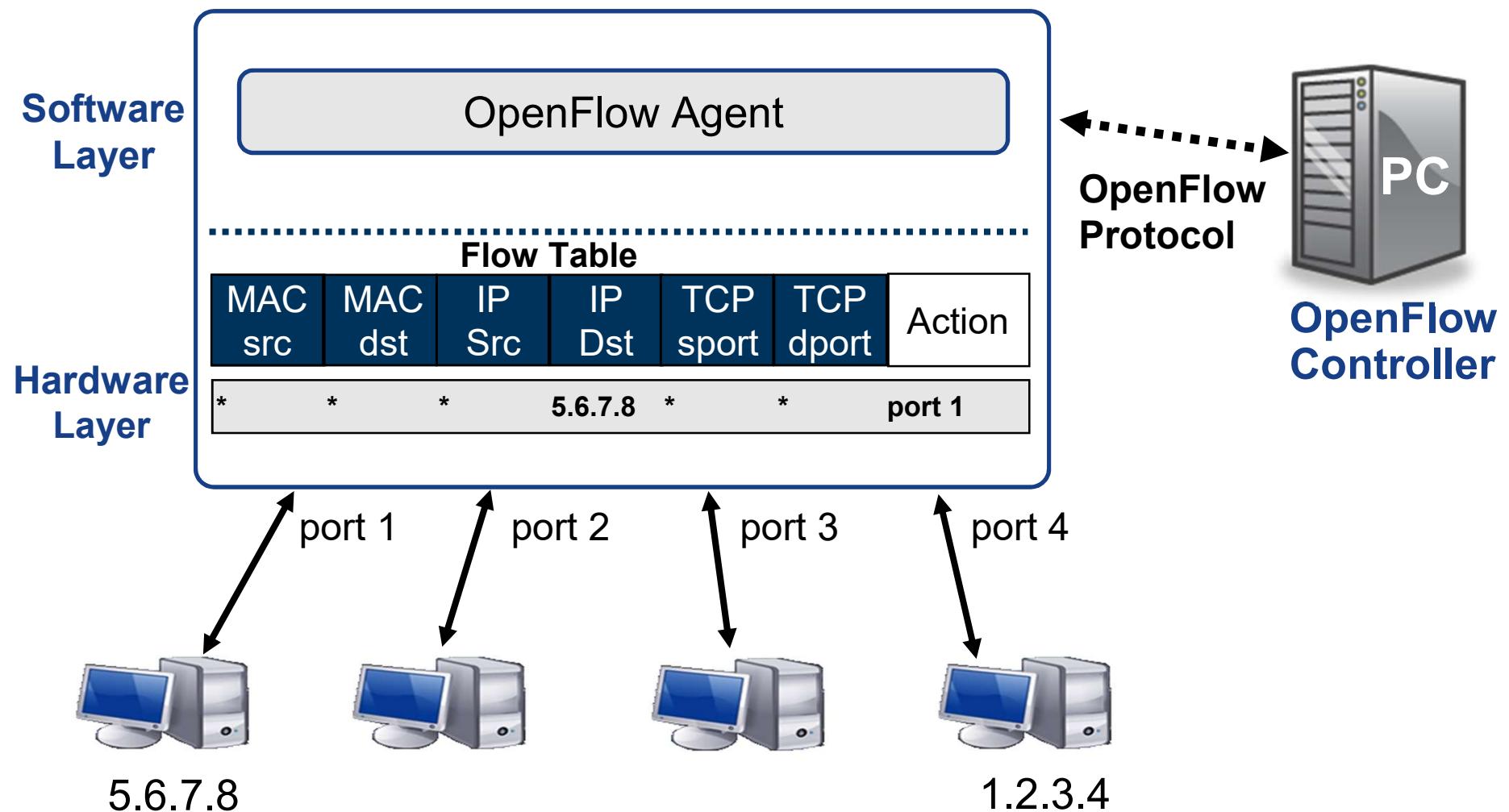
Traditional data plane

2、控制软件通过分布式协议确定二层或三层转发表



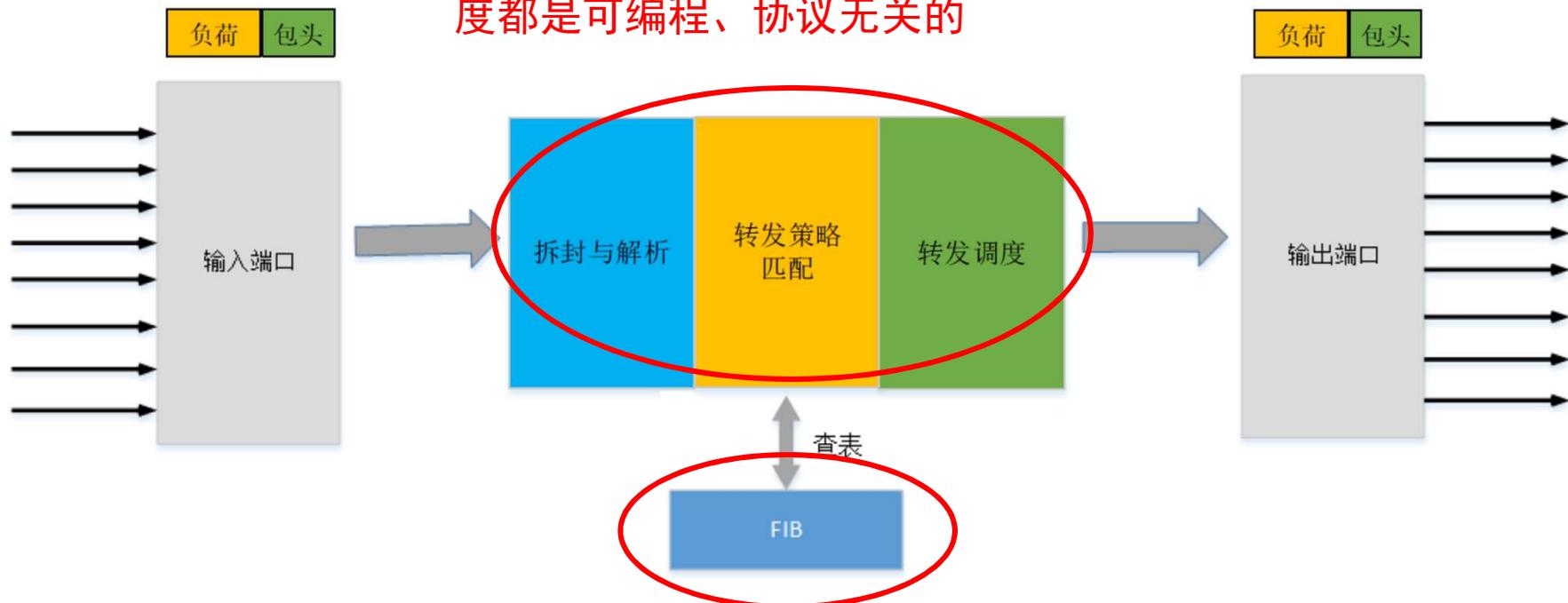
1、数据转发处理都是协议相关的、与硬件紧耦合、不支持用户编程

SDN Switch



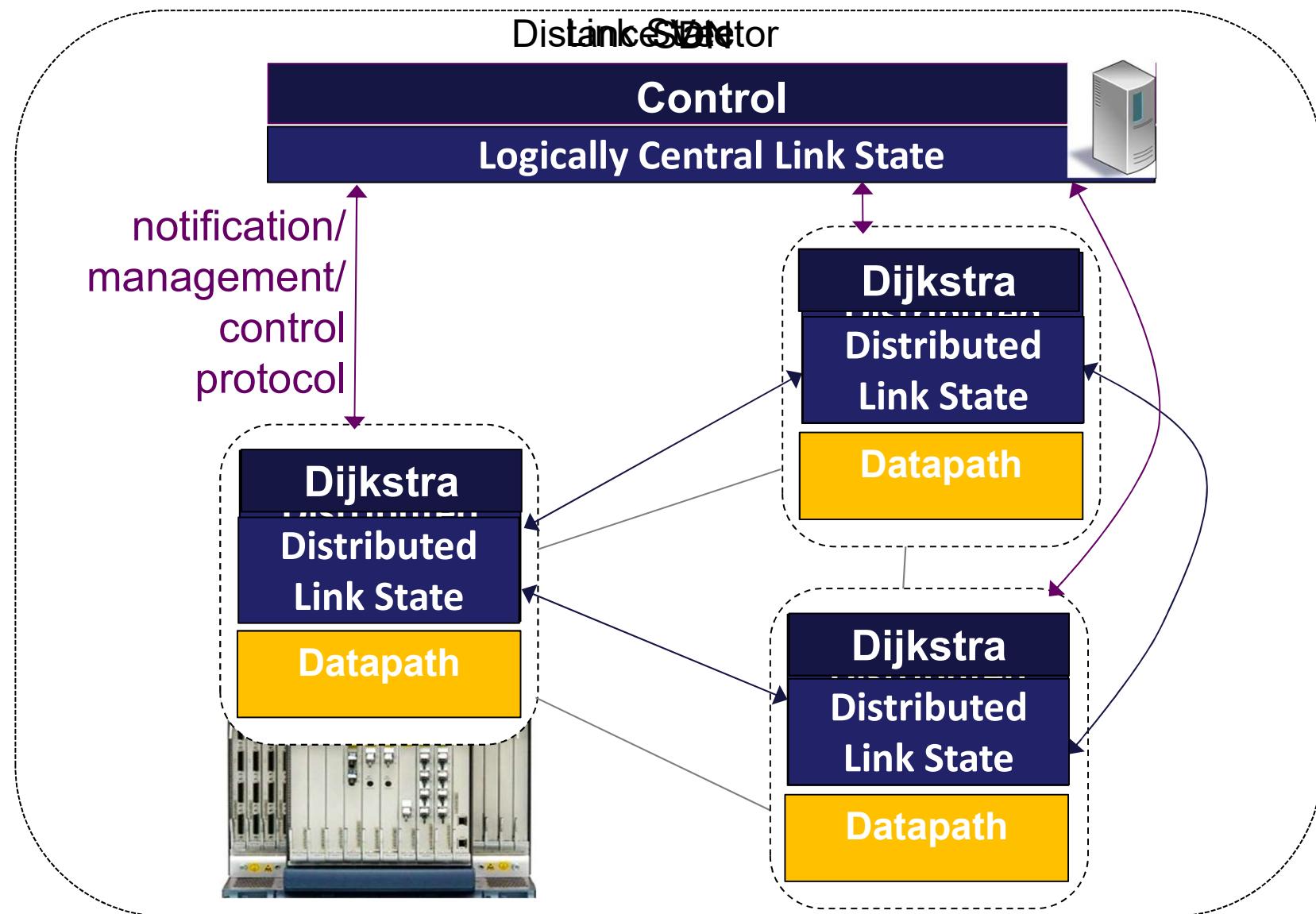
SDN data plane

2、处理流程中的：解析、转发和调度都是可编程、协议无关的



1、传统网络设备的二层或三层转发表抽象成流表；由控制器填写

An Evolution View of Intradomain Routing

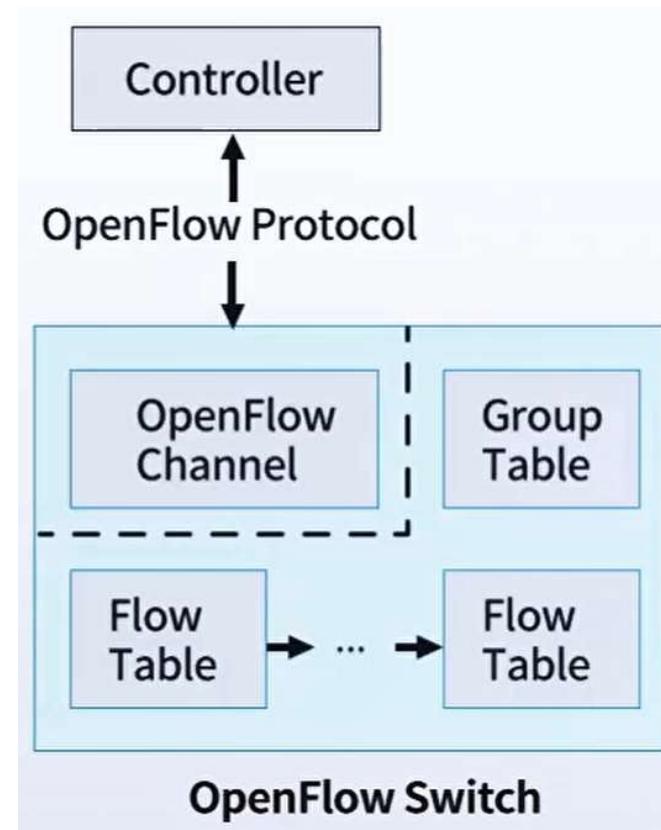


Why SDN (A “Wiseman” View)?

- Distributed computing is hard, e.g.,
 - FLP Impossibility Theorem
 - Arrow’s Impossibility Theorem
- Achieved good design for only few specific tasks (e.g., state distribution, leader election). Hence, one can consider SDN as a way of moving away from generic distributed computing, by focusing on utilizing the few well-understood primitives, in particular logically centralized state.

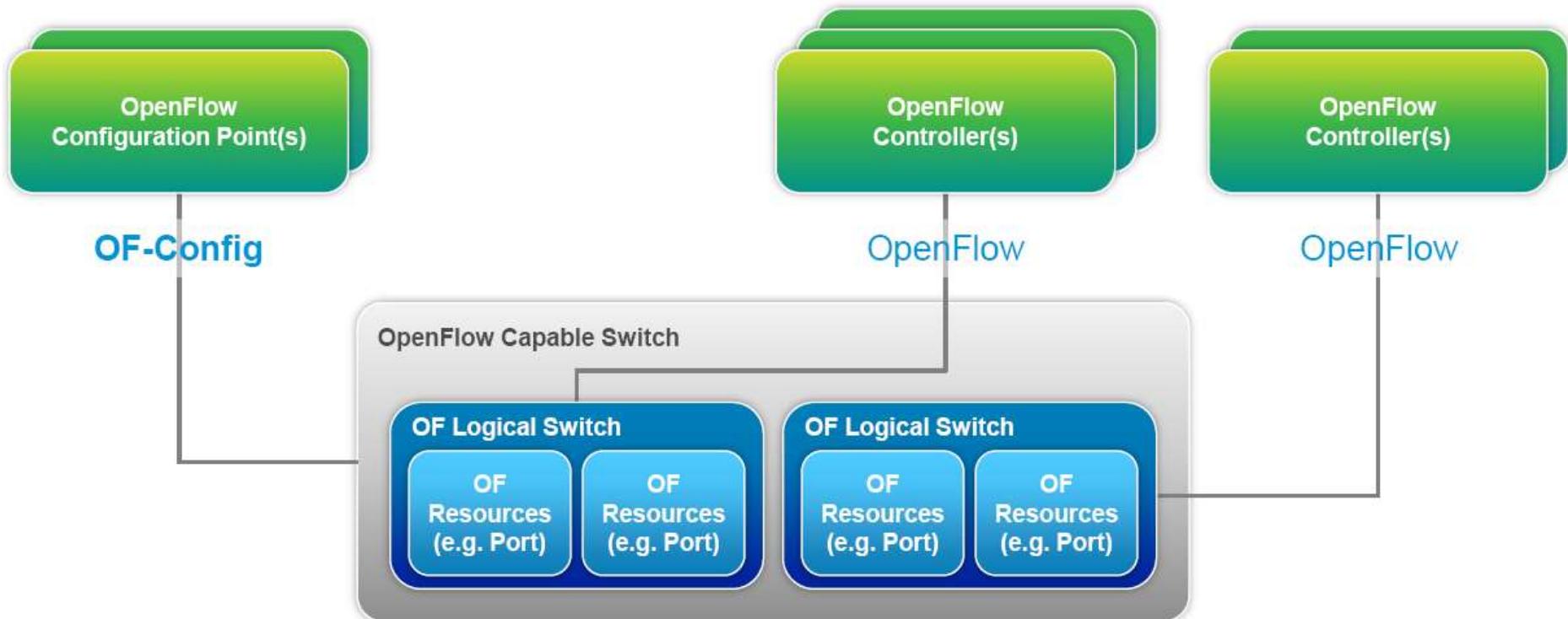
OpenFlow overview

- OpenFlow Components
 - Application Layer Protocol: OF-Protocol
 - Device Model: OF-Device Model (abstraction of a device with Ethernet interfaces and a set of forwarding capabilities)
 - Transport Protocol: Connection between OF-Controller and OF-Device*
- Observation
 - OF-Controller and OF-Device need pre-established IP-connectivity



* TLS, TCP – OF 1.3.0 introduces auxiliary connections, which can use TCP, TLS, DTLS, or UDP

OpenFlow & OF-Config



- IP地址、传输协议、端口号要预先配置
- OF逻辑交换机相关的资源要预先配置
- Controller要预先配置，与Controller的连接中断后如何处理也要设定好

OpenFlow versions status



Evolution of the specification:
Mature and Evolve

"Working code before new standards"

"ONF should not anoint a single reference implementation but instead encourage open-source implementations"; ONF board encourages multiple reference implementations

OpenFlow 1.0.X : no work planned

OpenFlow 1.3.X: long term support

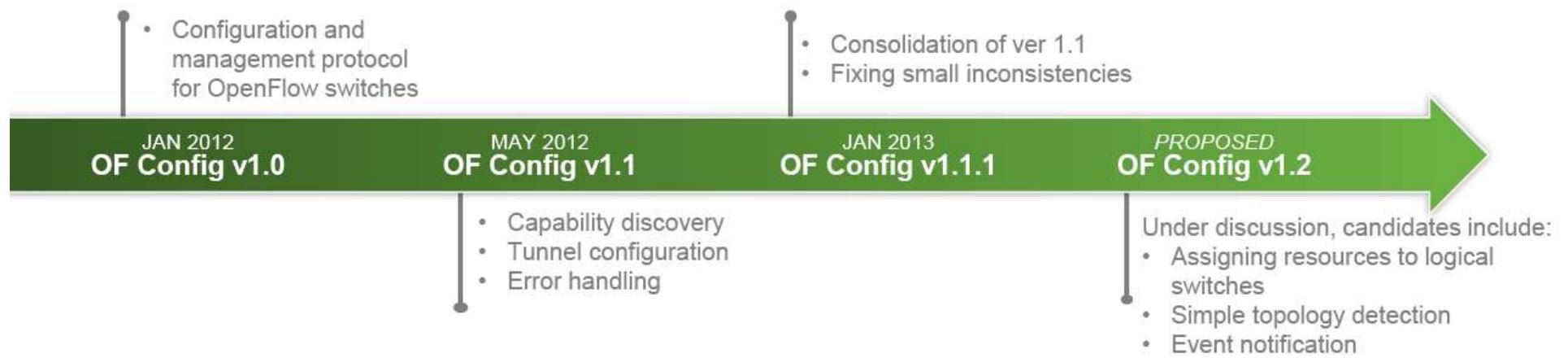
OpenFlow 1.4: extensibility, incremental improvements

OpenFlow versions status

OpenFlow 版本	版本代碼	釋出時間	釋出組織
1.0.0	0x01	2009/12	openflow.org
1.1.0	0x02	2011/2	openflow.org
1.2.0	0x03	2011/12	ONF
1.3.0(最新 1.3.5)	0x04	2012/6 (2015/4)	ONF
1.4.0(最新 1.4.1)	0x05	2013/10 (2015/4)	ONF
1.5.0(最新 1.5.1)	0x06	2014/12 (2015/4)	ONF

ONF: Open Networking Foundation

OpenFlow configuration protocol evolution



Discussions led by the ONF
Configuration and
Management Working Group

OpenFlow Terminology

术语	描述
Byte	一个8bit的序列，网络处理基本单元
Packet	以太帧，包括 header 和payload
Port	packets 进出OpenFlow pipeline 的地方，由物理端口， Switch 定义的logic端口， OpenFlow 协议定义的 reserved端口
Pipeline	OFS 中实现matching 、转发和 packet 修改的flow table流水线
Flow table	pipeline 的一个stage ，包含 flow entries
Flow entry	flow table 中的元素，用于 packets查找和处理，包括用于包匹配的一组 match fields ， 描述匹配顺序的 priority，跟踪记录 packets 的一组counters ， 一组待用的 instructions
Match Field	packet 匹配查找时使用的域，包括 packet header 、输入端口和 metadata值，一个 match field 可以是通配符（匹配任意值），也可以是 bitmasked 型
Metadata	一个 Maskable register value 用于table 之间的信息传递
Instruction	当一个 packet 匹配某个flow entry 时，对应的 instruction 描述特定的 OpenFlow 处理。一个 instruction 可以修改pipeline processing 过程，比如直接把 packets 发给另一个 flow table，也可以包含一组 actions 用来添加到 action set，或包含一个 action list立即应用给 packet

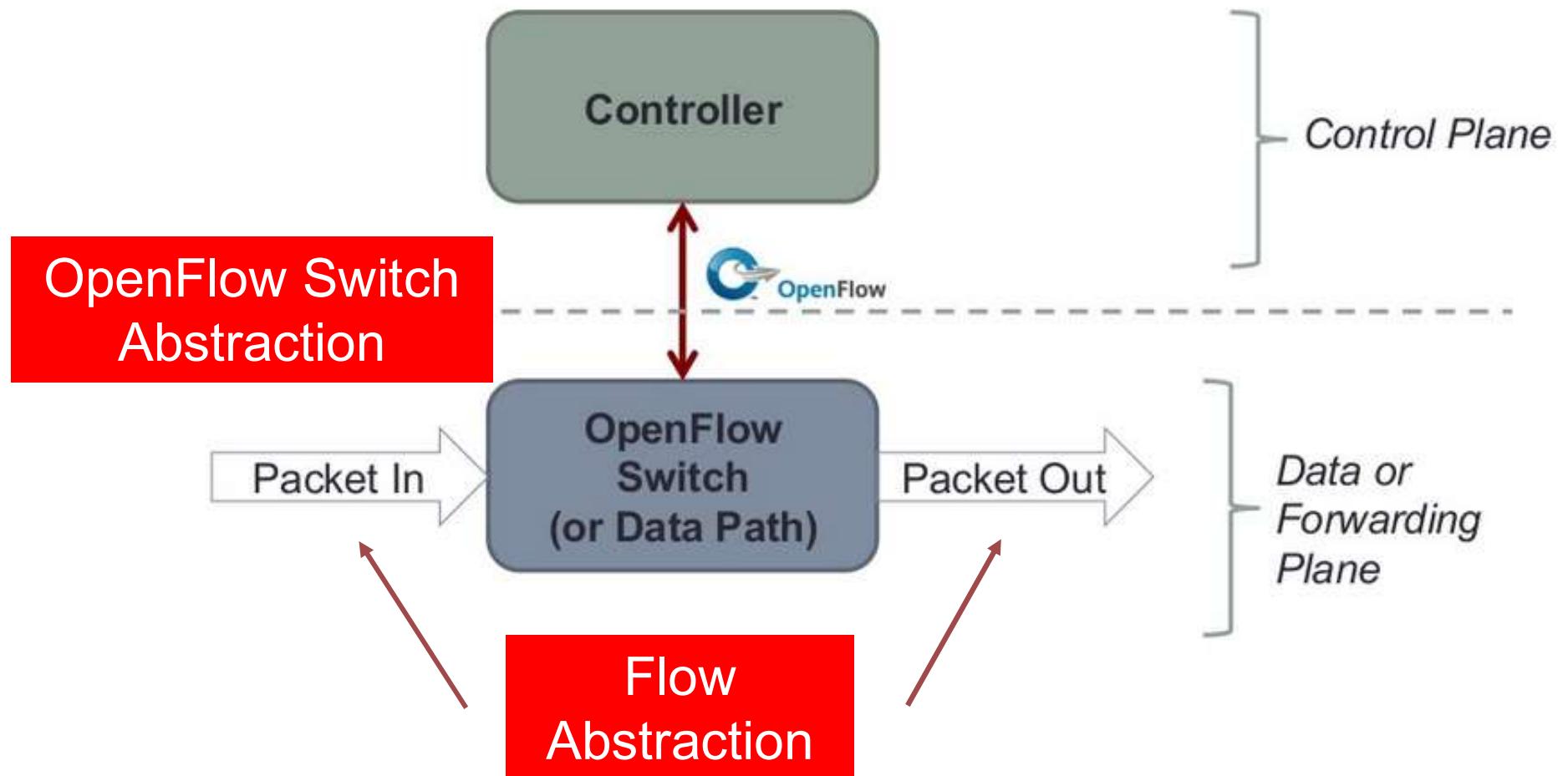
OpenFlow Terminology

术语	描述
Action	转发 packet 到某个端口或修改 packet 的一个操作，比如减小 TTL 值。Actions 也可作为 flow entry 对应的 instruction set 的一部分，或者在 group entry 对应的一个 action bucket 中。Actions 可以被添加到 packet 的 action set 中，也可以立即应用给匹配的 packet
Action set	与 packet 相关的一组 actions，当每个 flow table 处理 packet 时会添加 action set，当 instruction set 表示 packet 会送出 processing pipeline 时，action set 会被执行
Group	一个 action buckets 列表，从其中选择一个或多个 buckets 用在每个 packet 上
Action bucket	一组 actions 以及相关参数，在 group 中定义
Tag	通过 push 和 pop actions 插入 packet 或从 packet 删除的一个 header
Outermost tag	离 packet 起始位置最近的一个 tag
Controller	通过 Openflow 协议与 OFS 交互的设备
Meter	一个 switch 基本单位，能测量和控制 packets 速率，如果 packet 速率或 byte 速率 经过 meter 超出预定义的范围，meter 会激活一个 meter band，如果 meter band drop packet，就是所谓的 Rate Limiter

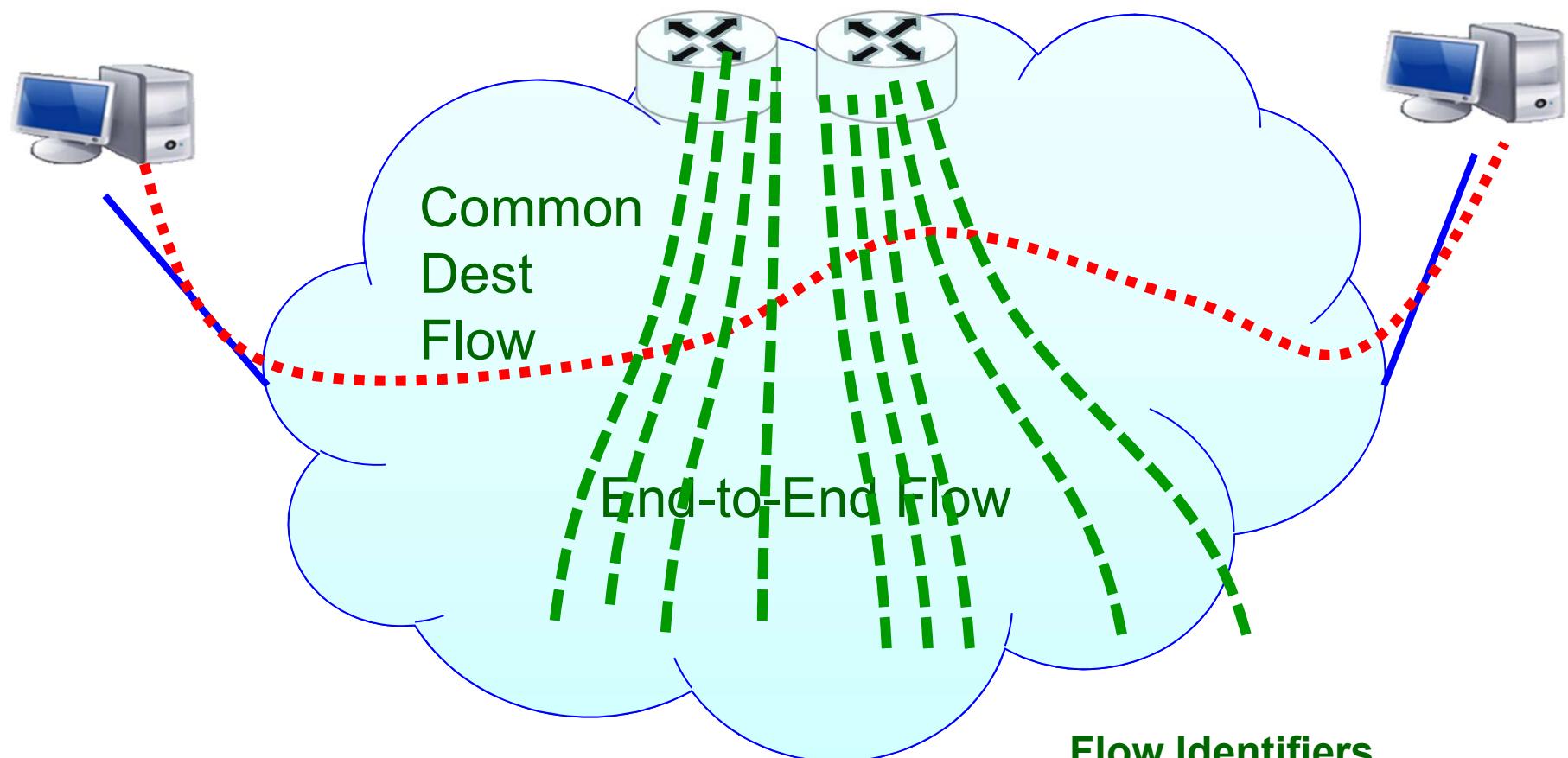
OpenFlow

- OpenFlow overview
- **Flow abstraction**
- OF-switch abstraction
- Flow table pipeline
- Group table & meter table
- OpenFlow protocol
- Modes of operations

OpenFlow = Open Interface + Flow



Flow abstraction



Flow Identifiers

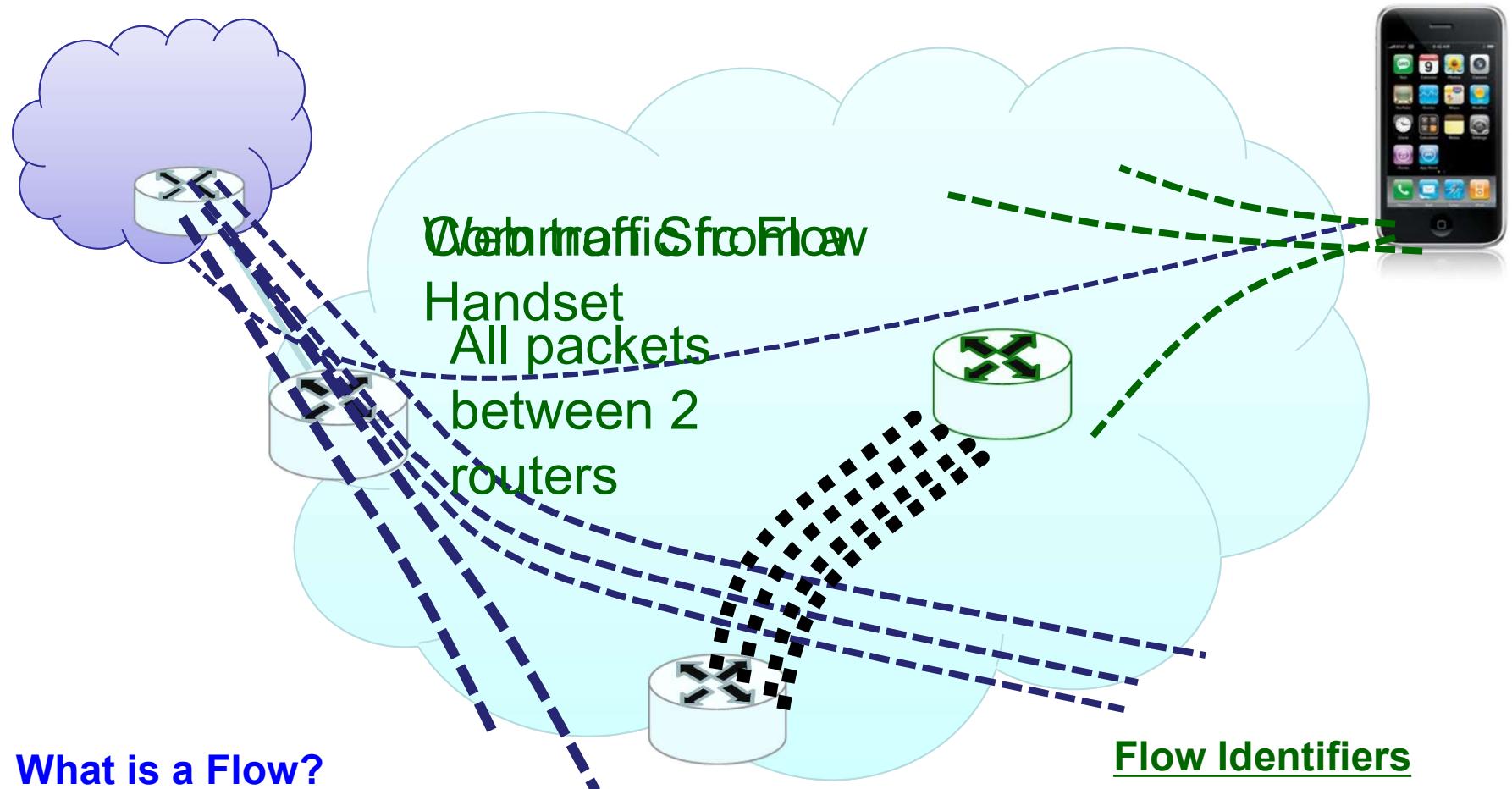
L4: TCP src/dst port

L3: IP dst/dst for L3proto

L2.5:

L2:

Flow abstraction



What is a Flow?

- Classification of packets that have a logical association
- Action & Maintaining Flow State
- Flow based Accounting & Resource Management

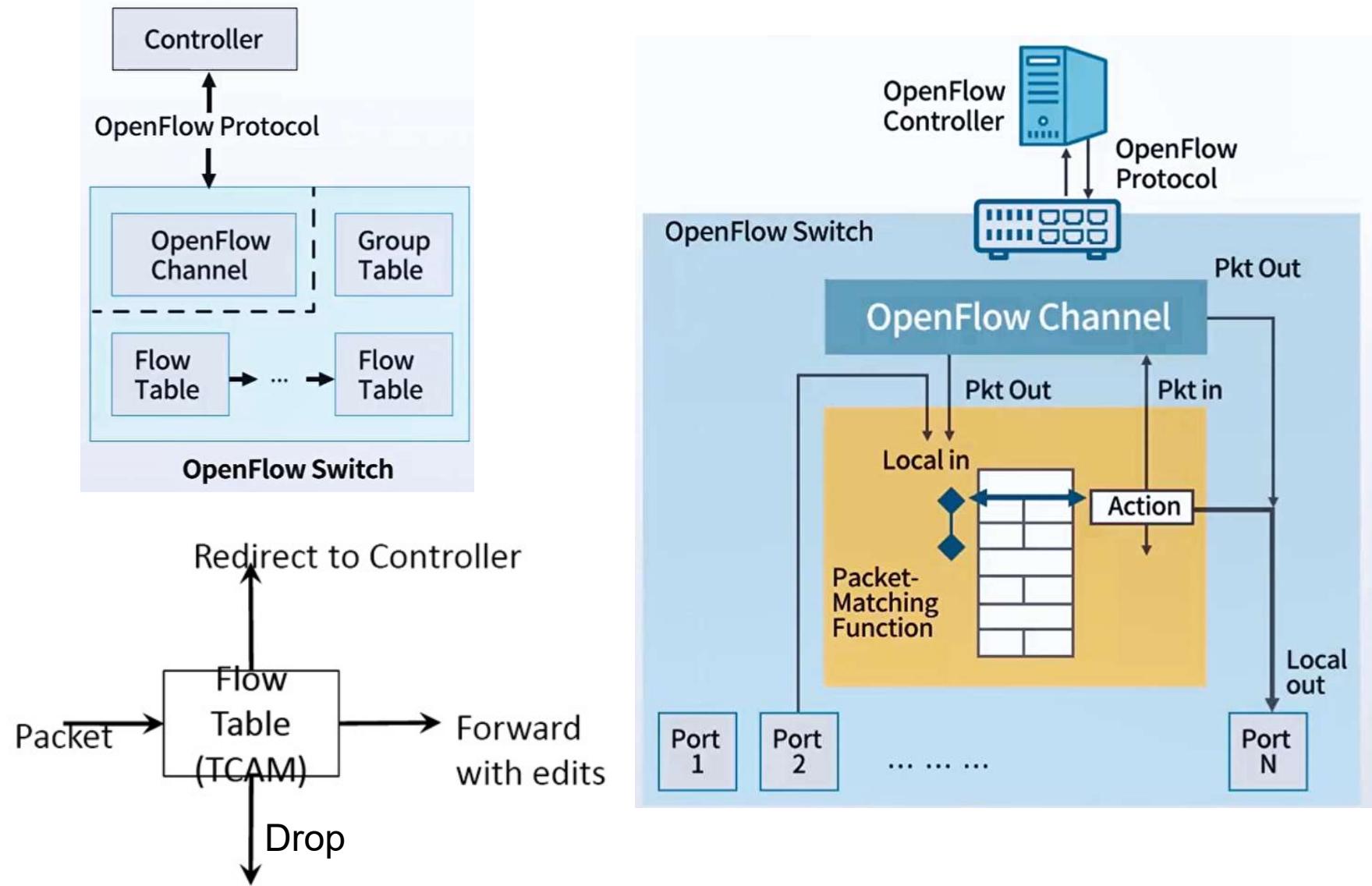
Flow Identifiers

- L4: TCP dst port 80
- L3: IP prefix for branch
- L2.5: MPLS Label ID
- L2: MAC src

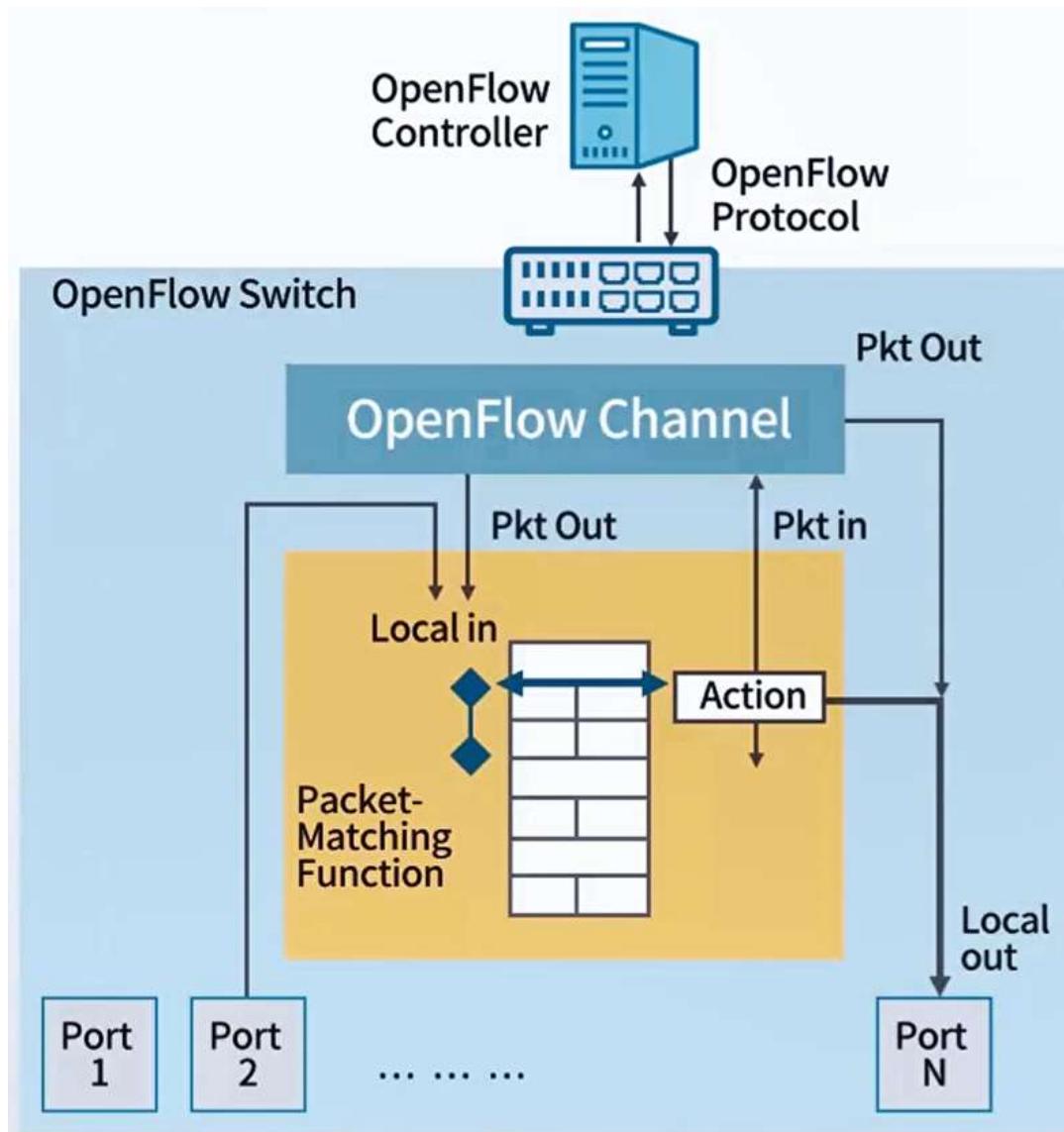
OpenFlow

- OpenFlow overview
- Flow abstraction
- **OF-switch abstraction**
- Flow table pipeline
- Group table & meter table
- OpenFlow protocol
- Modes of operations

OpenFlow switch abstraction

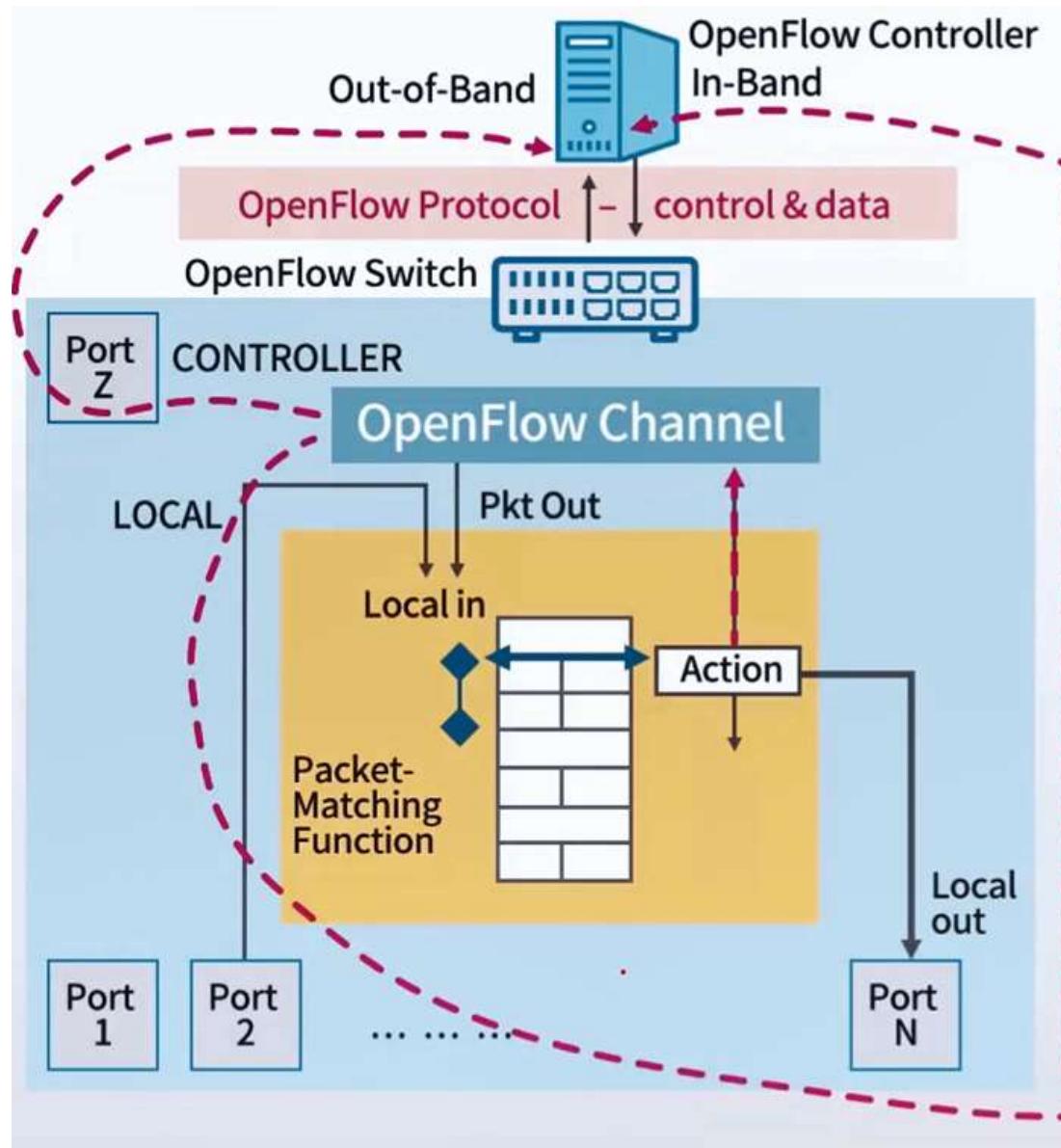


OpenFlow 1.0 switch (single table)



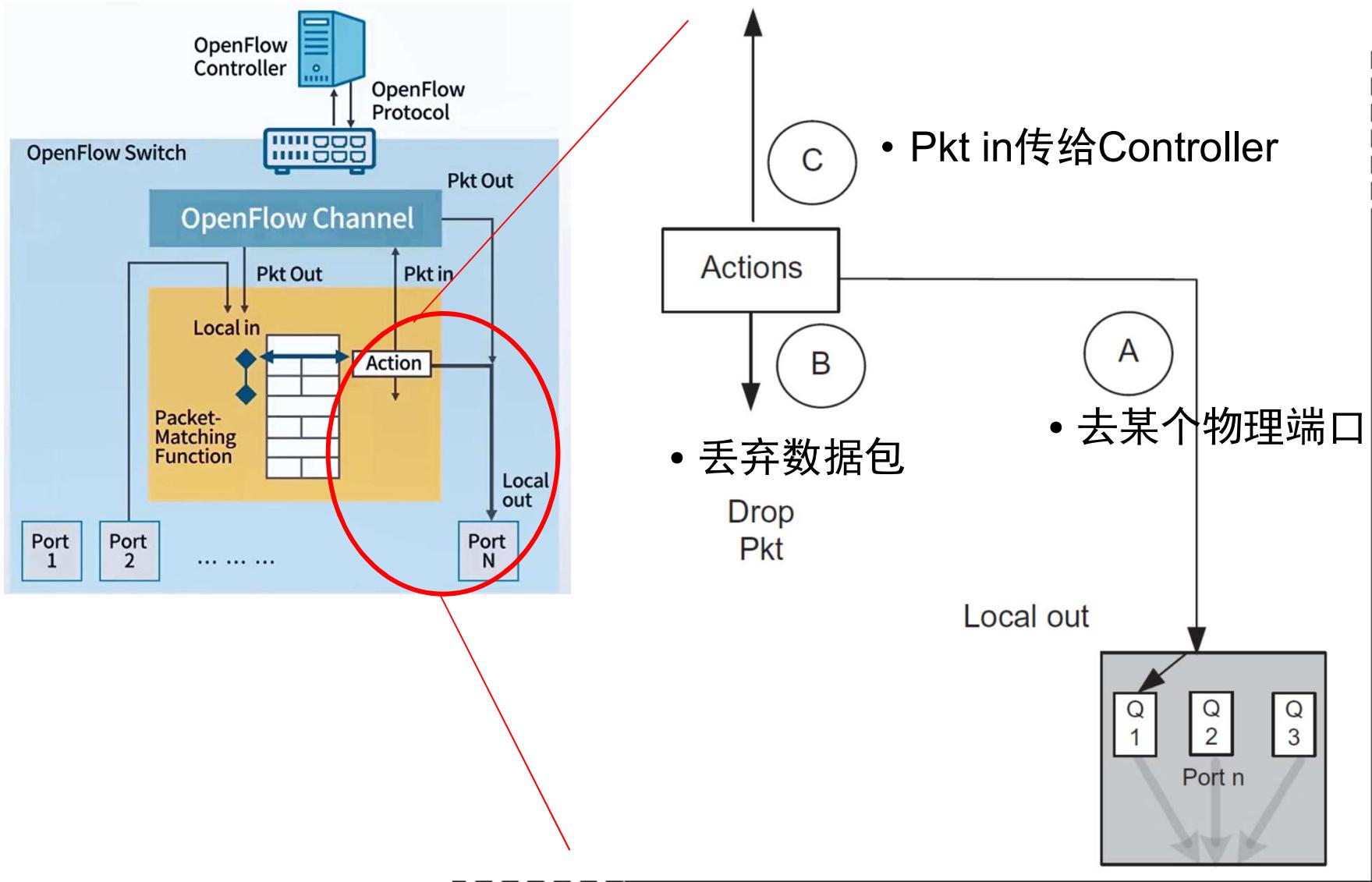
- Match
- Action
- Pkt-in/pkt-out
- Physical port
- Logical port
- Reserved port
 - All, Any, Local, Controller, ...

Controller-switch channel



- **In-band**
 - Local Port
 - TLS
- **Out-of-band**
 - Controller Port

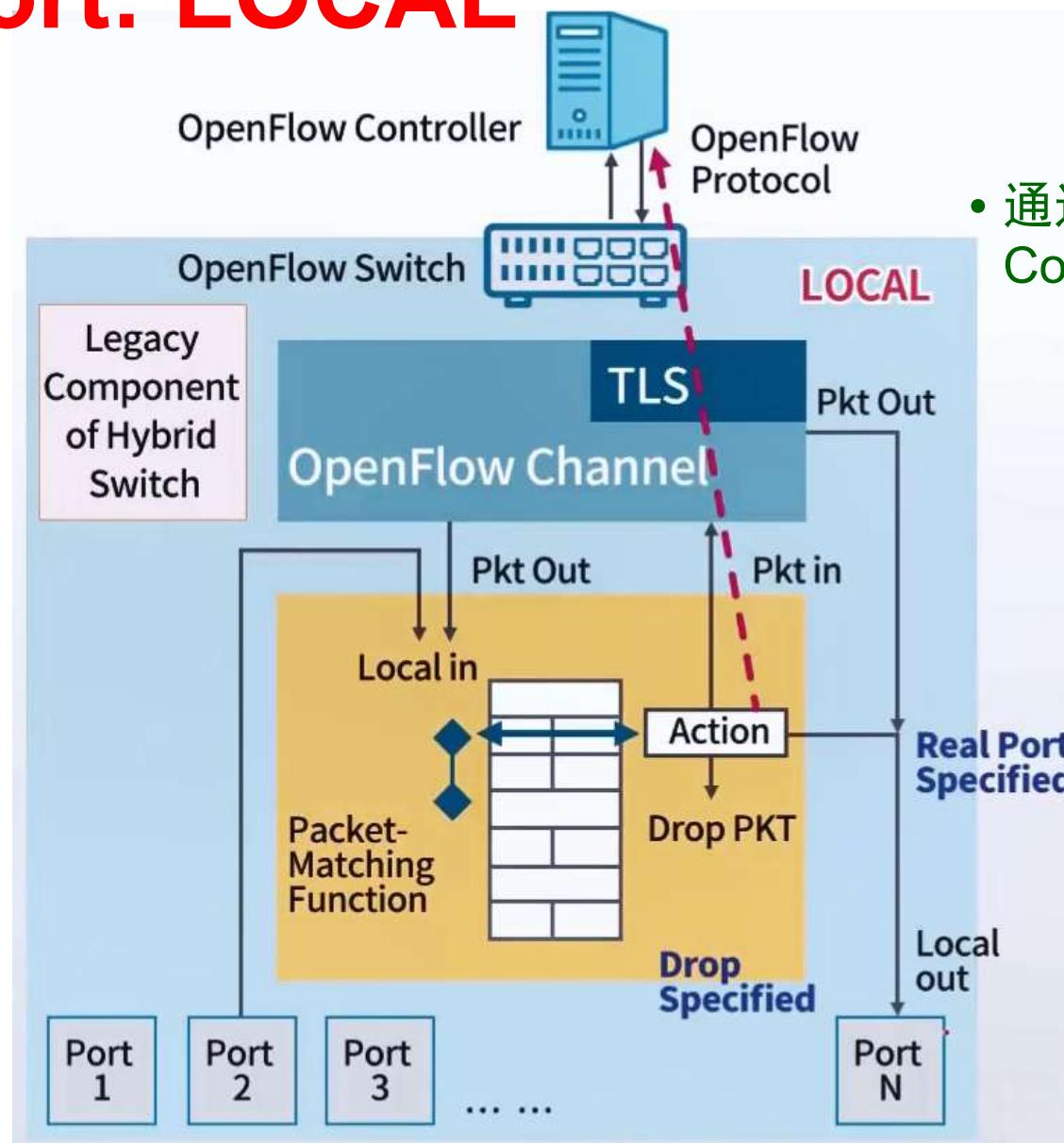
Multiple queues per port



Reserved ports

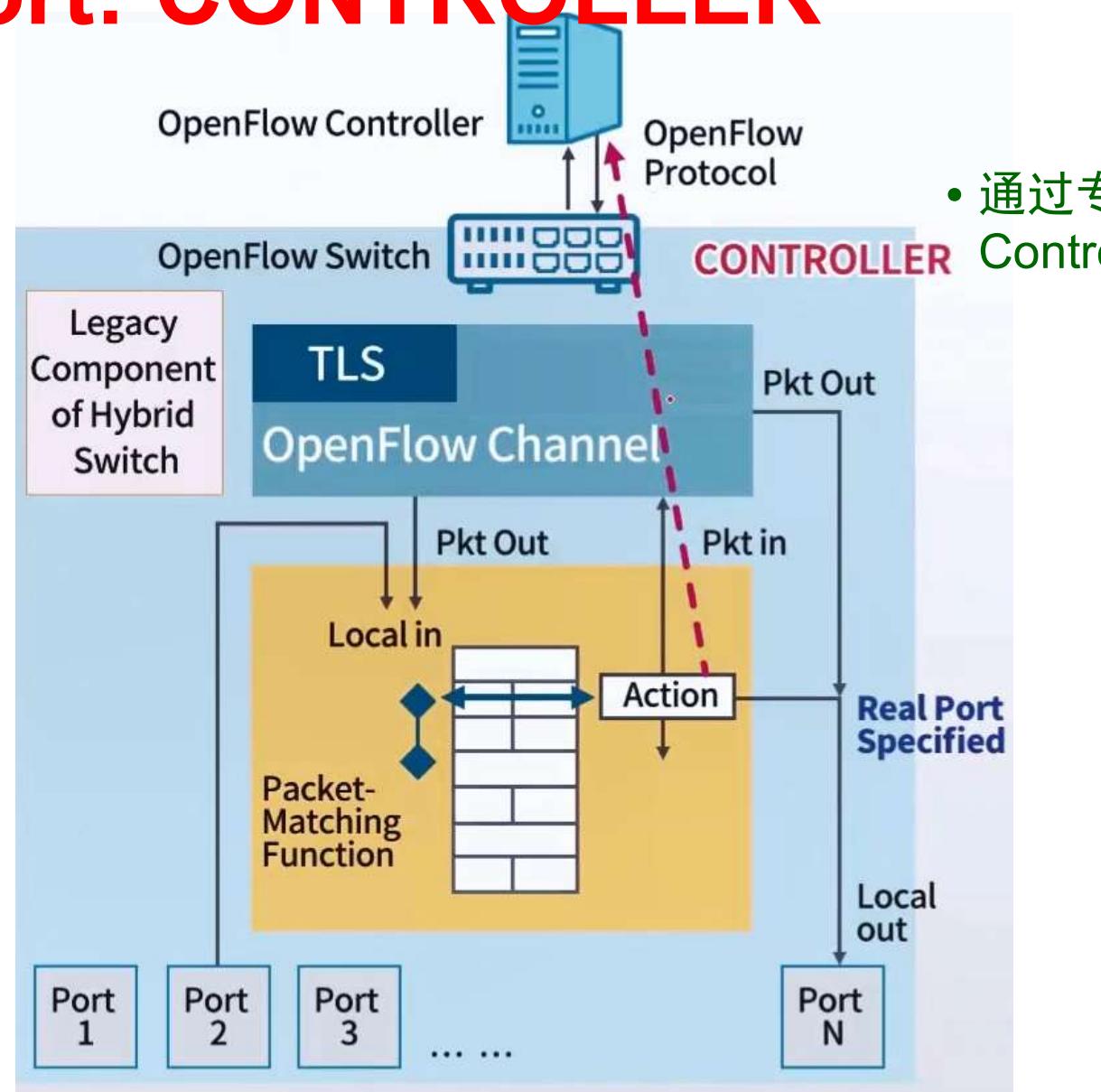
必须支持	描述
All	描述通用转发模型中能转发某个指定数据包的所有端口，只能用作输出端口。在这种情况下，这个数据包会被复制然后发送给所有的标准端口，当然不包括数据包的输入端口和配置为OFPPC_NO_FWD的端口
CONTROLLER	描述控制器的控制通道，可用作输入/输出端口，当用在输出端口，将数据包封装在packet-in消息中，按照协议规定的方式发送出去；当用在输入端口，标识一个来自控制器的数据包
TABLE	描述通用转发模型处理流水线的起始位置，只有在packet-out消息操作列表的output操作中才有效，将数据包提交个流水线的第一个flow table来处理
IN_PORT	描述数据包的输入端口，只能用作输出端口，发送数据包到自己的输入端口
ANY	当OpenFlow命令没有指定端口时使用的类型，不能用作输入端口和输出端口
可选支持	描述
LOCAL	描述通用转发模型本地的网络栈和管理栈，可用作输入/输出端口。使得远端设备通过OpenFlow网络本身与转发模型交互，使用其网络服务，而不是通过一个独立的控制网络。采用一些默认流选项可以实现网内控制器链接，不需要独立的控制器通道。
NORMAL	描述传统的非OpenFlow转发处理流水线，只能用作输出端口，使用传统流水线处理数据包。当转发模型不支持从OpenFlow流水线到传统处理流水线的转发时，必须指定不支持这种操作
FLOOD	描述传统处理流水线中的泛洪操作，只能用作输出端口，通常发送数据包给所有的标准端口，不包括输入端口和OFPPS_BLOCKED状态的端口。转发模型需要用数据包VLAN ID选择向那些端口执行flood操作

Rev. Port: LOCAL



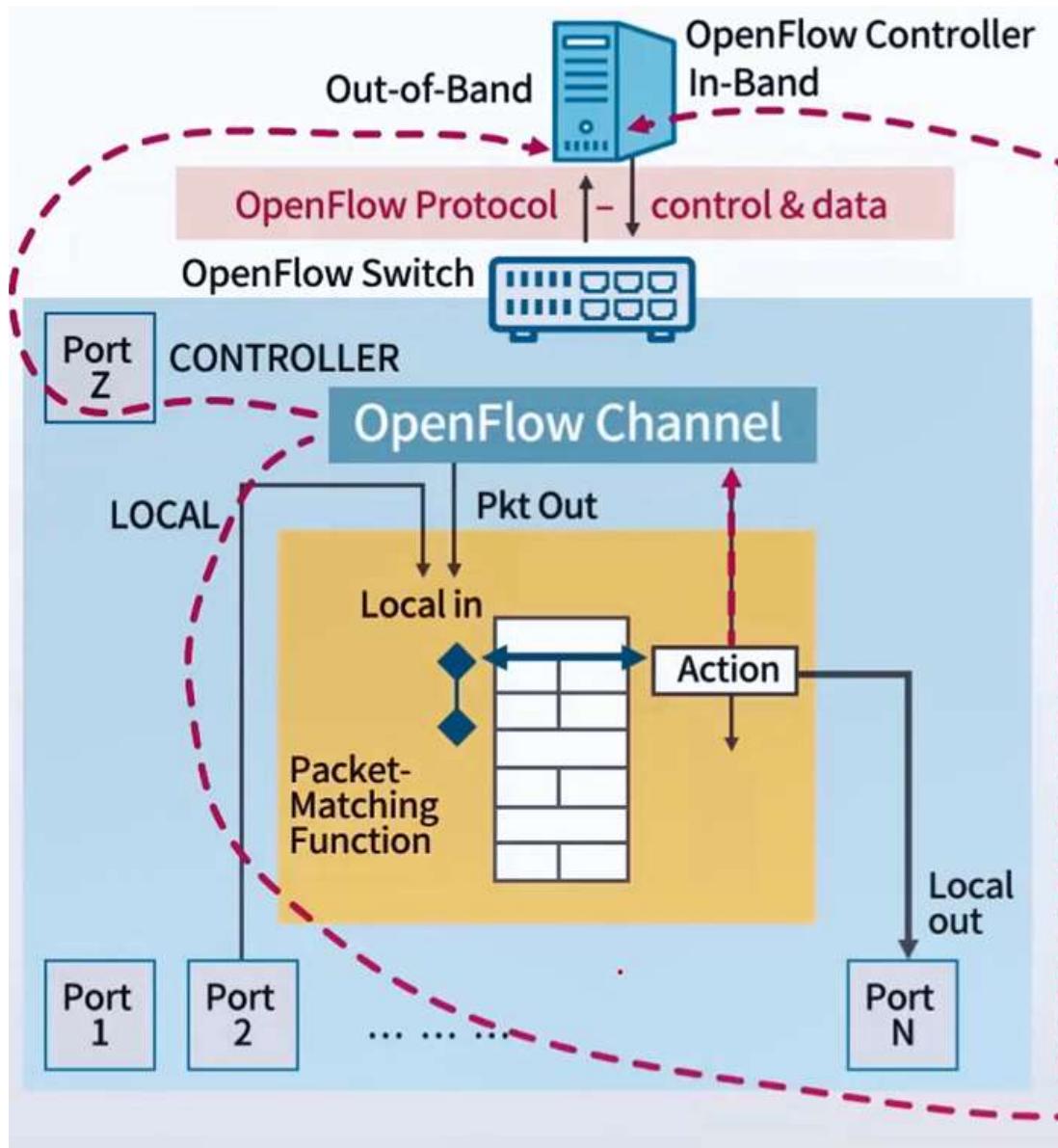
- 通过Local Port与Controller通信

Rev. Port: CONTROLLER



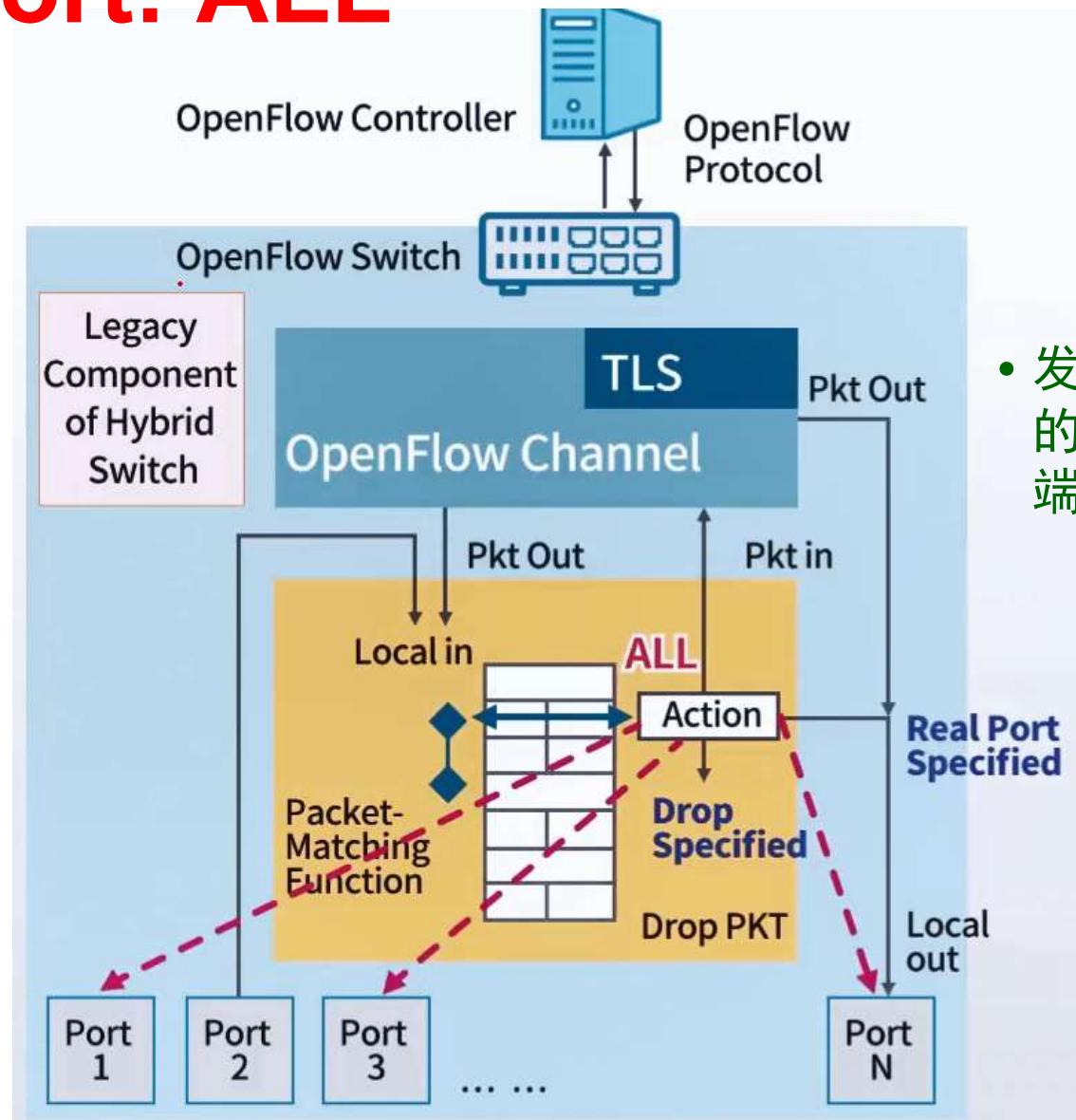
- 通过专用Port与Controller通信

Controller-switch channel



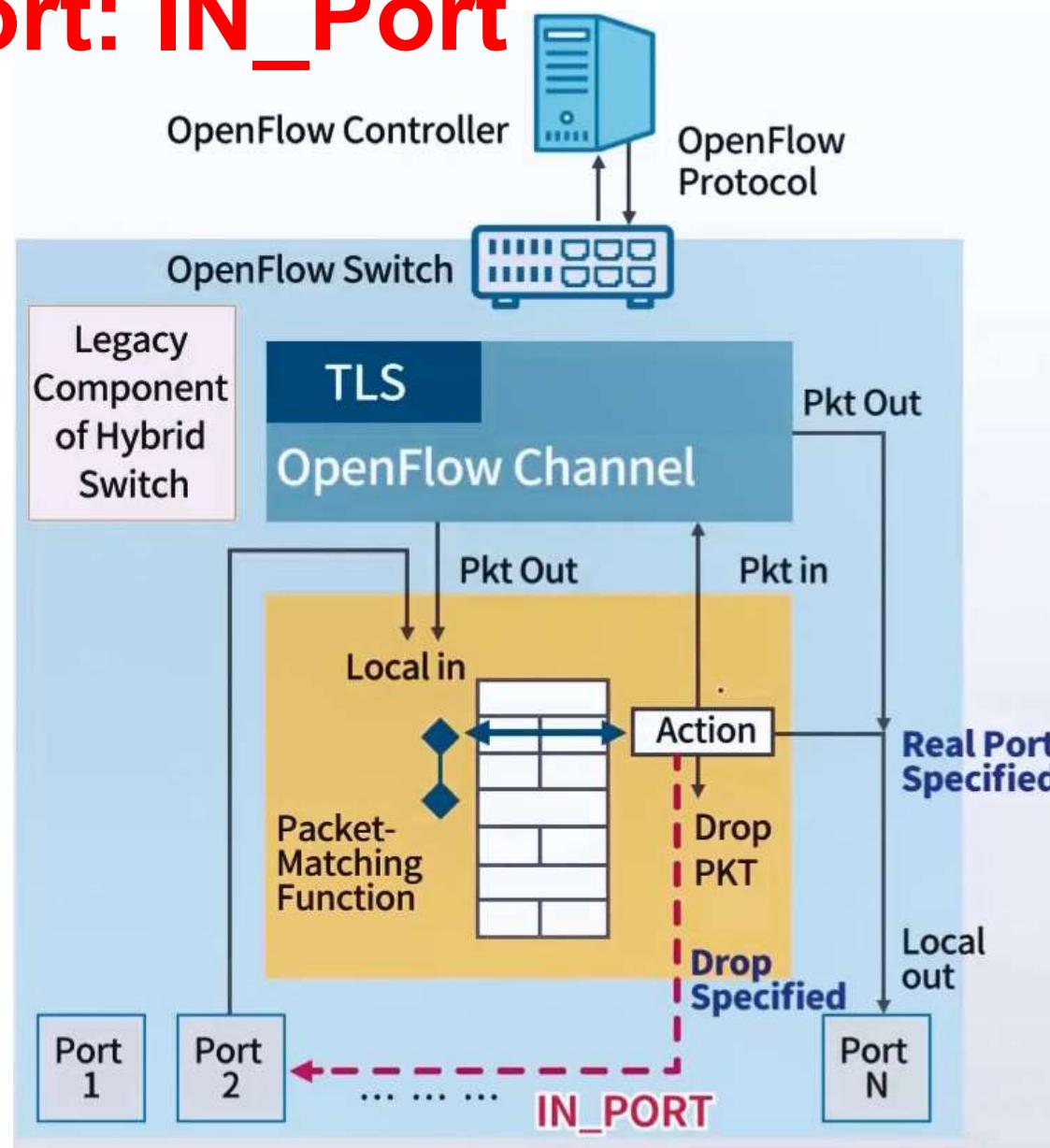
- **In-band**
 - Local Port
 - TLS
- **Out-of-band**
 - Controller Port

Rev. Port: ALL

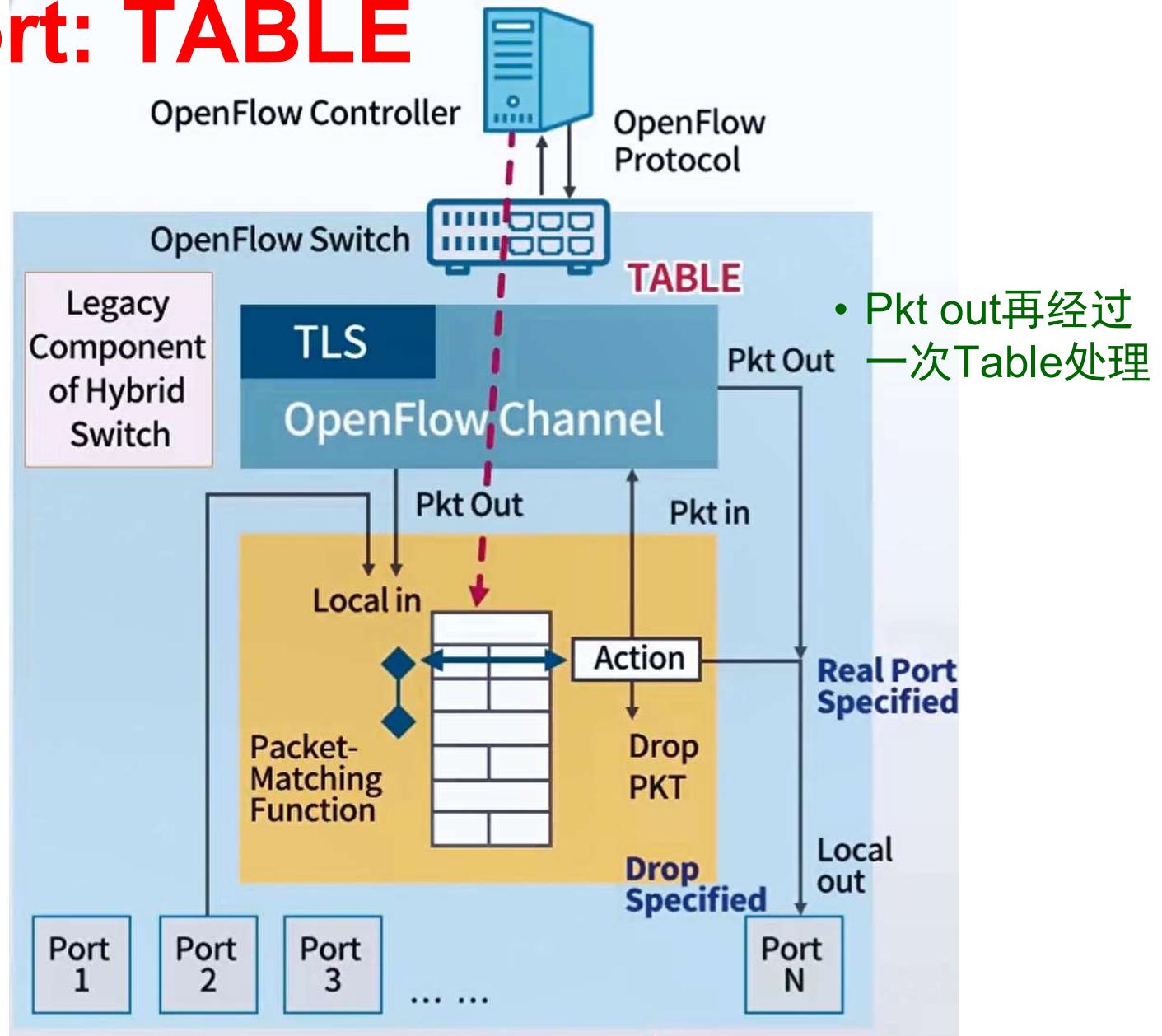


- 发给了除了In Port的所有其它物理端口

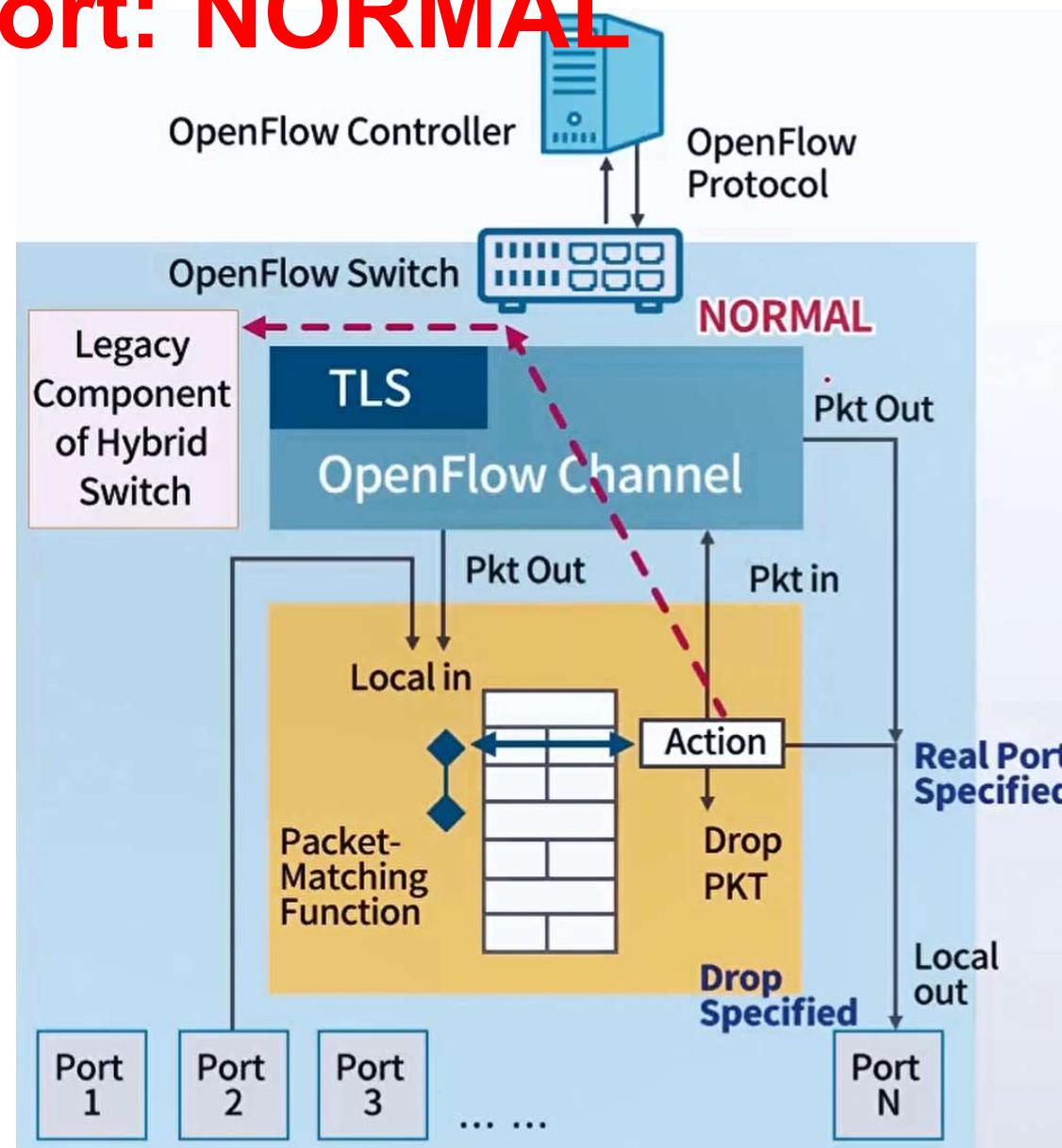
Rev. Port: IN_Port



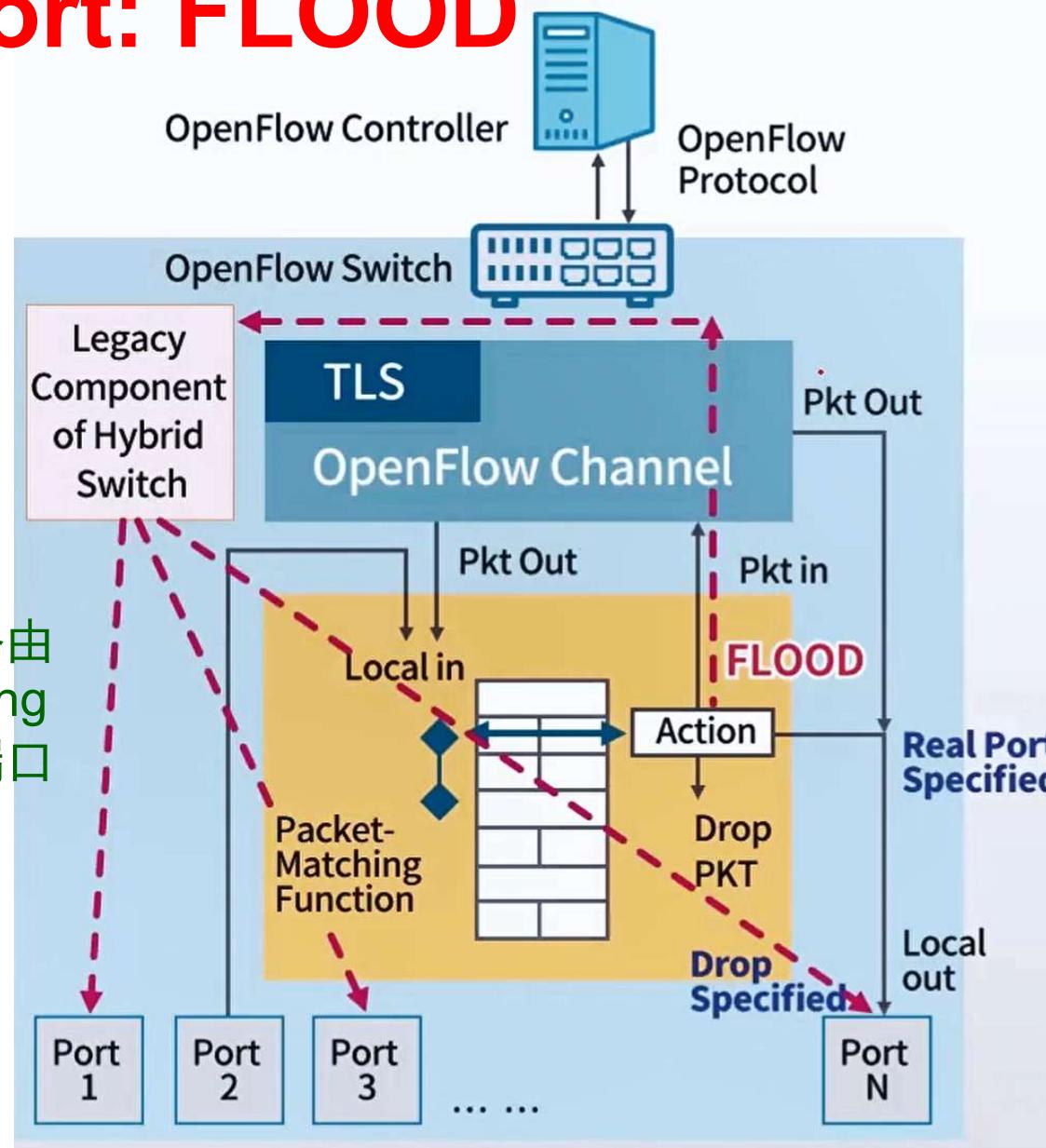
Rev. Port: TABLE



Rev. Port: NORMAL

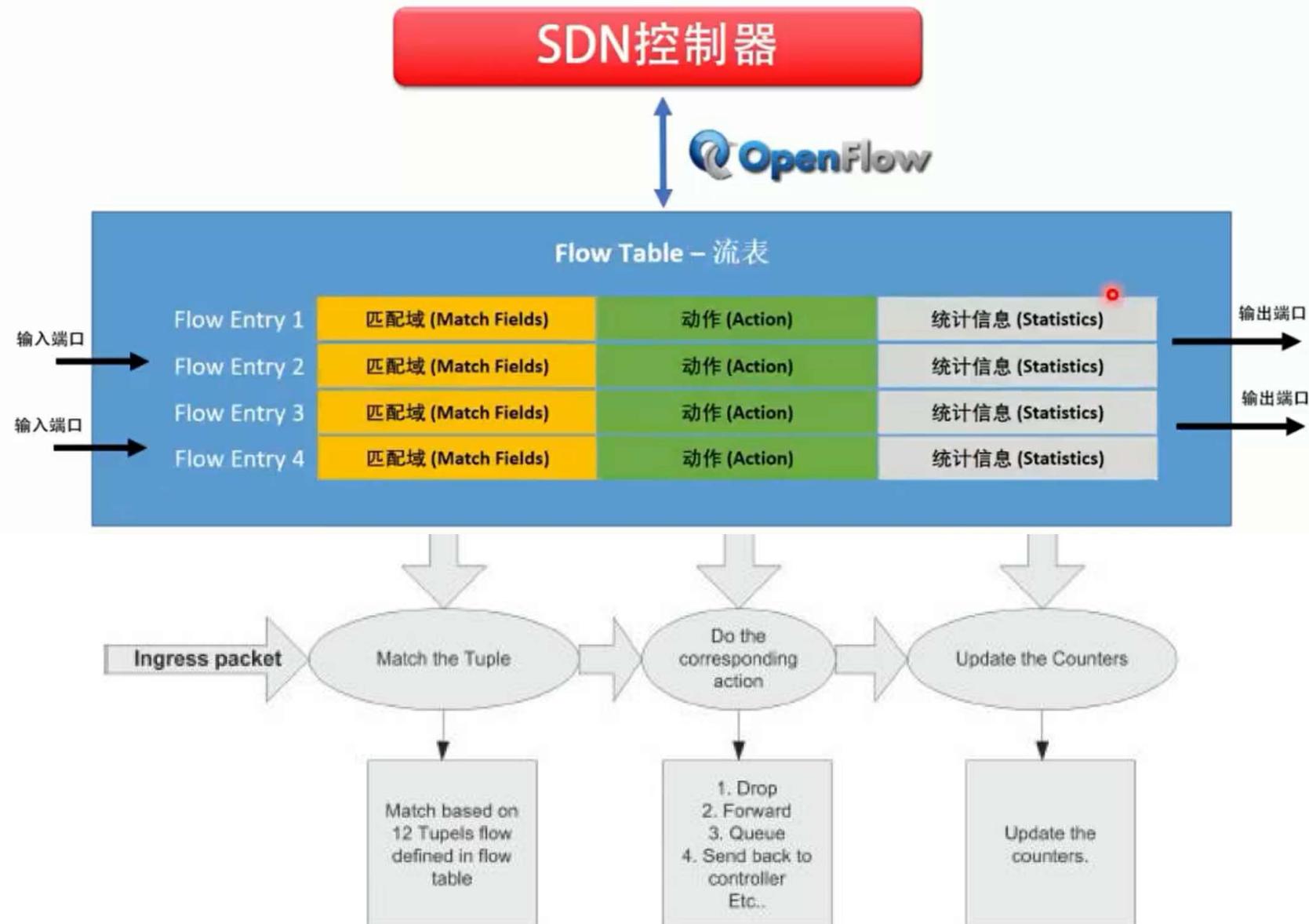


Rev. Port: FLOOD

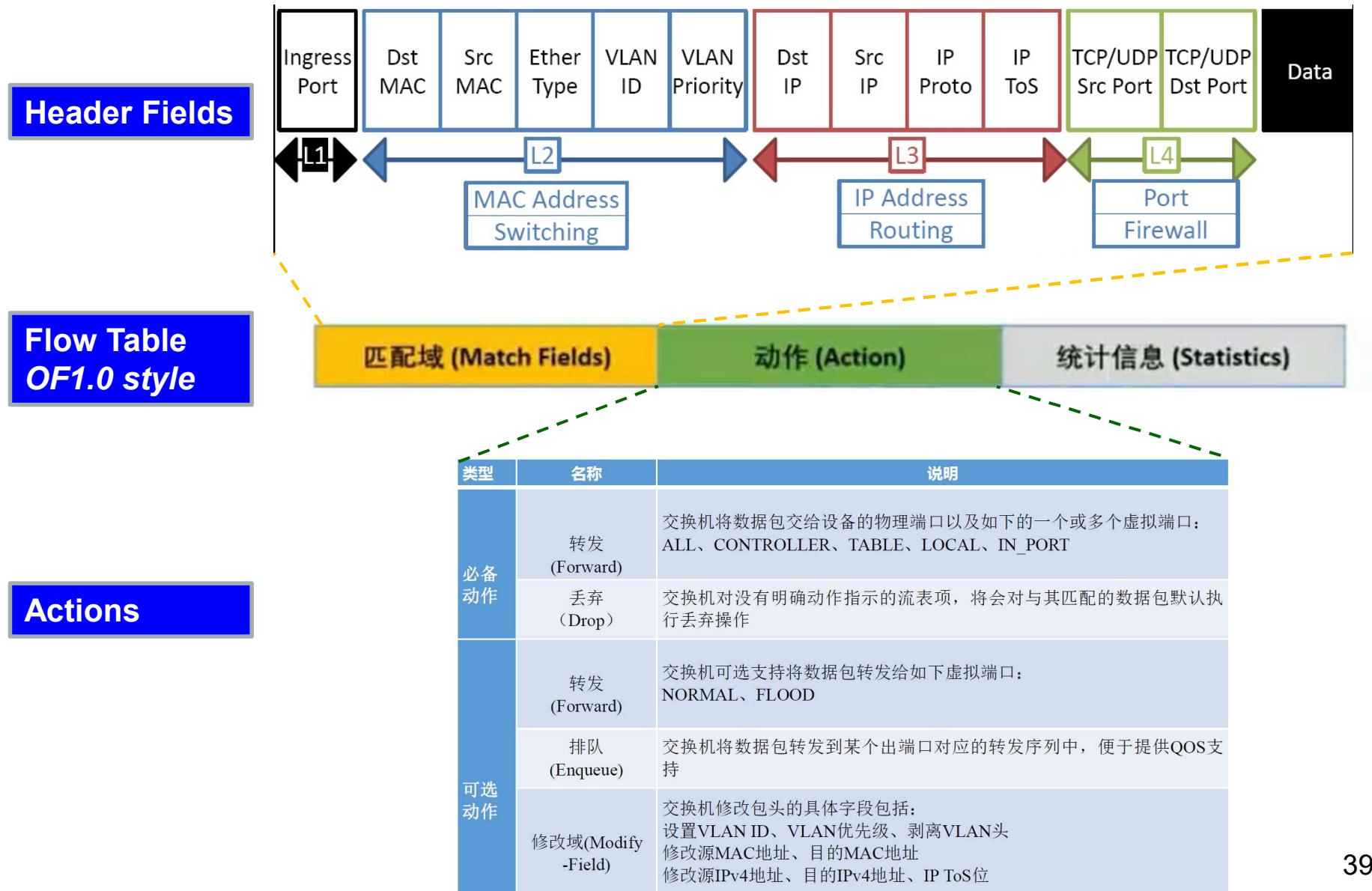


- 按照原先路由协议Flooding时的转发端口转发

Flow matching process

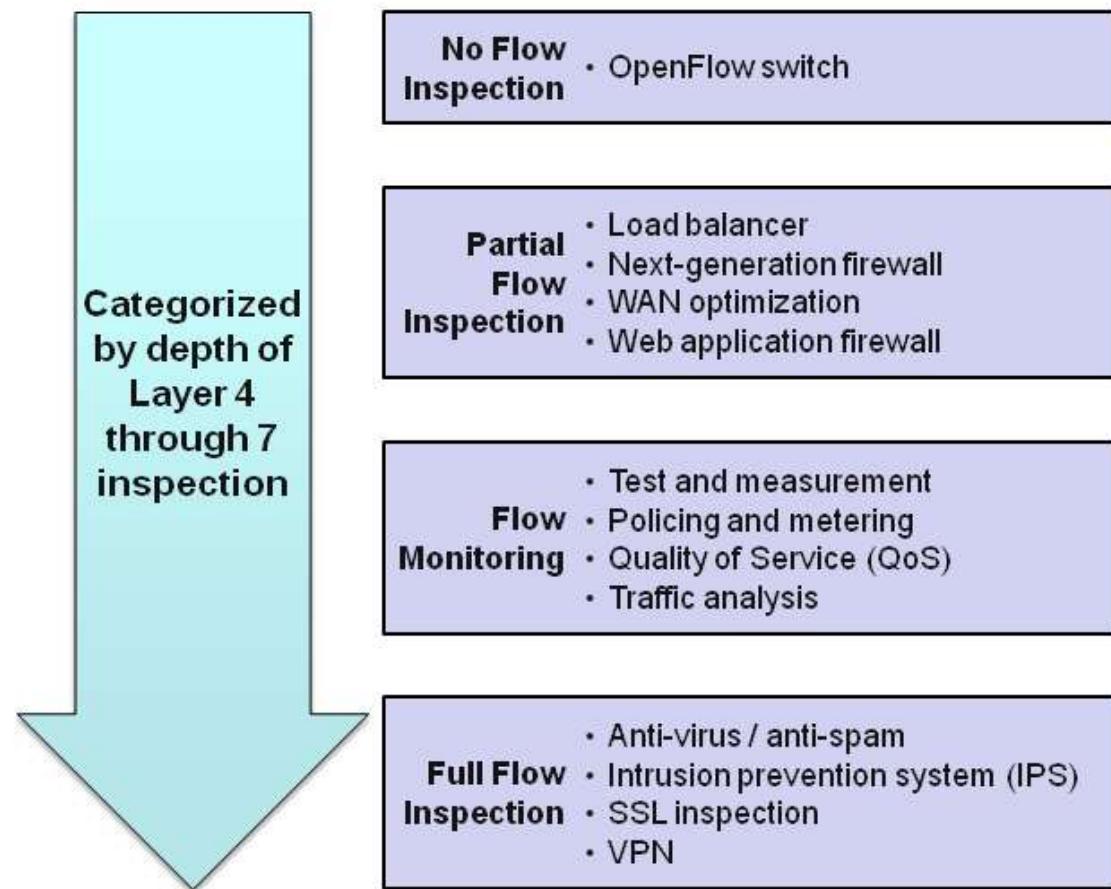


OpenFlow 1.0 flow table



Layer 1 to layer 4

- Layer 2 / Layer 3
 - Switching
 - Routing
 - Packet forwarding
 - OpenFlow
 - Architectures optimized to process individual packets
 - Cisco, HP, Juniper etc.
- Layer 4 through 7
 - Security
 - Load balancing
 - WAN optimization
 - Architectures optimized to process flows and content
 - F5, Riverbed, Sourcefire etc.



Match/action 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

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

Firewall

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

Match/action examples

VLAN Switching

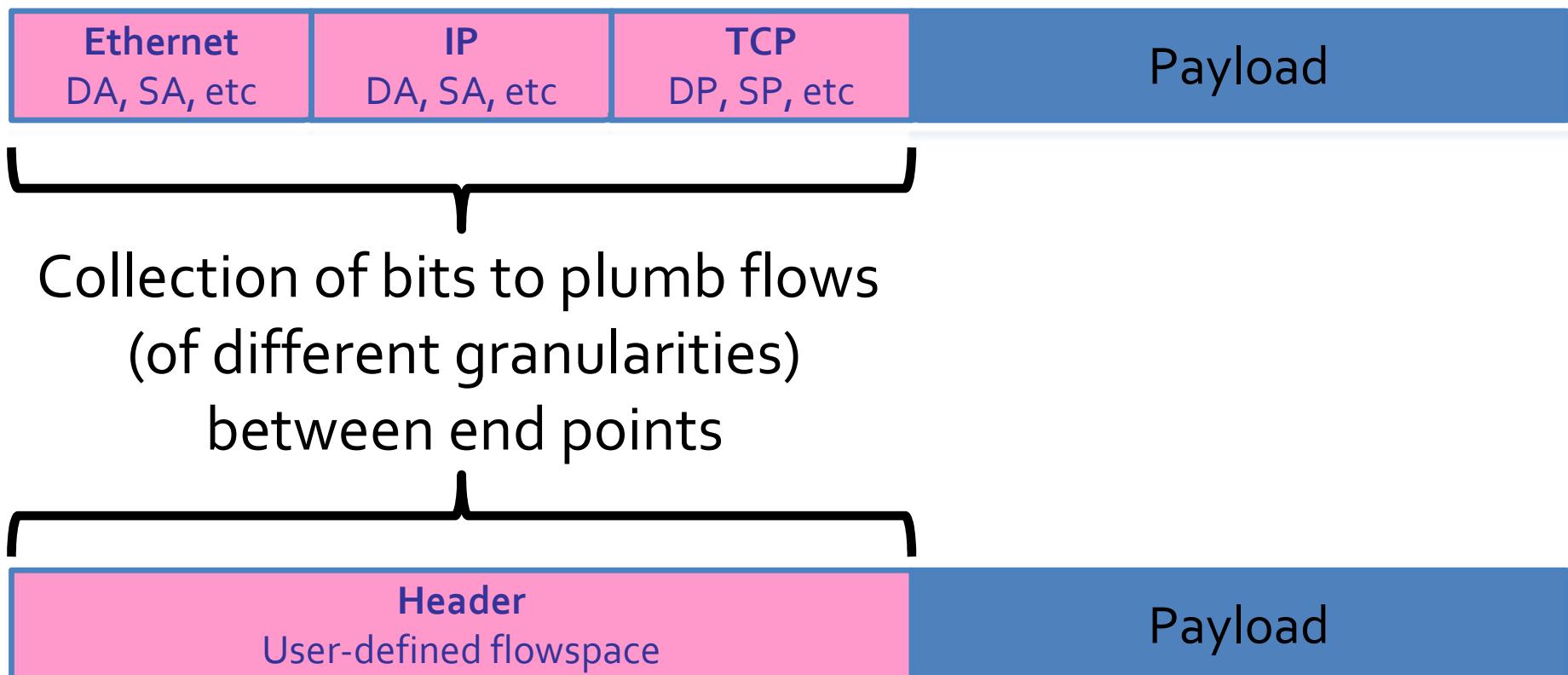
Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f..	*	vlan1	*	*	*	*	*	port6, port7, port9

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

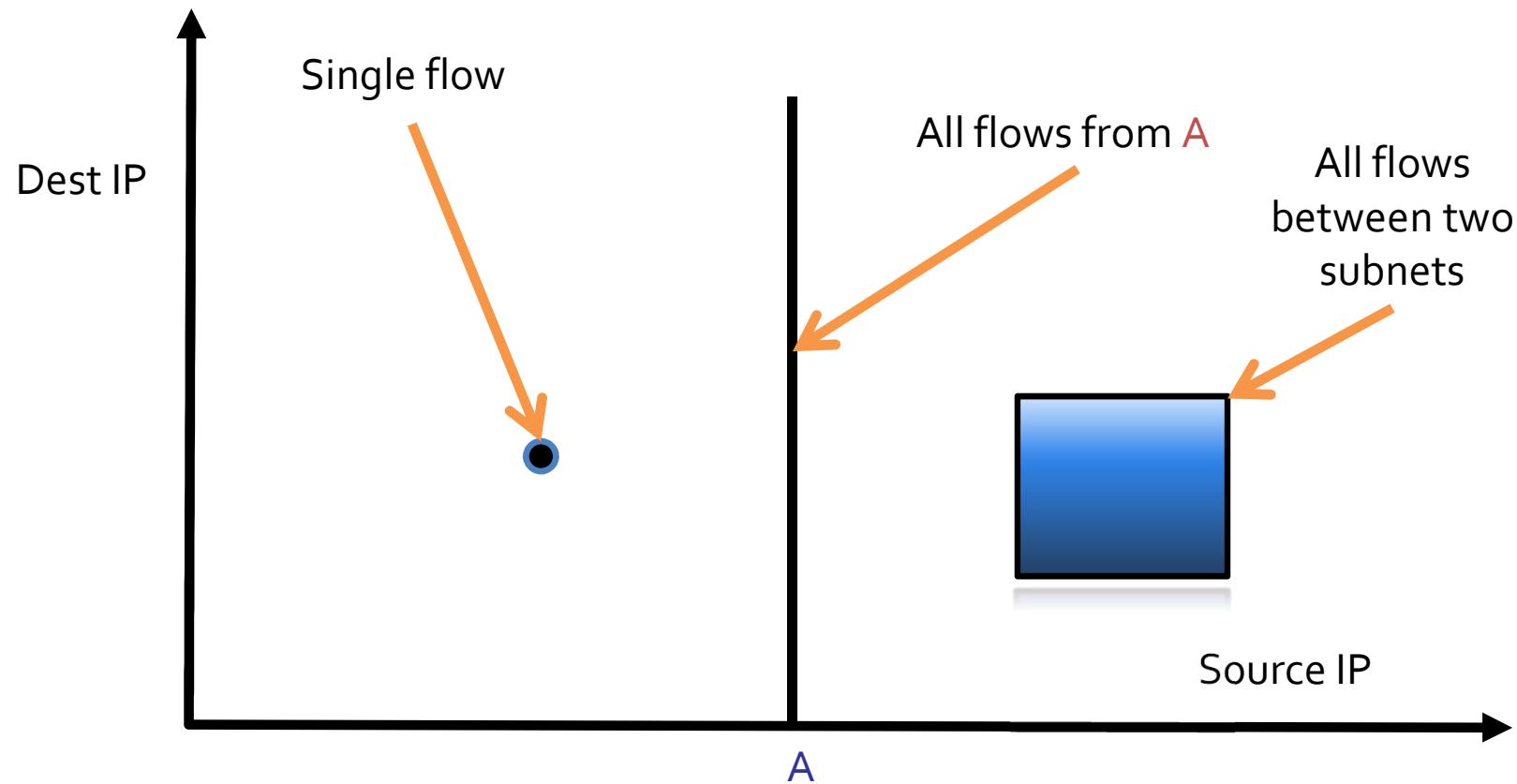
Flow-space

- Headers as a protocol-agnostic collection of bits

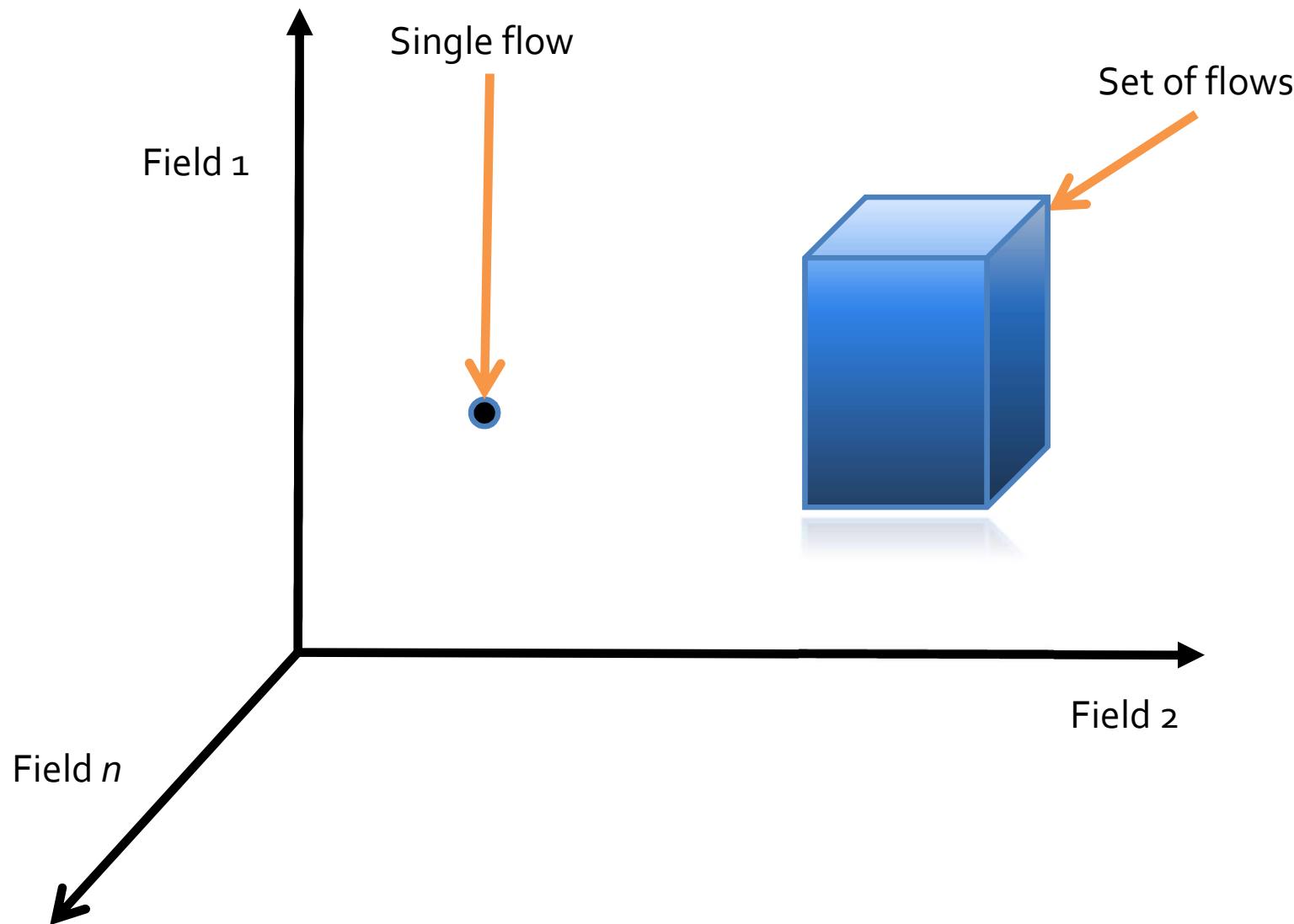


Flow-space

- Flow-space: a way to think about packets defined by match fields



Flow-space: multiple dimensions



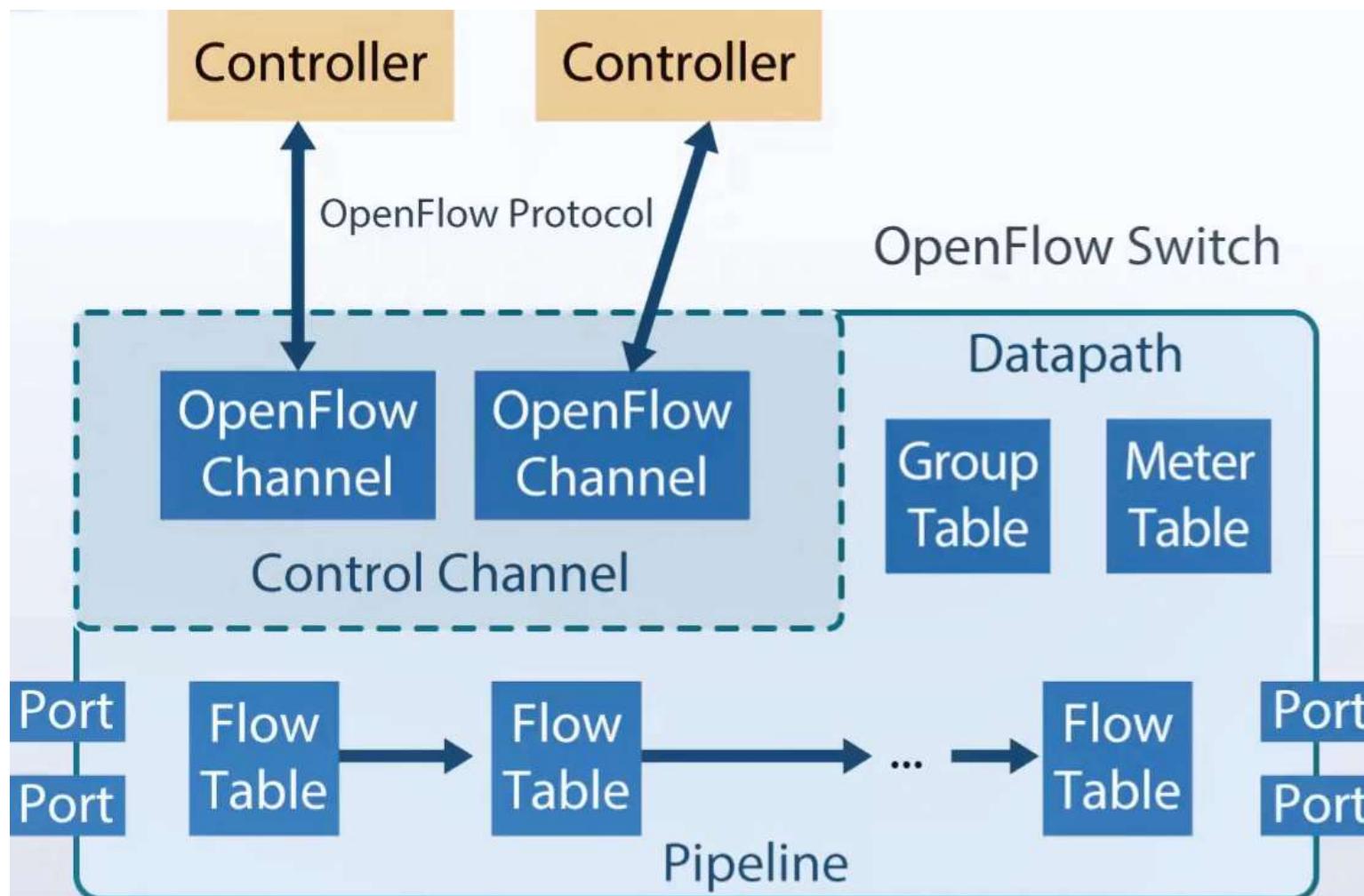
Characteristics of Flow Table

- Generalization of forwarding/switch table
- Easily implemented in hardware, e.g. using TCAMs
- “Flow-space”: Simple geometric construction that facilitates isolation of flows

OpenFlow

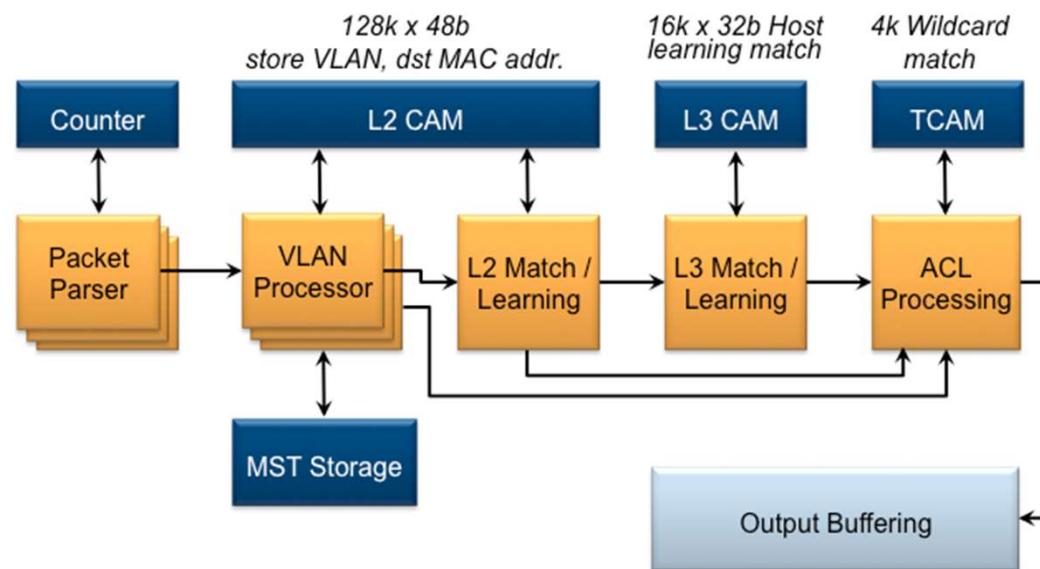
- OpenFlow overview
- Flow abstraction
- OF-switch abstraction
- **Flow table pipeline**
- Group table & meter table
- OpenFlow protocol
- Modes of operations

OF 1.1~1.5 (multiple flow tables)



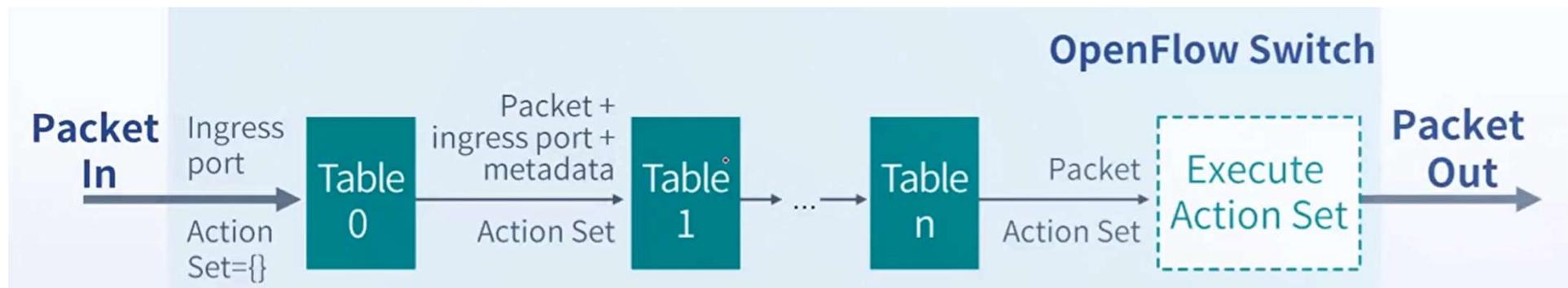
Multiple Tables

- Hardware has multiple tables internally (e.g. exact vs. wildcard match)
- Typical networking hardware layout:



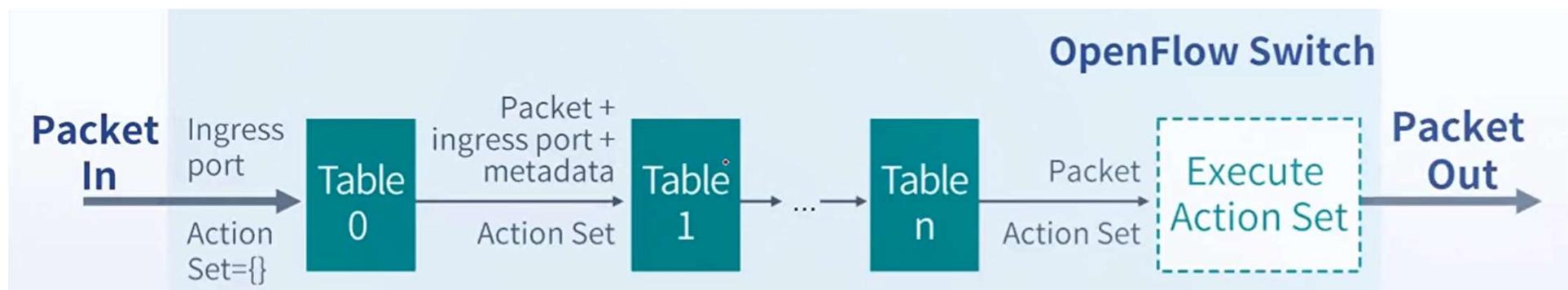
- Expose hardware for greater efficiency and flexibility

Flow table pipeline



- Separate **logical functions**, e.g.,
 - Table 1: Input firewall rules
 - Table 2: Network address translation
 - Table 3: Routing
- Reduce the number of flow entries in **TCAM**

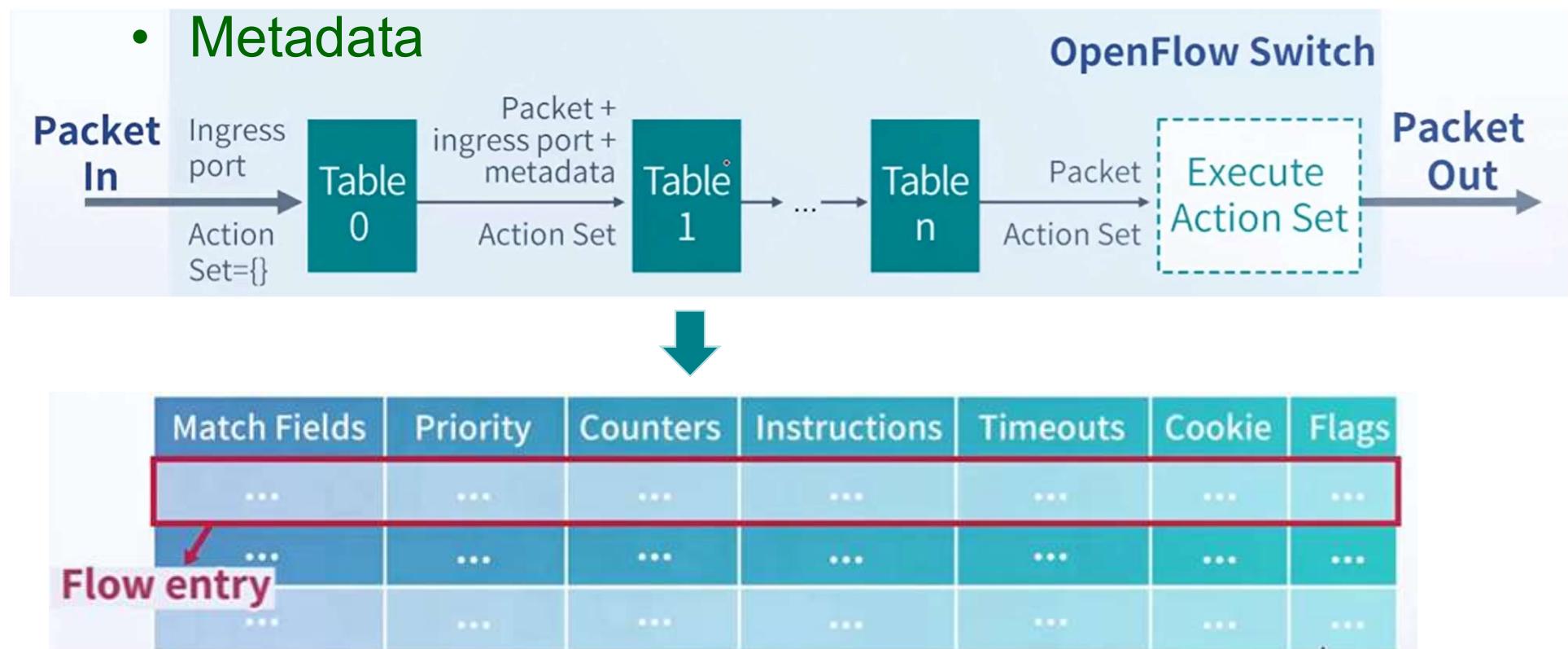
Flow table pipeline



- Generic **pipeline** that may be mapped to the hardware
- The **metadata** allow state to be carried between tables

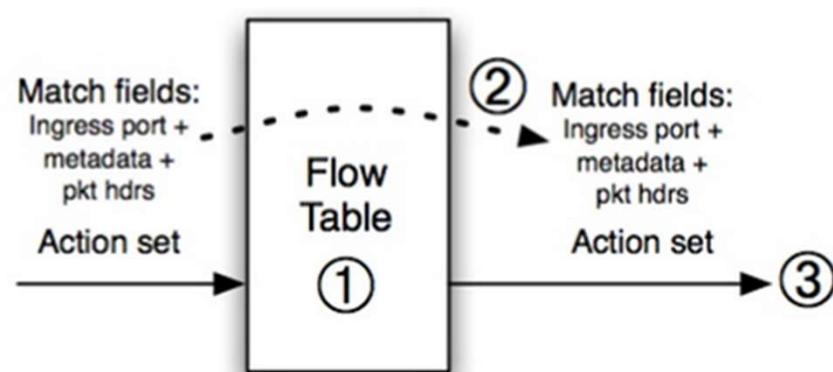
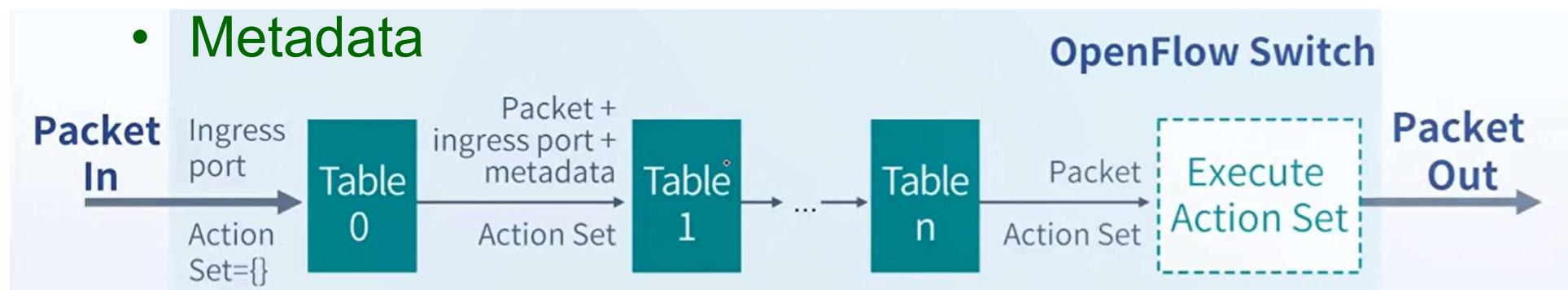
Flow table pipeline

- Multiple flow tables
- Action set
- Metadata



Flow table pipeline

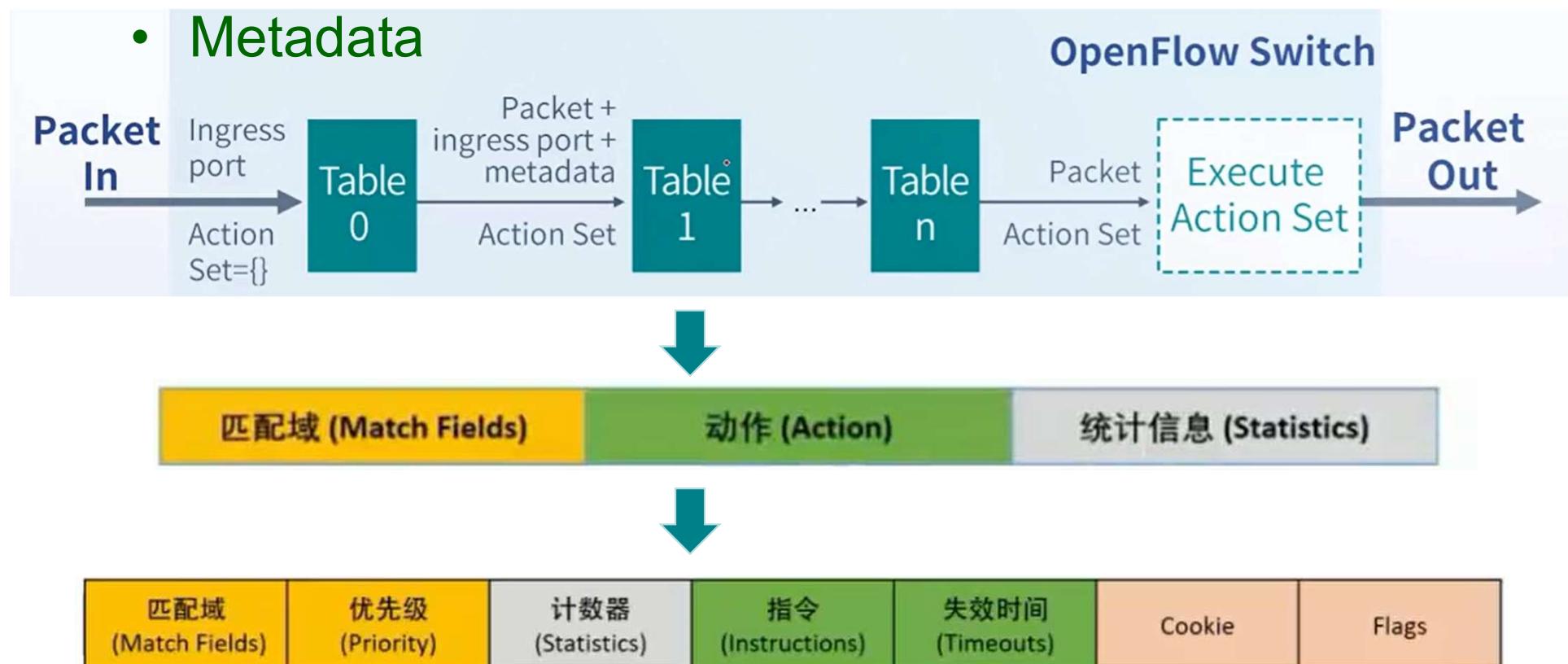
- Multiple flow tables
- Action set
- Metadata



- ① Find highest-priority matching flow entry
- ② Apply instructions:
 - i. Modify packet & update match fields (apply actions instruction)
 - ii. Update action set (clear actions and/or write actions instructions)
 - iii. Update metadata
- ③ Send match data and action set to next table

Flow table pipeline

- Multiple flow tables
- Action set
- Metadata



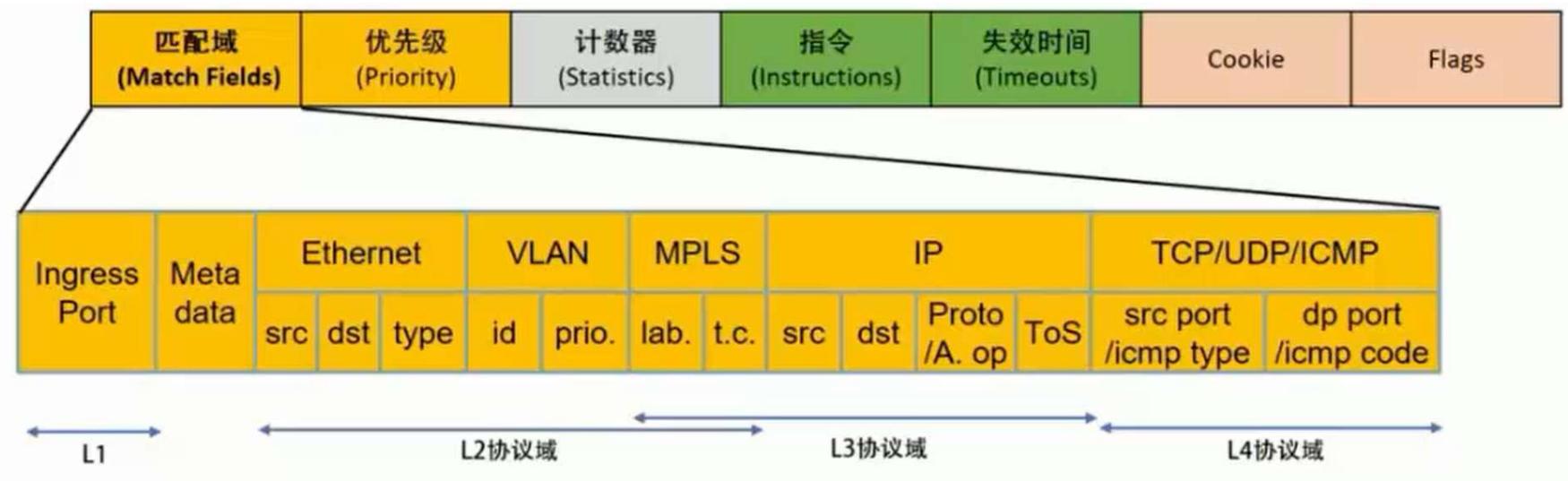
Flow table entries

匹配域 (Match Fields)		动作 (Action)		统计信息 (Statistics)		
匹配域 (Match Fields)	优先级 (Priority)	计数器 (Statistics)	指令 (Instructions)	失效时间 (Timeouts)	Cookie	Flags

Main components of a flow entry in a flow table

Match fields	To match against packets. These consist of the ingress port and packet headers
Priority	Matching precedence of the flow entry
Counters	e.g. packet and byte counters
Instructions	Determine action set or pipeline processing
Timeouts	Maximum amount of time or idle time before flow is expired by the switch
Cookies	Opaque data value chosen by the controller. Not used when processing packets.

Match fields



The match field contains either a specific value or a “wildcard”

- Ingress port
- Packet headers
- Optional metadata from previous table

Match fields

匹配域 (Match Fields)	优先级 (Priority)	计数器 (Statistics)	指令 (Instructions)	失效时间 (Timeouts)	Cookie	Flags
-----------------------	-------------------	---------------------	----------------------	--------------------	--------	-------

- 比如IP查找时最长前缀匹配，会出现多条entry匹配上

定义这个流表项的匹配优先级，当同时有多条表项匹配时

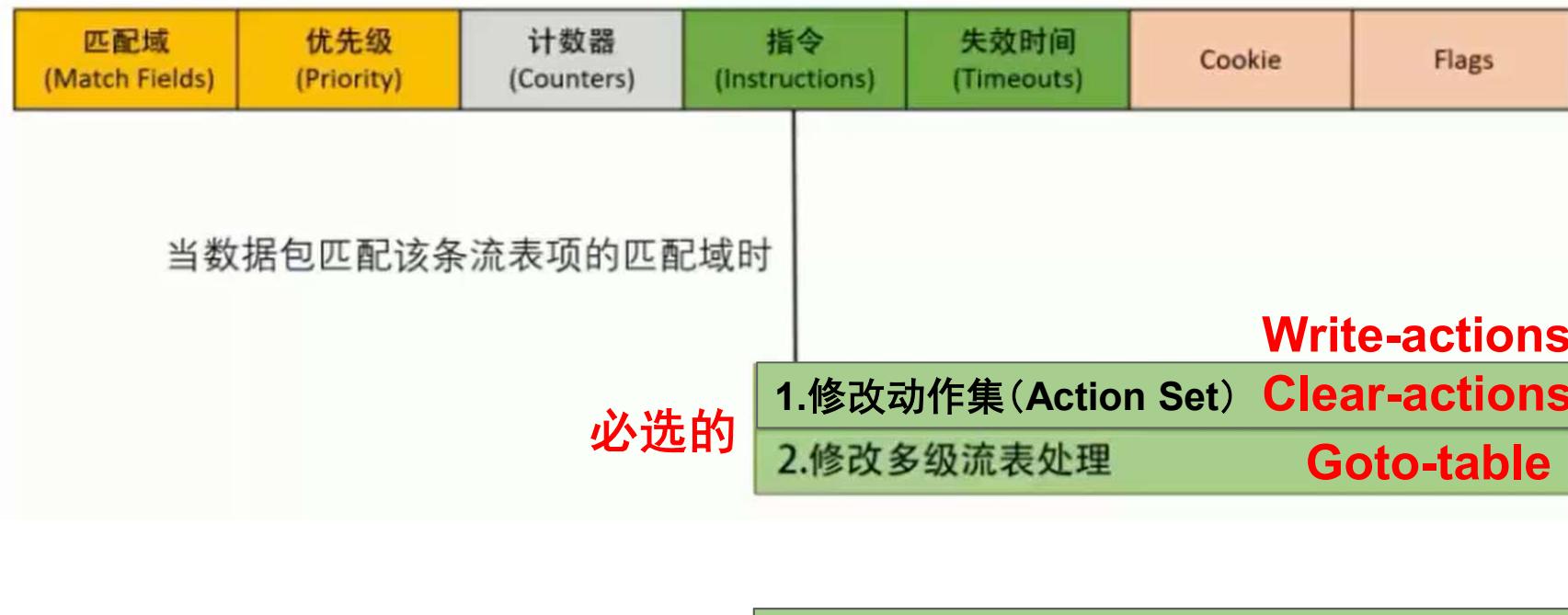
+

一组网络数据包协议域的组合，用来识别该条表项对应的Flow，也叫待匹配内容

||

Flow ID

Instructions



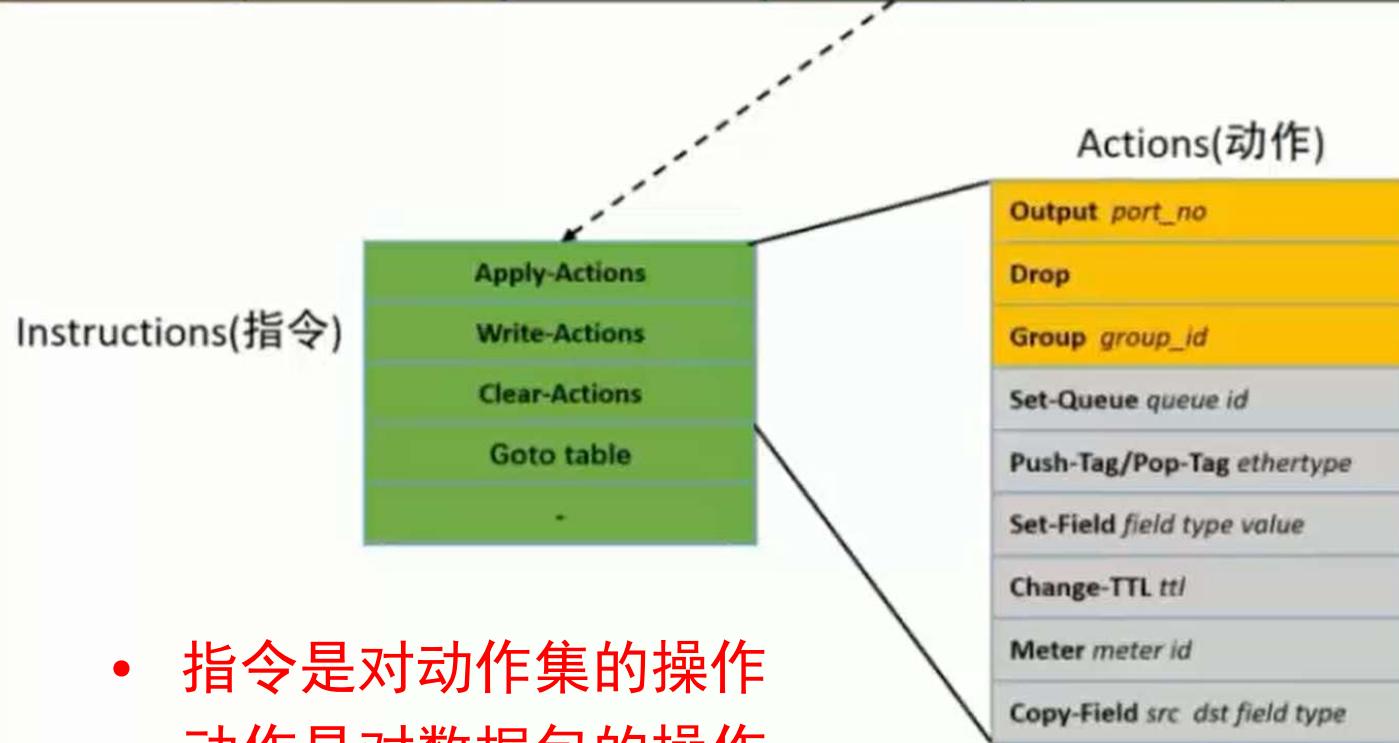
Instructions

必须支持	描述
Goto-Table <i>next-table-id</i>	指示processing pipeline中的下一个table，其中table-id必须大于当前的table-id，即向后跳转，最后一个table不能包含该Instruction
Write-Actions <i>actions</i>	把指定的actions添加到当前的action set，如果当前set中存在系统类型的action，就重写并添加
Clear-Action	立即清除action set中的所有actions
可选支持	描述
Apply-Actions <i>actions</i>	立即执行指定的actions，不对action set做任何修改 用于修改两个table之间的packets，或执行多个相同类型的actions
Write-Metadata <i>metadata/mask</i>	配置metadata域中的masked metadata value，这个mask指定metadata寄存器中的哪些bit应该被修改
Stat-Trigger <i>stat thresholds</i>	Generate an event to the controller if some of the flow statistics cross one of the stat threshold values(阈值)
Experimenter instructions	自定义扩展指令

Instructions

- Idle-timeout
- Hard-timeout

匹配域 (Match Fields)	优先级 (Priority)	计数器 (Counters)	指令 (Instructions)	失效时间 (Timeouts)	Cookie	Flags
-----------------------	-------------------	-------------------	----------------------	--------------------	--------	-------

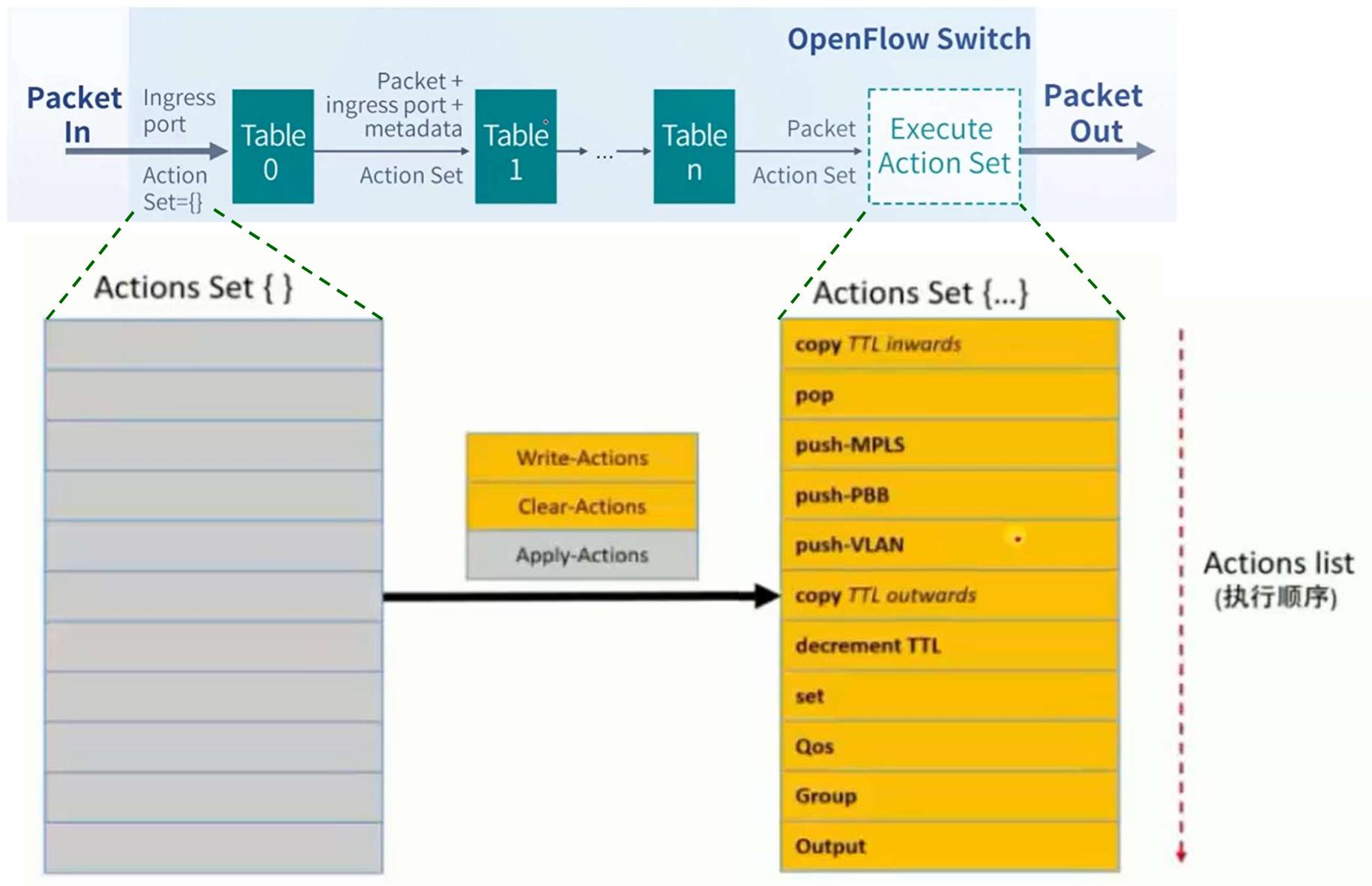


- 指令是对动作集的操作
- 动作是对数据包的操作

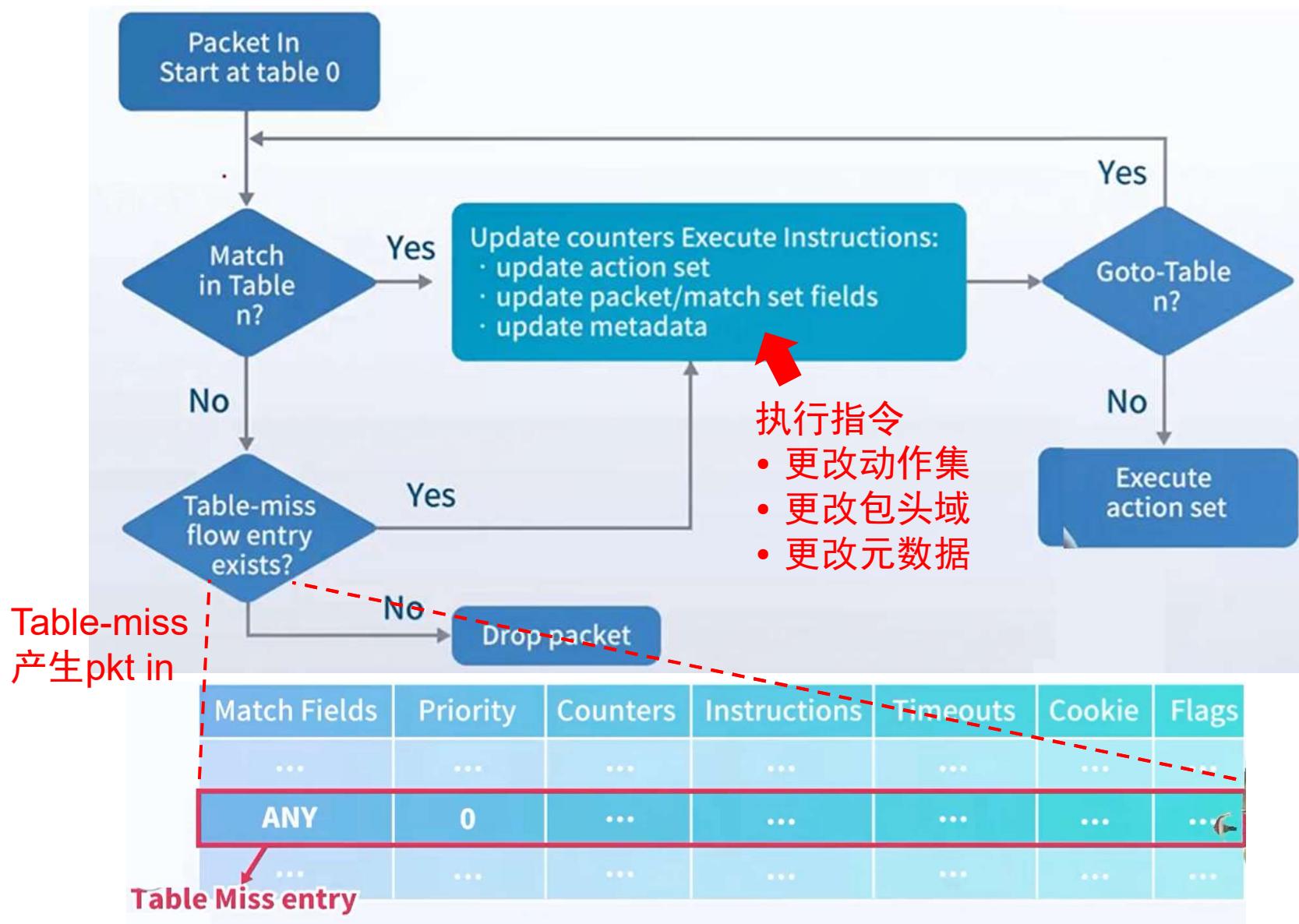
Instructions vs. Actions

- Each flow entry contains a set of **instructions**
 - Direct to another flow table
 - Update action set
- An **action set** is associated with each packet
 - It's executed at the end of the pipeline
 - Actions: send to controller, forward, flood, decrement TTL, etc.

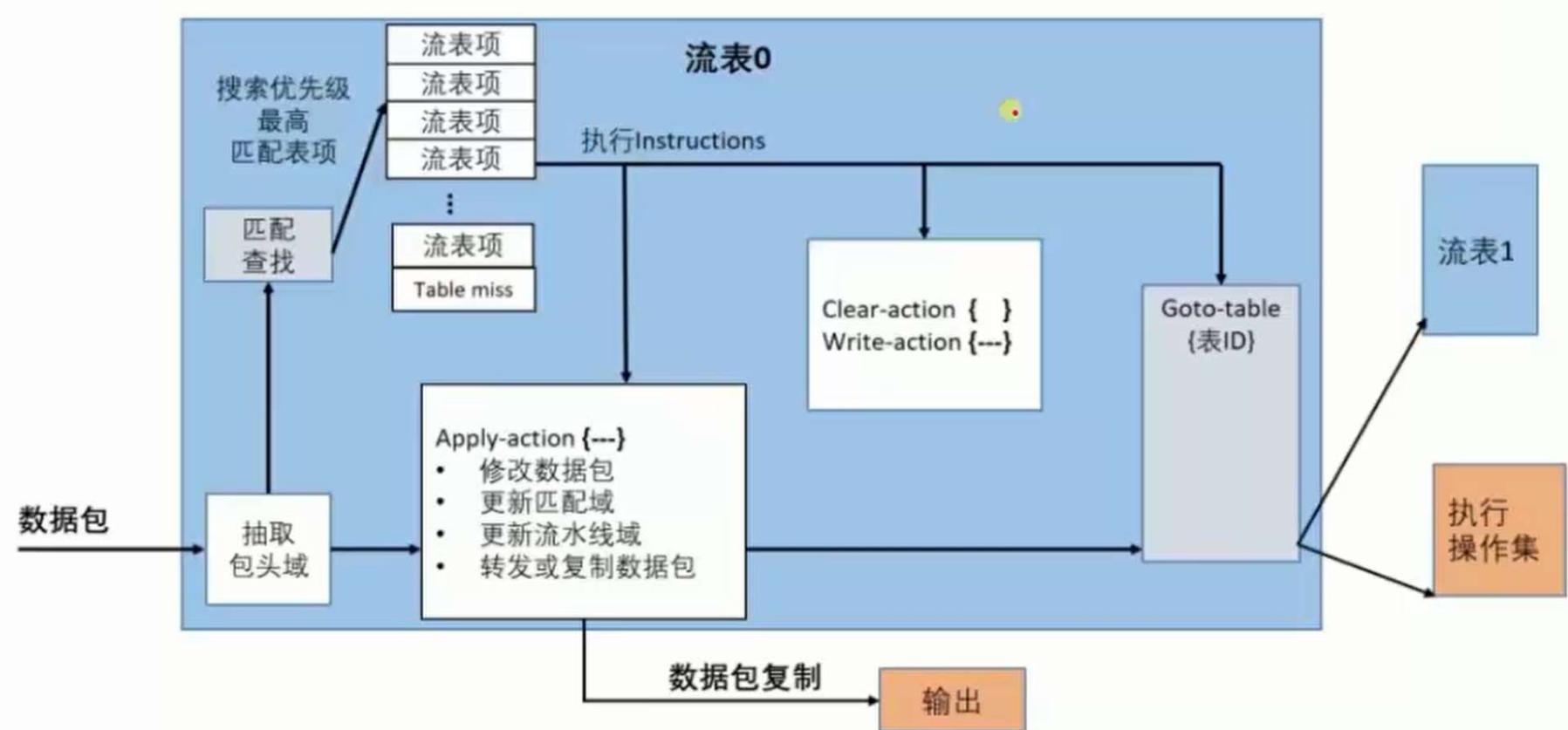
Action set



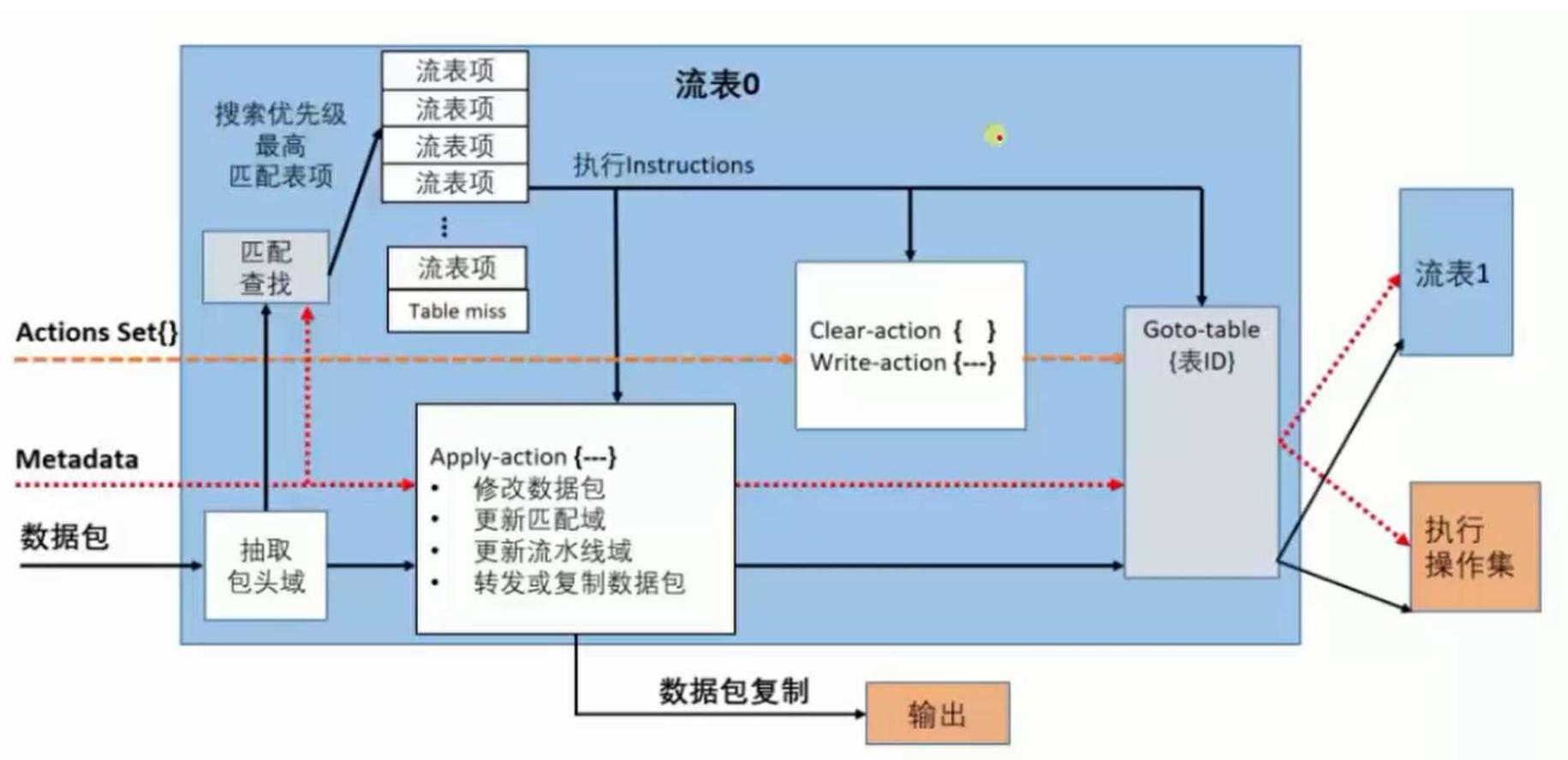
Pkt processing flow



Pkt processing flow



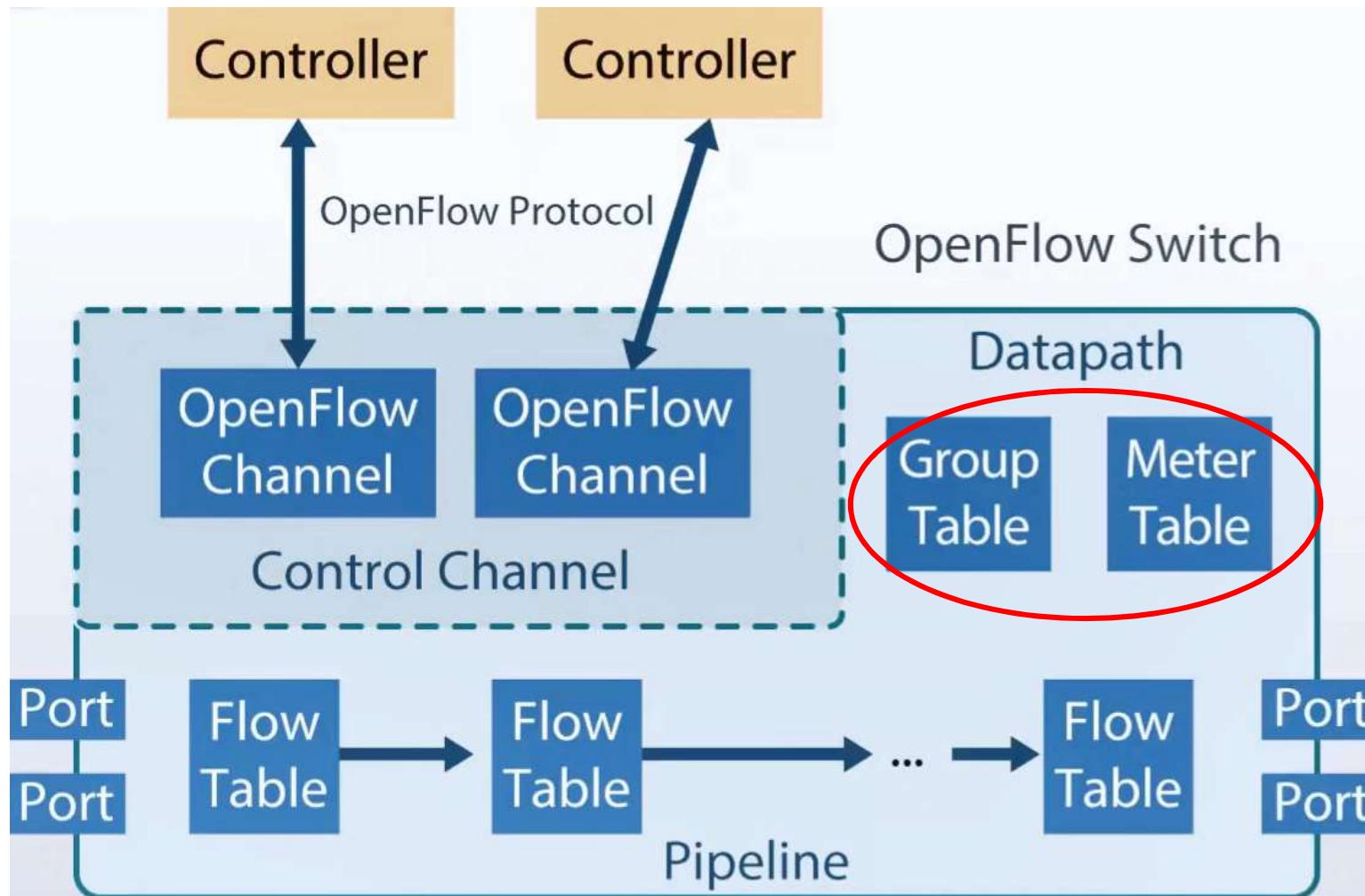
Pkt processing flow



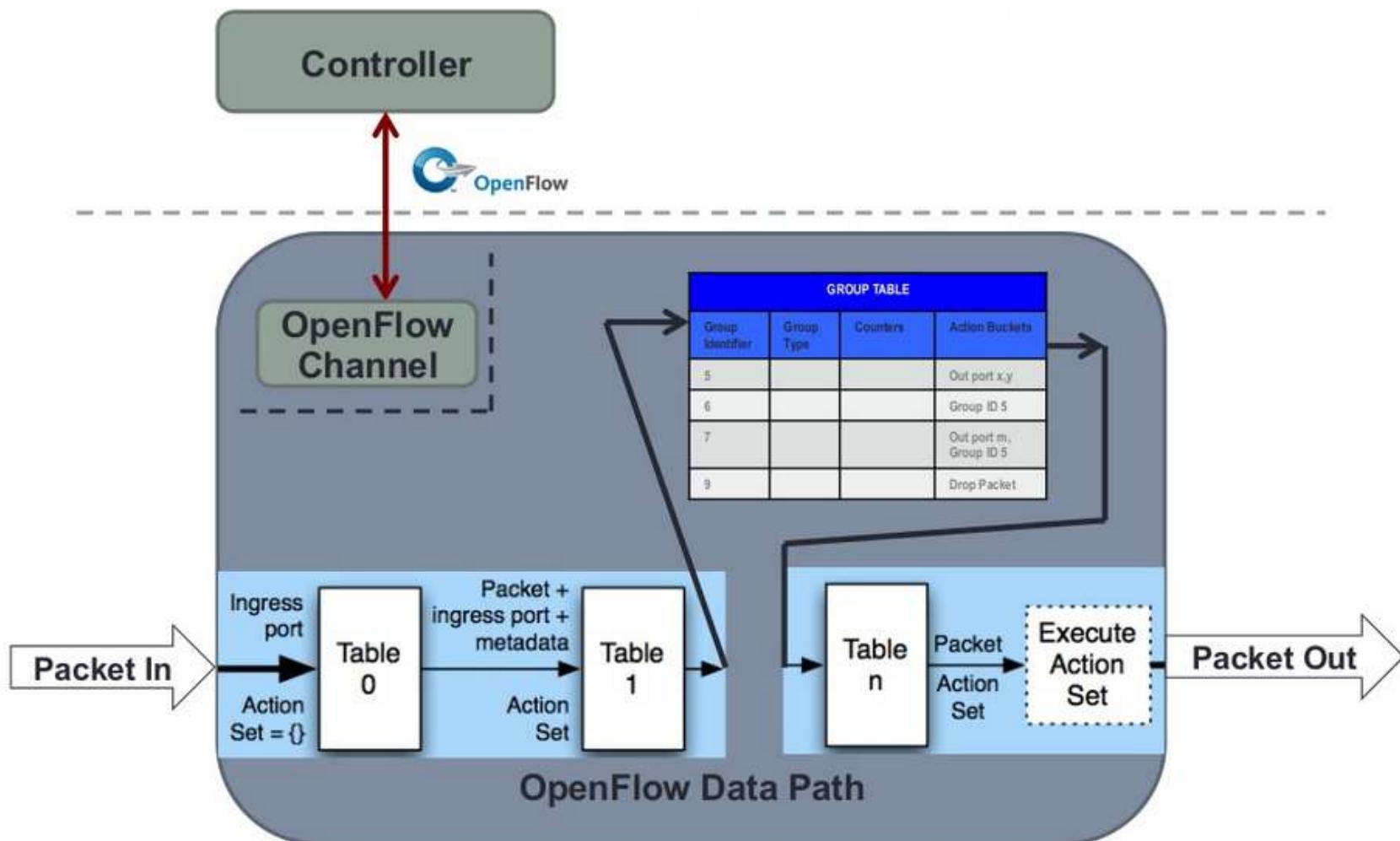
OpenFlow

- OpenFlow overview
- Flow abstraction
- OF-switch abstraction
- Flow table pipeline
- **Group table & meter table**
- OpenFlow protocol
- Modes of operations

Group table & meter table

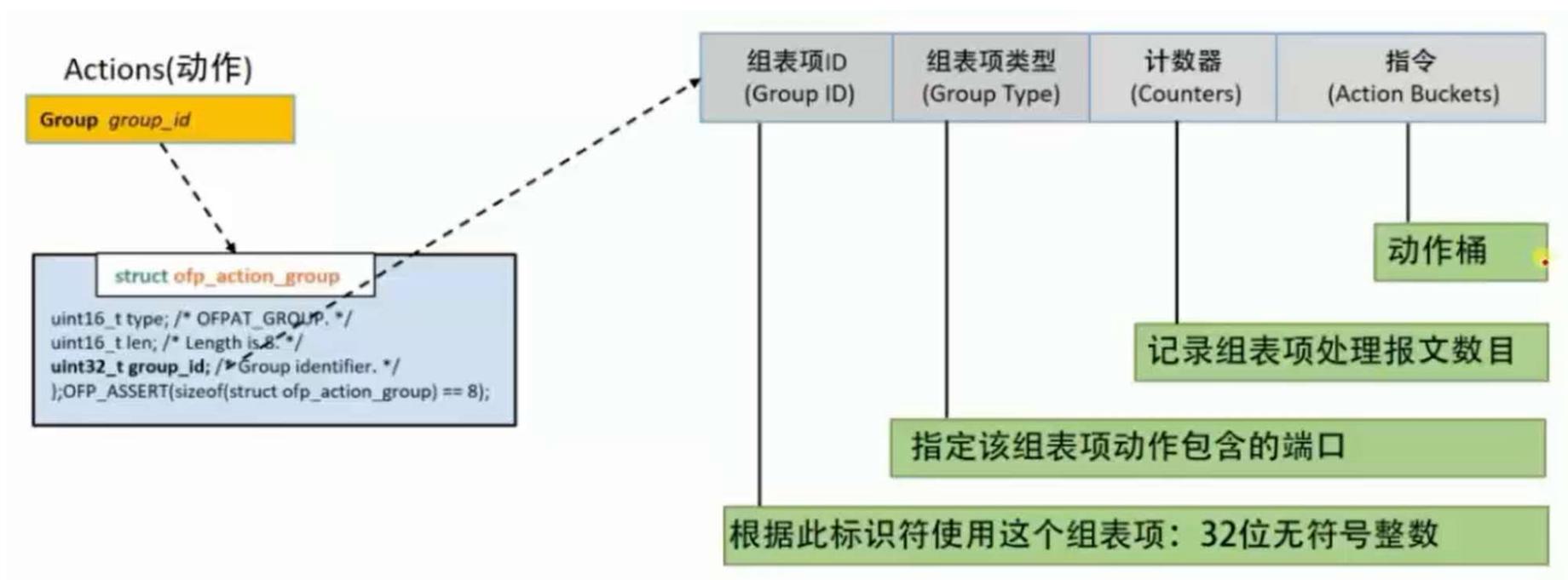


Group table



Group table

Group Identifier	Group Type	Counters	Action Buckets
...	{...} {...} {...}
...	{...}
...	{...} {...}

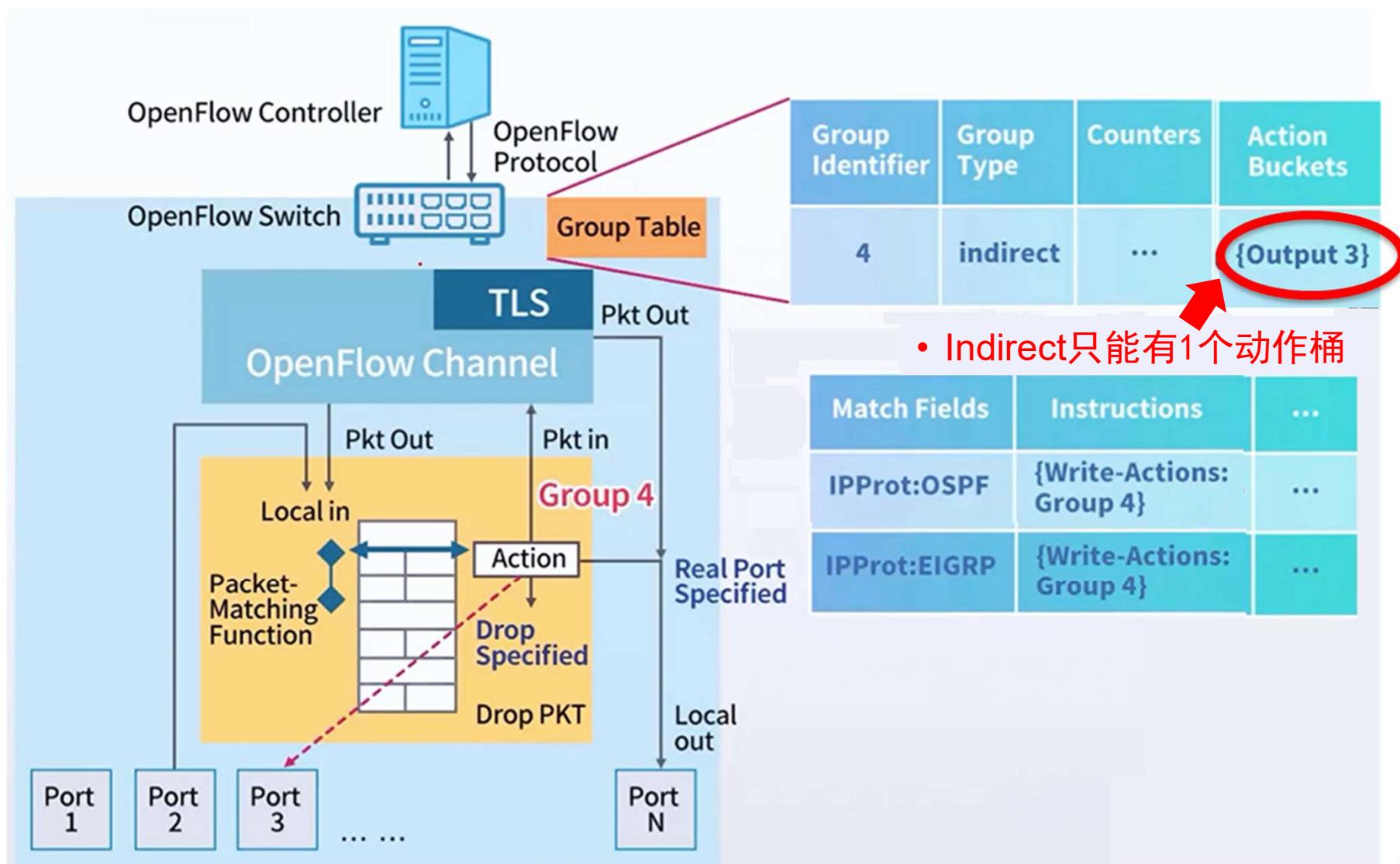


Group table

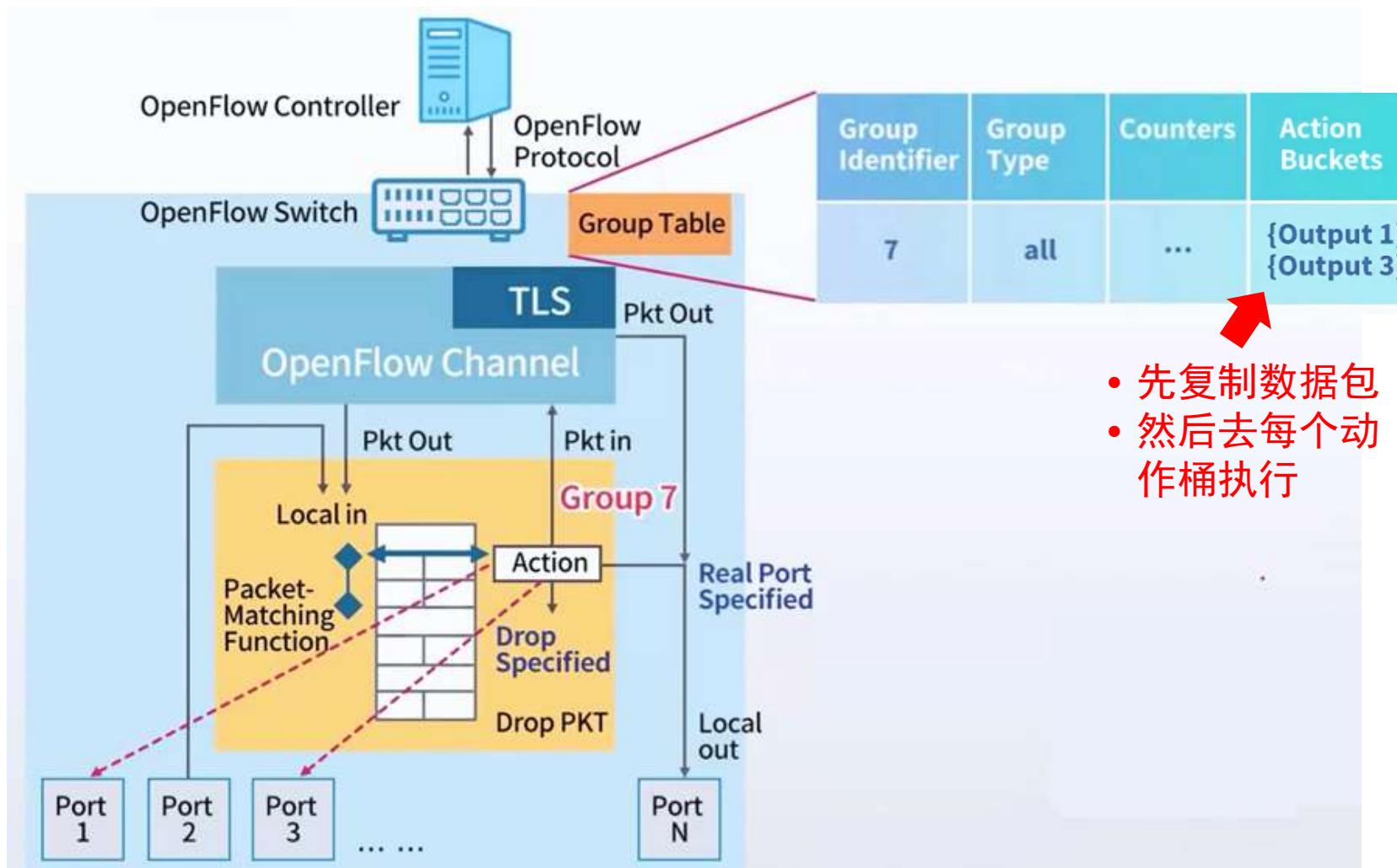
- A flow entry can point to a group
- Groups represent multiple ports

组表项ID (Group ID)	组表项类型 (Group Type)	计数器 (Counters)	指令 (Action Buckets)
必须支持	描述		
Indirect	这种组表项只能支持一个Action Bucket。用来实现多个流表项或多个组表项指向单一组表ID的情况，支持快速、高效的汇聚功能。		
All	这种组表项会执行Action Buckets中所有Action Bucket中的操作动作集，可用来实现网络中的组播和广播功能。		
可选支持	描述		
Select	这种表项只选择执行一个Action Bucket的动作集，通常用交换机自己的算法来选择这个Action Bucket，比如根据用户配置元组的哈希算法，或者简单的轮询算法。		
Fast failover	这种表项只执行第一个激活的Action Bucket。这种组表类型能让转发模型自己调整转发操作，不需要每次都请求控制器。Fast Failover类型组表多用于容灾备份场景		

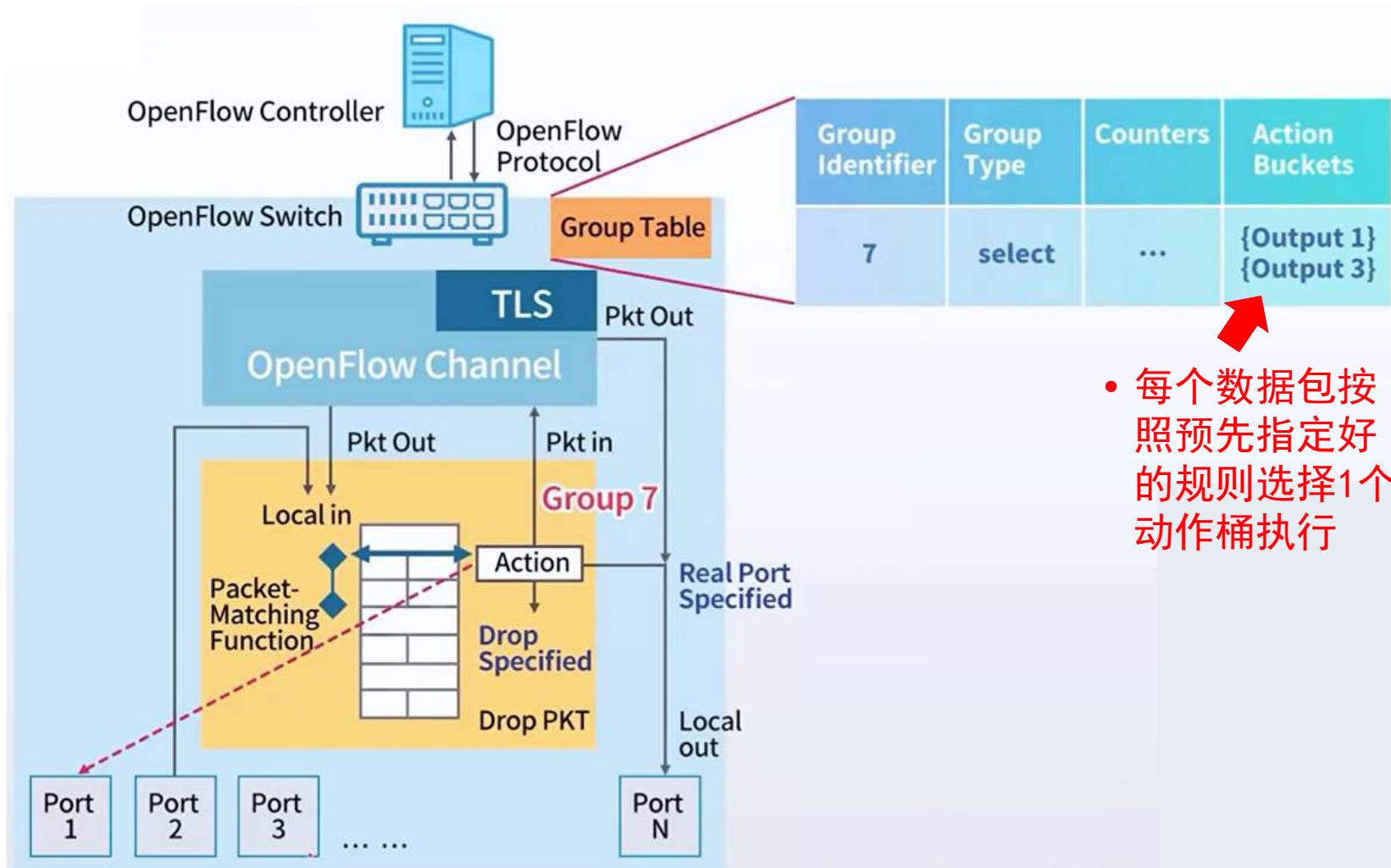
Indirect using group table



Multicast using group table



Load balance using group table

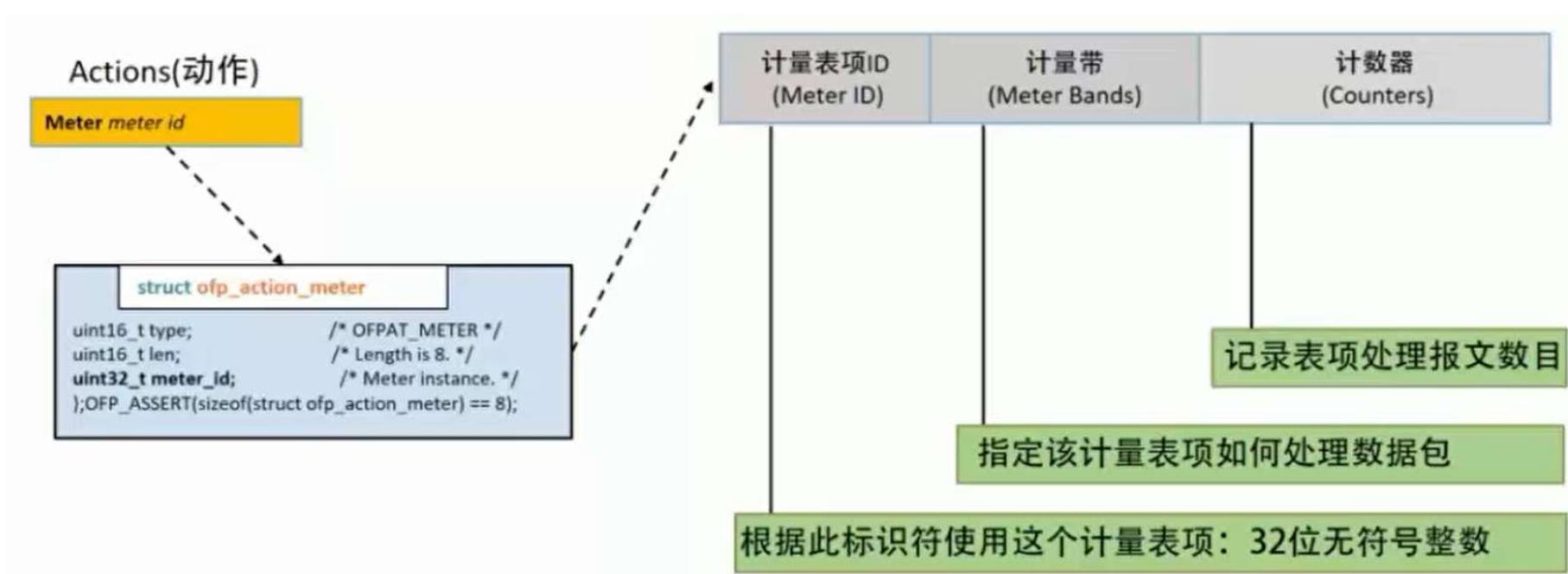


Meter table

- A meter table consists of **per-flow meters** used for rate limiting
 - A flow entry points to a meter
 - A meter measures the rate of packets
 - Packets can be dropped or marked (DiffServ)
- Can be combined with per-port queues to support some QoS tasks
- Queue scheduling and configuration is outside the scope of OpenFlow

Meter table

Meter Identifier	Meter Bands	Counters
...	{...} {...}	...
...	{...}	...



Meter table

匹配域 (Meter ID)	计量带 (Meter Bands)	计数器 (Counters)		
				量杯上的水位，超过了指定的水位，就执行给定的动作
Band类型 (Band Type)	计量速率 (Rate)	Burst	计数器 (Counters)	类型参数 (Type specific arguments)
可选支持	描述			
drop	当前流量速率超过这个速率阈值时，丢弃数据包			
dscp remark	需要依据报文的DSCP优先级提供差分服务时，可以改写remark dscp命令配置重标记报文的DSCP优先级			

Meter table

Flow Table

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags
...	...	21Mbps	{Meter 8}

Meter Table

Meter Identifier	Meter Bands	Counters
8	{drop, 20, ...} {dscp_remark, 10, ...}	...



OpenFlow Switch Counters

匹配域 (Match Fields)	优先级 (Priority)	计数器 (Counters)	指令 (Instructions)	失效时间 (Timeouts)	Cookie	Flags
匹配域 (Group ID)	组选项类型 (Group Type)	计数器 (Counters)	指令 (Action Buckets)			
匹配域 (Meter ID)	计量带 (Meter Bands)	计数器 (Counters)				

OpenFlow Switch Counters

- 针对交换机中的每张流表、每个数据流、每个设备端口、每个转发队列进行维护，用于统计数据流量的相关信息：
 - 对每张流表，统计当前活动的表项数、数据包查询次数、数据包匹配次数等；
 - 对每个数据流，统计接收到的数据包数、字节数、数据流持续时间等；
 - 对每个设备端口，除统计接收到的数据包数、发送数据包数、接收字节数、发送字节数等指标外，还可以对各种错误发生的次数进行统计；
 - 对每个队列，统计发送的数据包数和字节数，还有发送时的溢出错误次数等。

OpenFlow Switch Counters

Per Flow Table	
Reference Count (active entries)	32
Packet Lookups	64
Packet Matches	64

Per Queue	
Transmit Packets	64
Transmit Bytes	64
Transmit Overrun Errors	32
Duration (s, ns)	32

Per Meter	
Flow Count	32
Input Packet Count	64
Input Byte Count	64
Duration (s, ns)	32

Per Flow Entry	
Received Packets	64
Received Bytes	64
Duration (s, ns)	32

Per Group	
Reference Count (flow entries)	32
Packet Count	64
Byte Count	64
Duration (s, ns)	32

Per Group Bucket	
Packet Count	64
Byte Count	64

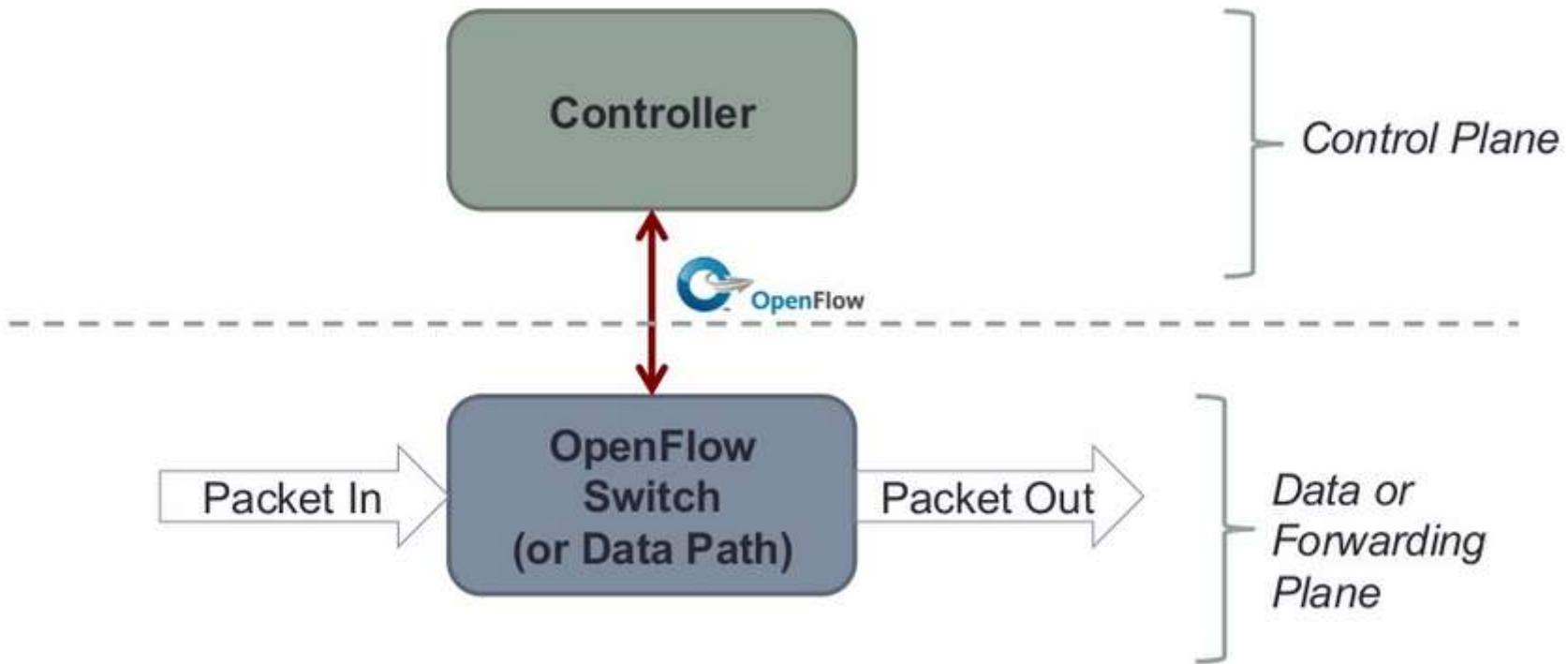
Per Meter Band	
In Band Packet Count	64
In Band Byte Count	64

Per Port	
Received Packets	64
Transmitted Packets	64
Received Bytes	64
Transmitted Bytes	64
Receive Drops	64
Transmit Drops	64
Receive Errors	64
Transmit Errors	64
Receive Frame Alignment Errors	64
Receive Overrun Errors	64
Receive CRC Errors	64
Collisions	64
Duration (s, ns)	32

OpenFlow

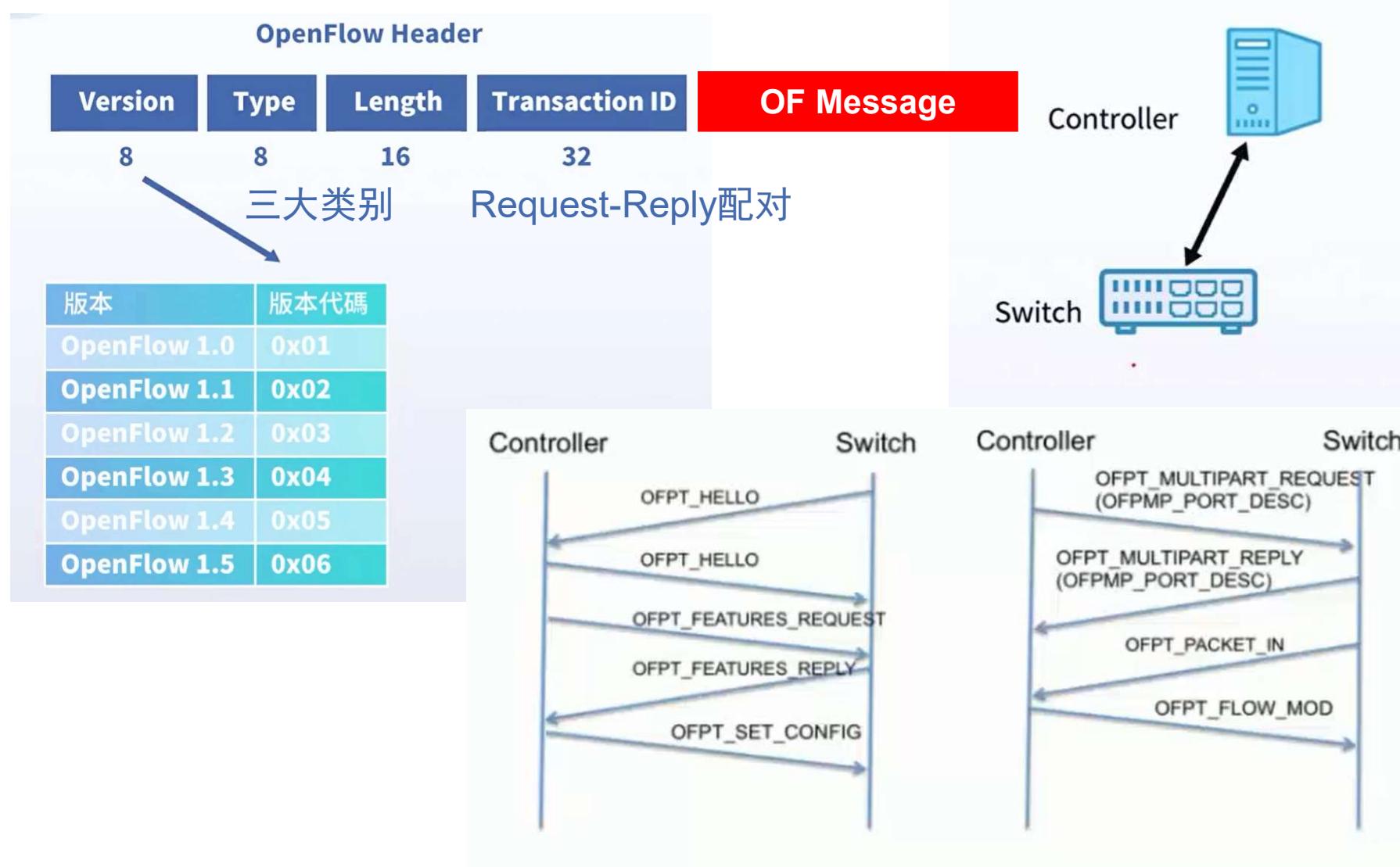
- OpenFlow overview
- Flow abstraction
- OF-switch abstraction
- Flow table pipeline
- Group table & meter table
- **OpenFlow protocol**
- Modes of operations

Controller-switch channel



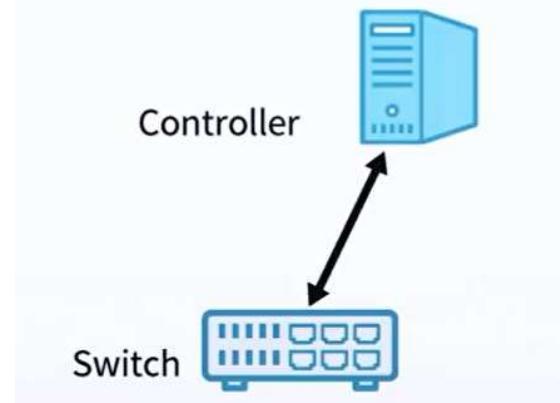
- Manual controller discovery
- TCP (with TLS) connections
- Send link/port state to controller
- Encapsulate packets for controller

OpenFlow message format



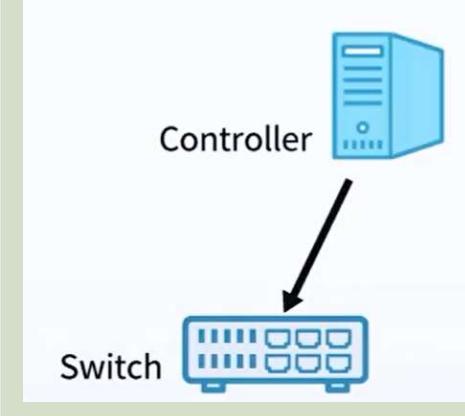
OpenFlow message types

- Controller-to-switch messages
 - Manage flow entries
 - Request info on switch capabilities and counters
 - Send a packet back to a switch
- Asynchronous messages
 - Send to controller a packet that does not match
 - Inform the controller that a timer has expired or that an error has occurred
- Symmetric messages
 - Hello and echo messages



Controller-to-switch messages

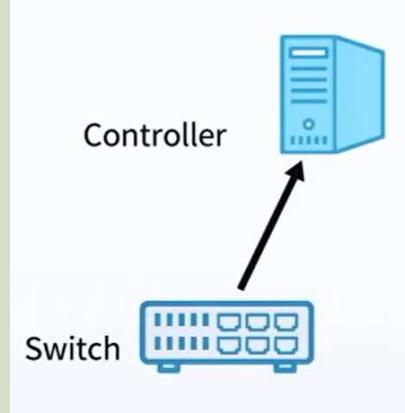
名称	说明	备注
Features	发送Feature_Request消息请求交换机的特性信息	<ul style="list-style-type: none">➤ 由控制器发起，可能会要求交换机回复对应的消息。
Configuration	设置或查询交换机的配置信息	
Modify-state	管理交换机流表项和端口状态等	
Read-state	收集交换机的各种信息，包括配置信息、统计信息和功能特性	
Send-packet	通过Packet-out消息向交换机的指定端口发送数据包	
Barrier	确保动作执行顺序，保持策略一致性	



Controller-to-switch messages

子类型	消息名称	功能描述
Switch features	OFPT_FEATURES_REQUEST OFPT_FEATURES_REPLY	控制器获取交换机的功能信息：Datapath ID (64bits), n_buffers等
Switch Configuration	OFPT_GET_CONFIG_REQUEST OFPT_GET_CONFIG_REPLY OFPT_SET_CONFIG	控制器会设置和查询交换机配置参数
Modify-State	OFPT_TABLE/FLOW/GROUP/PORT/METER_MOD	控制器配置或更新Flow Table的动态信息： 添加、修改或删除flow/group/meter表项 新增或删除组表的action buckets 设置switch端口参数
Multipart	OFPM_P_FLOW/AGGREGATE/GROUP/GR OUP_DESC/GROUP_FEATURES	查询特定Flow的statistics/description/features信息
Packet-out	OFPT_PACKET_OUT	控制器指定从交换机某个端口发送数据包 转发通过Packet-in消息接收的数据包
Barrier	OFPT_BUNDLE_CONTROL	控制器用来确保request/reply 消息之间的独立性
Role-Request	OFPT_ROLE_REQUEST OFPT_GET_ASYNC_REQUEST OFPT_GET_ASYNC_REPLY OFPT_SET_ASYNC	设置控制器与交换机连接通道的一些信息 设置一个异步消息过滤器

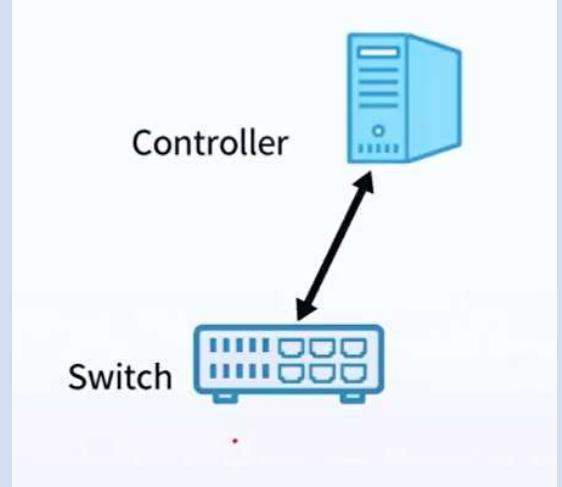
Asynchronous messages

名称	说明	备注
Packet-in	<p>流表中没有匹配项，则发送Packet-in 消息给控制器。</p> <p>有足够的缓存，只需要将分组头中的控制信息和交换机缓存标识（buffer id）； 不支持缓存（或缓存不足）：发送整个数据包</p>	<p>➤ 交换机主动发起的，用于告知控制器新数据包的到达和交换机状态的改变</p> 
Flow-removed	流表项因为超时或修改等原因被删除时，会触发该消息。	
Port-status	端口状态发生变化时（Up/Down），触发该消息。	
Error	通知控制器发生的问题	

Asynchronous messages

子类型	消息名称	功能描述
Packet-in	OFPT_PACKET_IN	交换机将数据包可以发送给控制器
Flow-Removed	OFPT_FLOW_REMOVED	流表项失效时间满足时，OFS会向控制器发送该消息
Port-status	OFPT_PORT_STATUS	OFS会向控制器报告端口状态更新消息，添加，删除或更改端口
Controller Role status	OFPT_ROLE_STATUS	OFS会向控制器报告一个控制器角色更新消息
Table Status	OFPT_TABLE_STATUS	OFS会向控制器报告一个表状态更新消息
Request Forward	OFPT_REQUESTFORWARD	某个控制器成功修改表状态，OFS需要向其他控制器发送此消息
Controller Status	OFPT_CONTROLLER_STATUS	OFS需要向所有控制器报告某个控制器链接变化信息

Symmetric messages

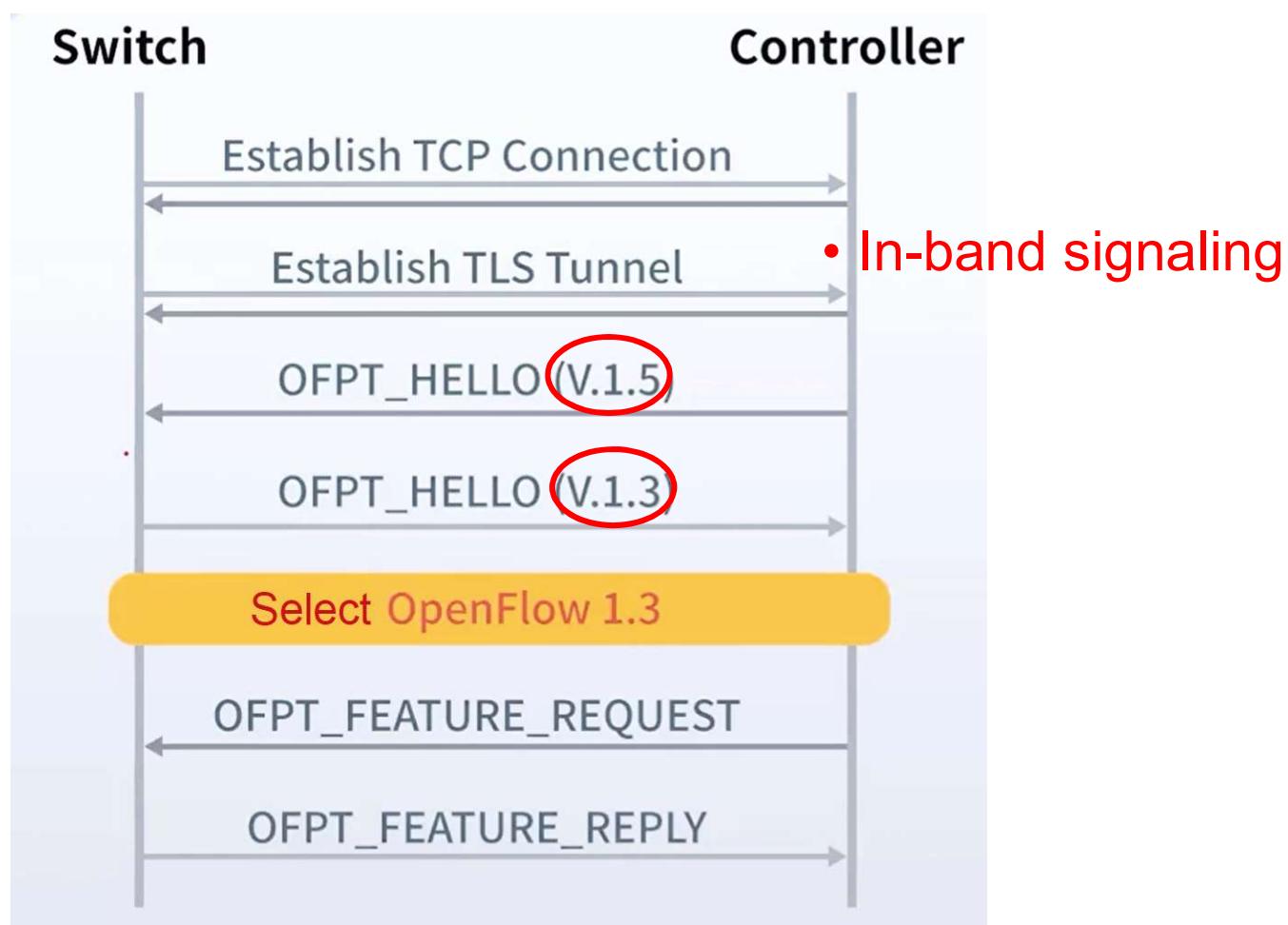
名称	说明	备注
Hello	双方通过握手消息Hello建立安全连接。	➤ 任意一方发送，无需对方的许可或请求
Echo	用于测量延迟、带宽，或用于保持活动连接 Echo-Request Echo-Reply	 A diagram illustrating the communication between a Controller and a Switch. A blue server icon labeled "Controller" is connected to a blue switch icon labeled "Switch" by a black arrow pointing from the Controller to the Switch.
Vendor	为额外的功能预留空间	

Symmetric messages

子类型	消息名称	功能描述
Hello	OFPT_HELLO	Hello消息在用户建立连接时控制器和交换机之间使用
Echo Request	OFPT_ECHO_REQUEST	Echo消息用来确定Controller-to-Switch连接的延时，验证Controller-to-Switch连接是否保持活跃的状态
Echo Reply	OFPT_ECHO_REPLY	Echo消息用来确定Controller-to-Switch连接的延时，验证Controller-to-Switch连接是否保持活跃的状态
Error	OFPT_ERROR_MSG	Error消息用来通知某种非正常情况
Experimenter	OFPT_EXPERIMENTER	Experimenter消息用于厂商或用户的消息扩展。

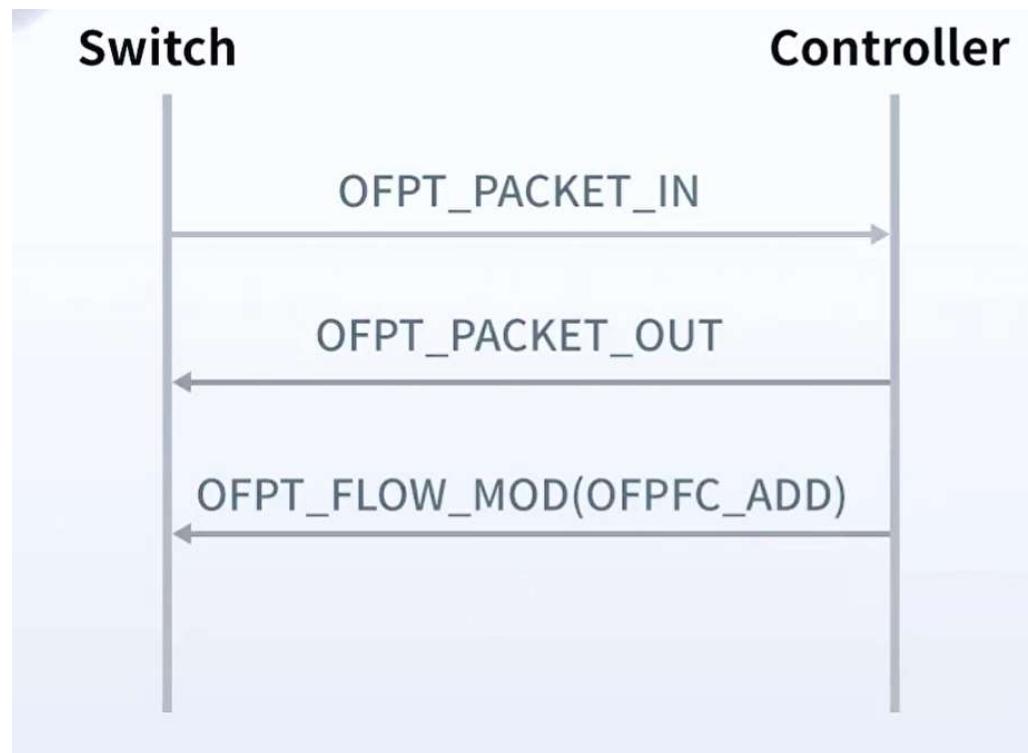
Controller-switch communication

- Establishing a new connection



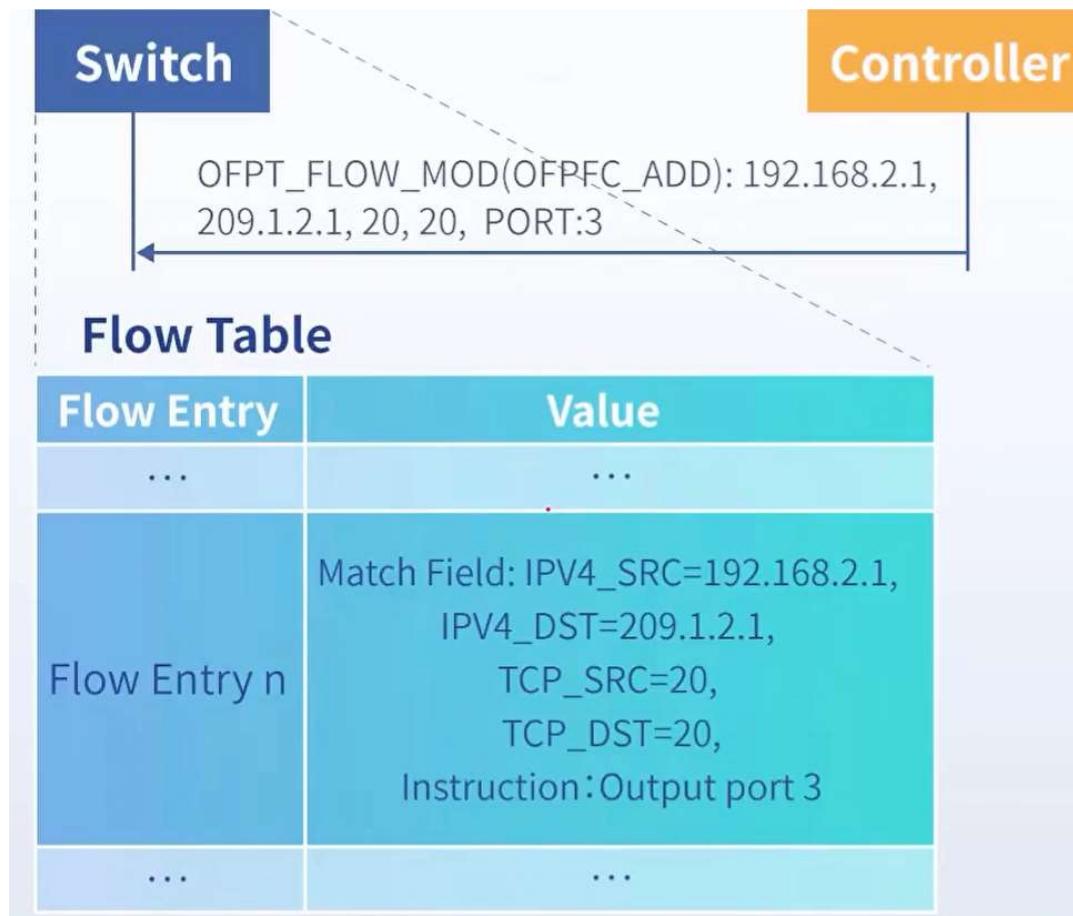
Controller-switch communication

- Asking how to handle a new packet



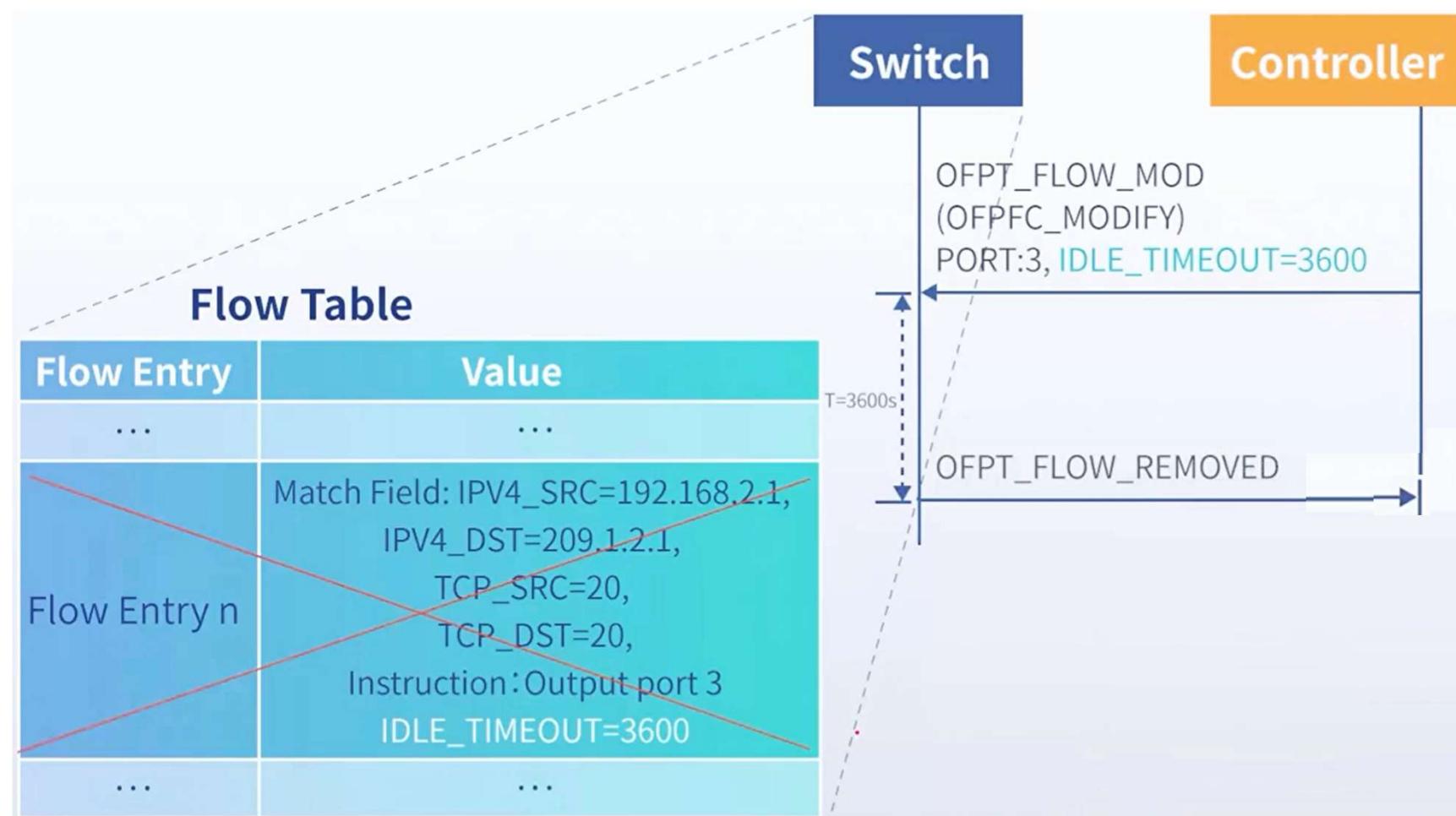
Controller-switch communication

- Adding a new flow entry



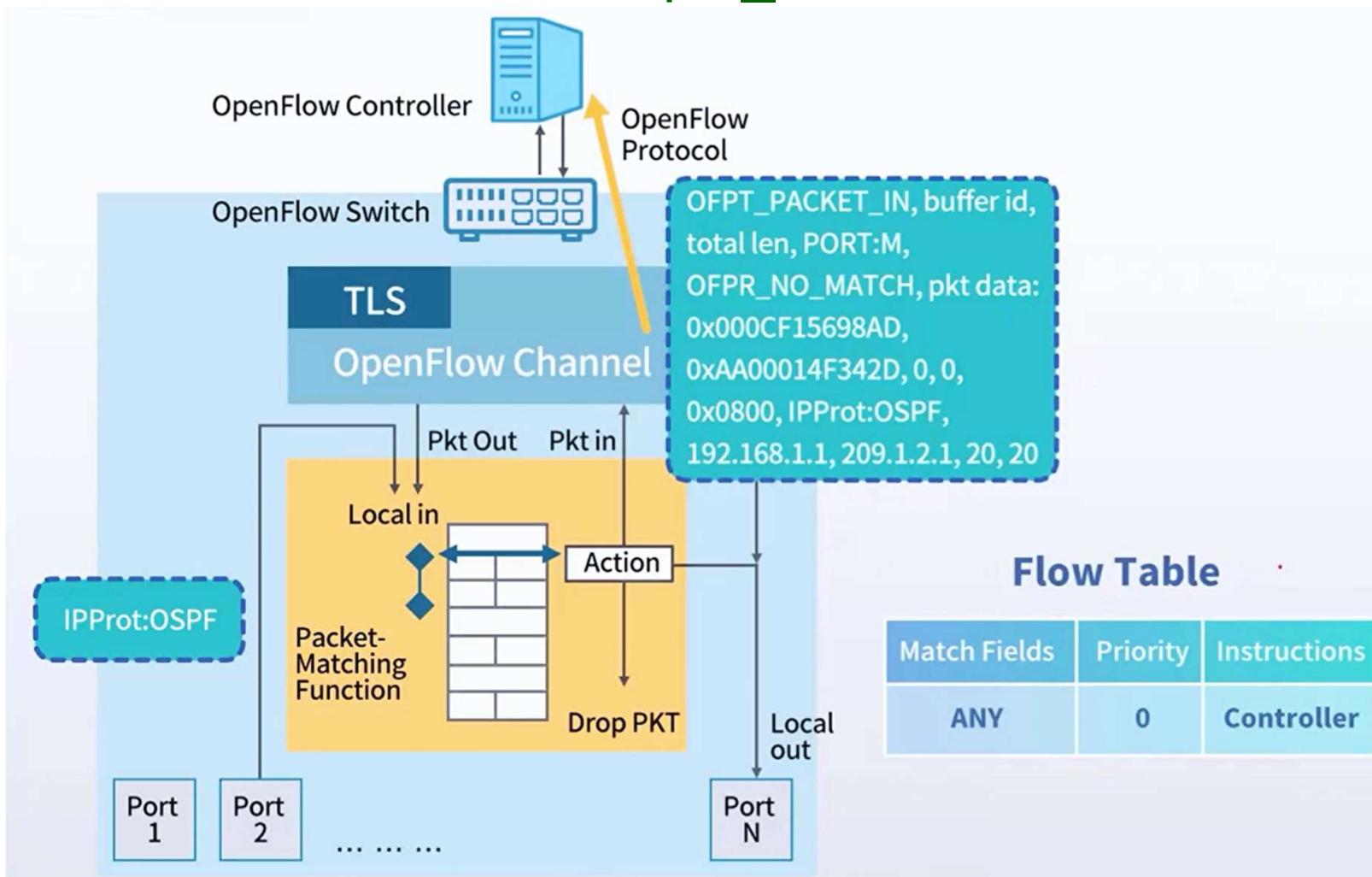
Controller-switch communication

- Adding a timer for a flow entry

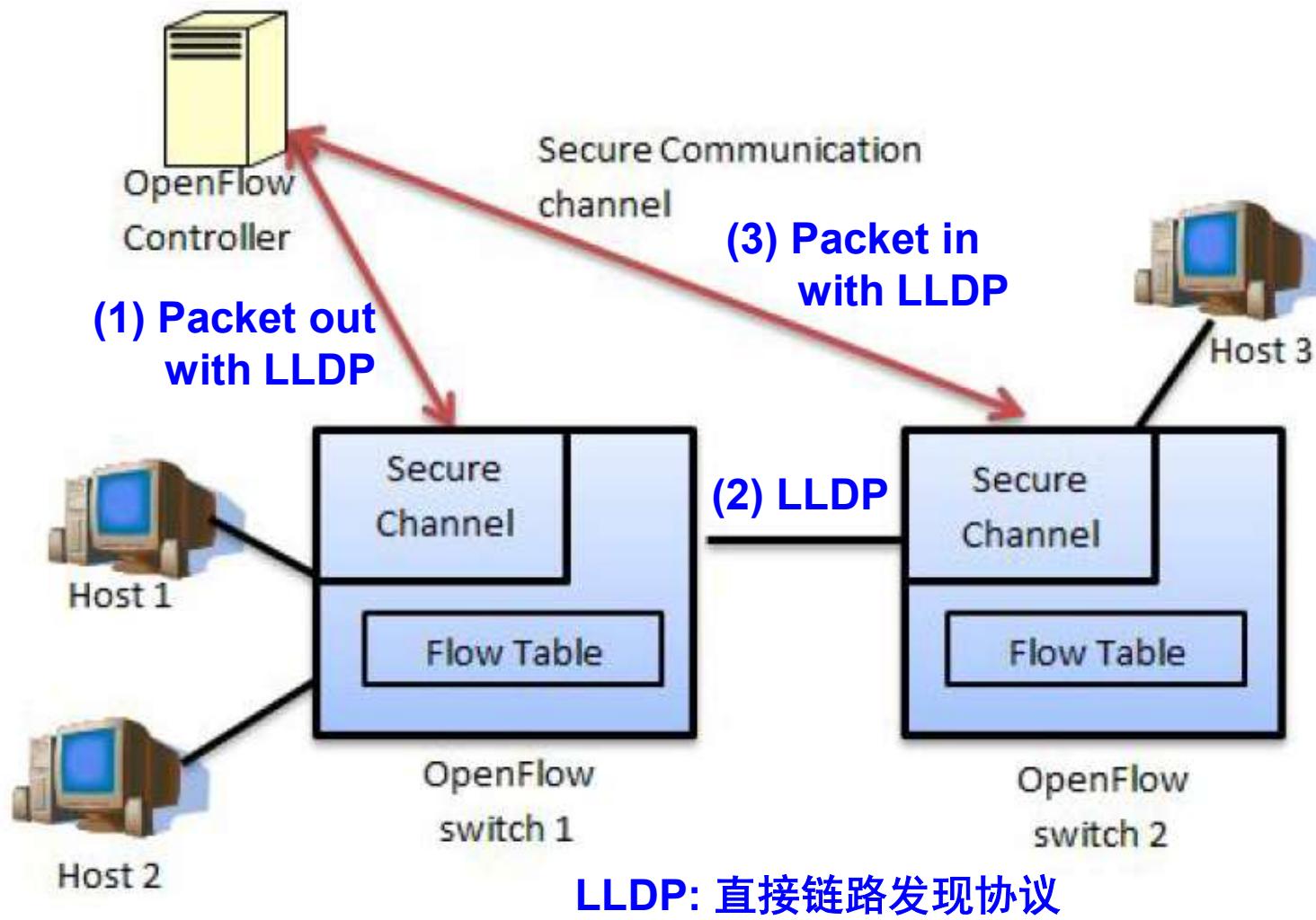


Controller-switch communication

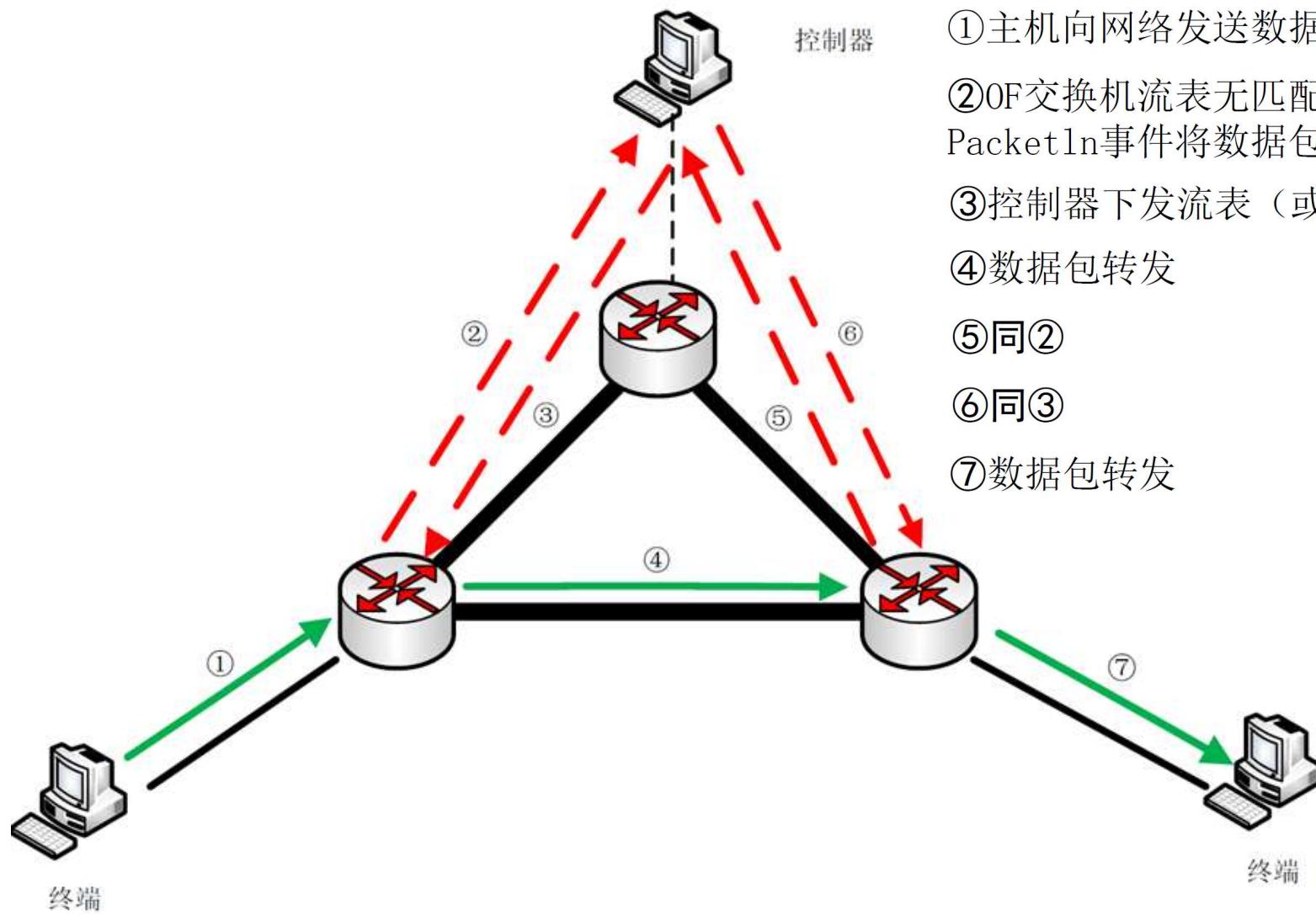
- Table miss -> pkt_in



Topology discovery



Routing with OF



- ① 主机向网络发送数据包
- ② OF交换机流表无匹配项，通过PacketIn事件将数据包上报给控制器
- ③ 控制器下发流表（或PacketOut）
- ④ 数据包转发
- ⑤ 同②
- ⑥ 同③
- ⑦ 数据包转发

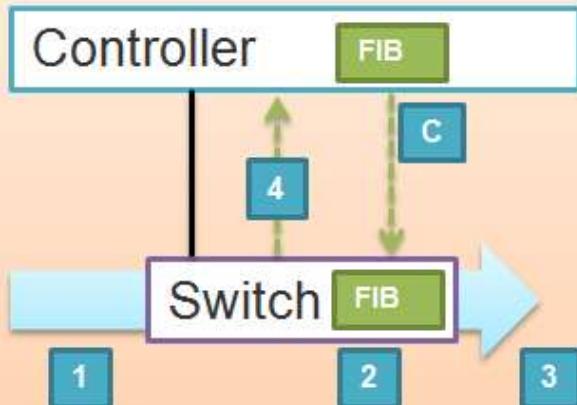
OpenFlow

- OpenFlow overview
- Flow abstraction
- OF-switch abstraction
- Flow table pipeline
- Group table & meter table
- OpenFlow protocol
- **Modes of operations**

Modes of operation

Reactive – Data plane driven

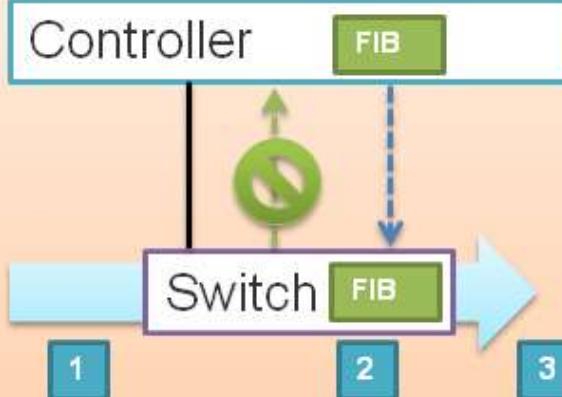
- Base principle of flow caching



1. Receive packet
2. Perform lookup in local FIB
3. Hit: forward to port
4. Miss: forward to controller
 - Controller inspects packet
 - Performs route computation
 - C: Inserts new flow entry

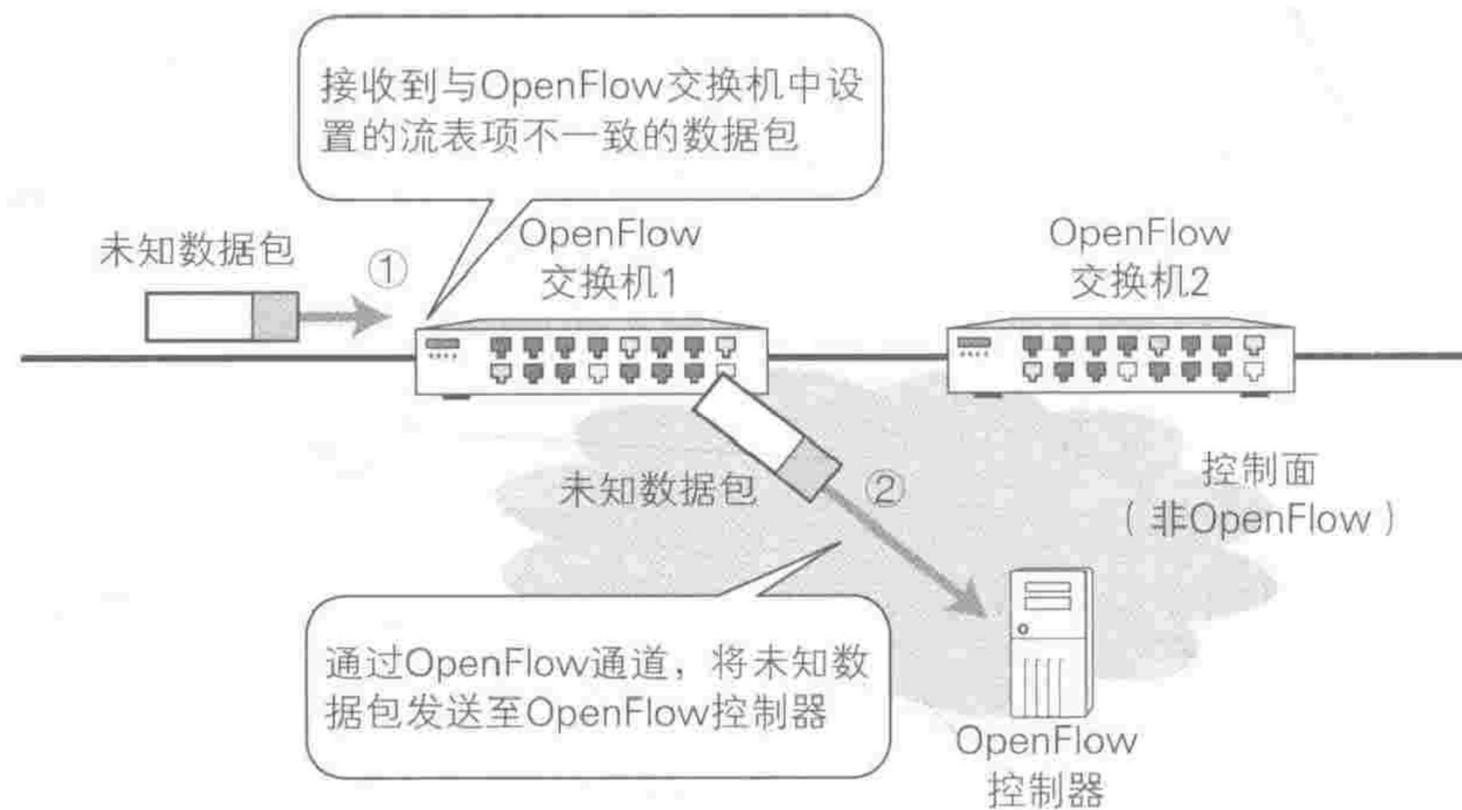
Pro-active – Configuration driven

- Like Static routes & LSPs
- Not stored in configuration file

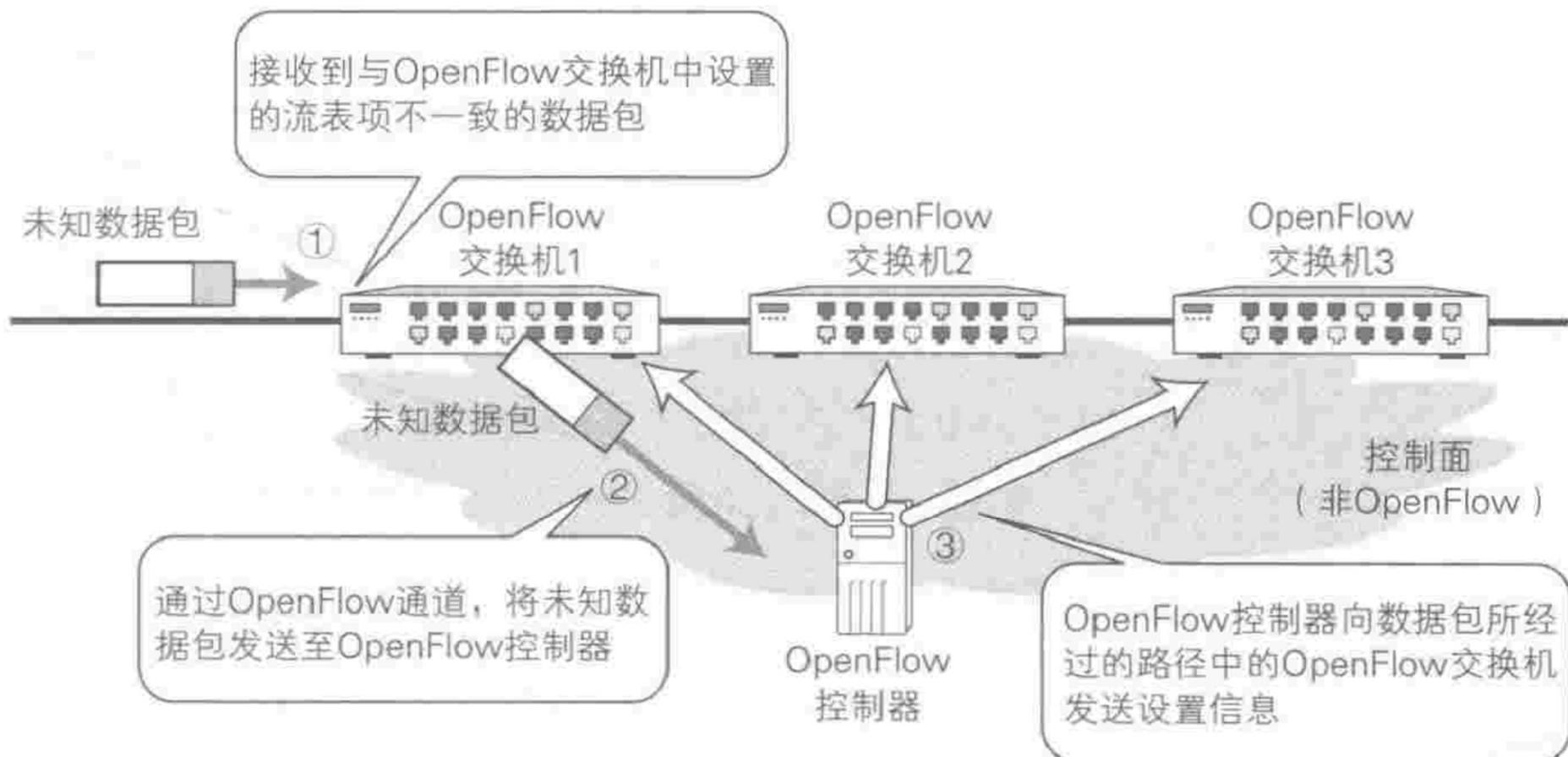


1. Receive packet
2. Perform lookup in local FIB
3. Hit: forward to port
4. Miss: DROP

Reactive mode

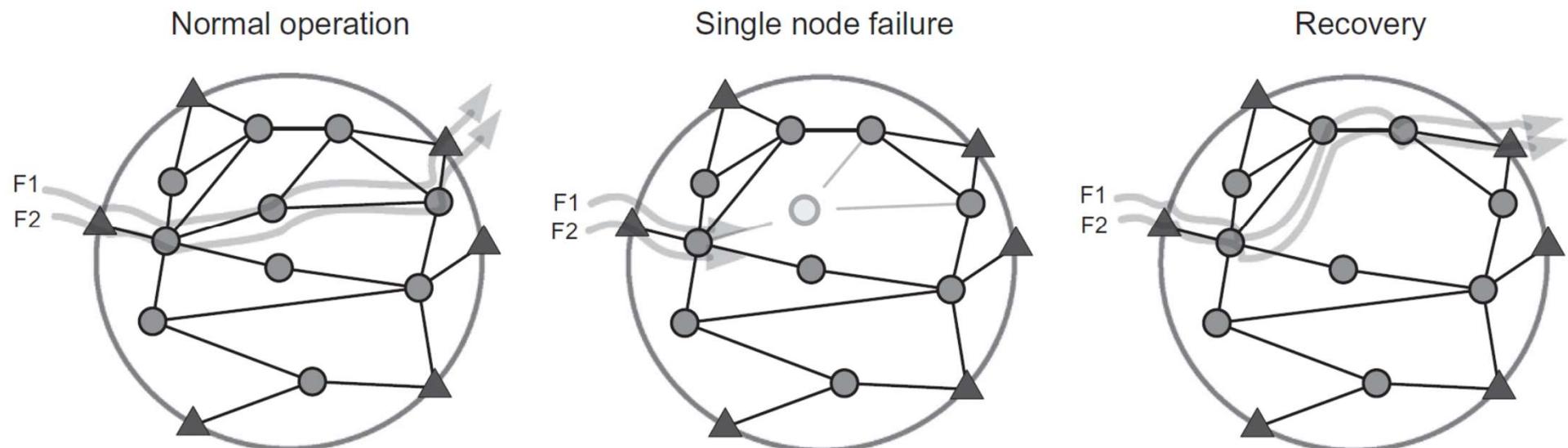


Reactive/proactive mode



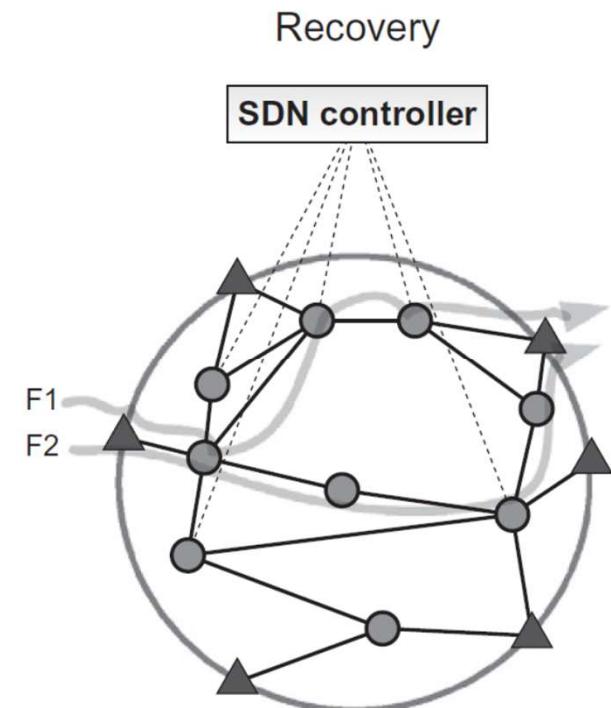
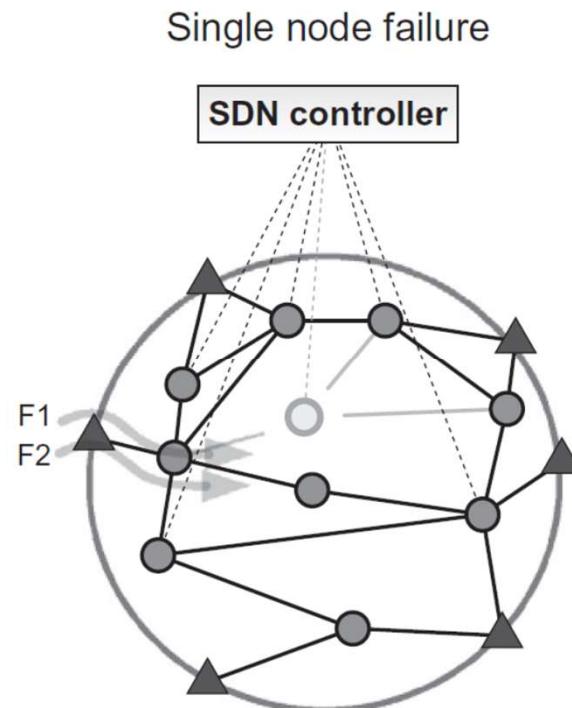
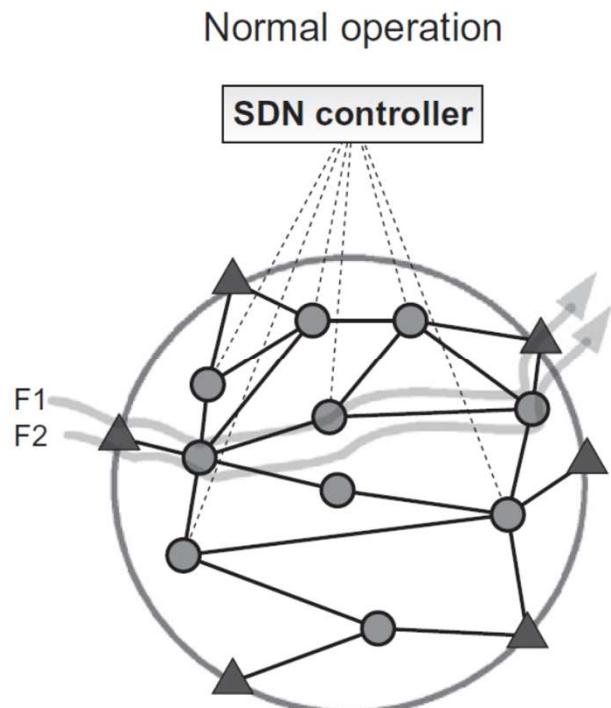
Single point of failure

- Traditional network failure recovery



Single point of failure

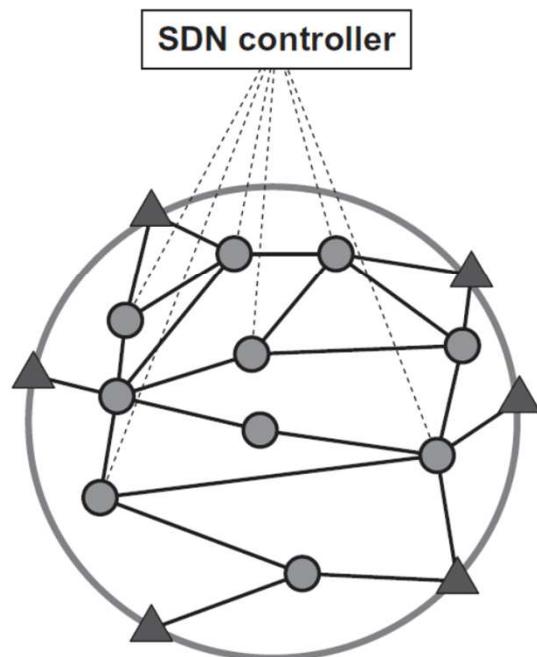
- SDN network failure recovery.



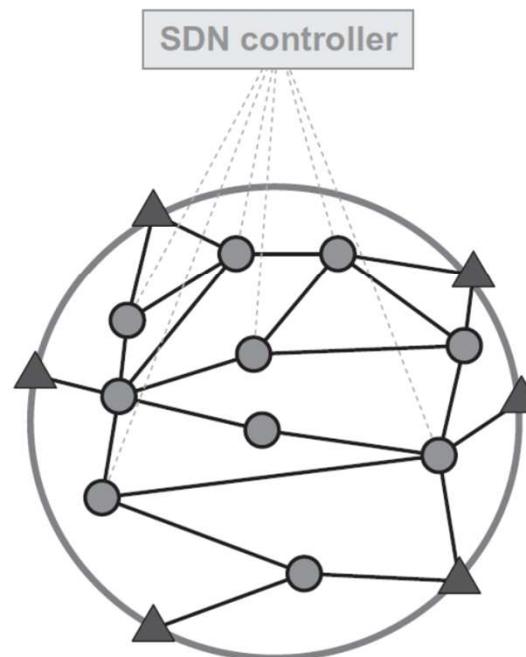
Single point of failure

- SDN controller as single point of failure

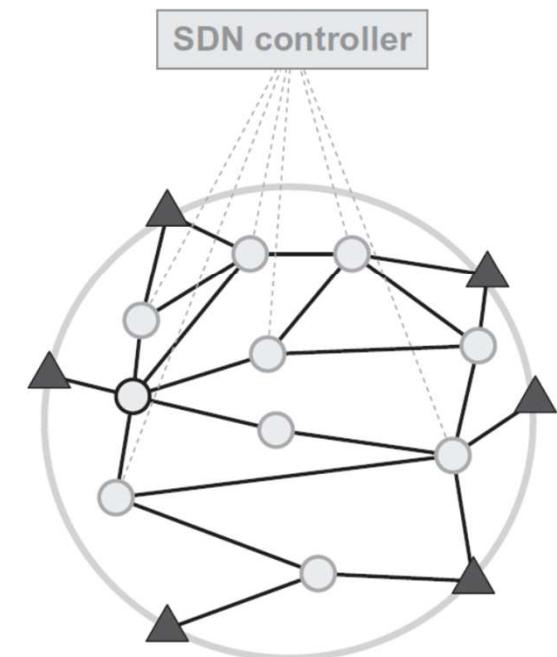
Normal operation



Controller is the
single node failure

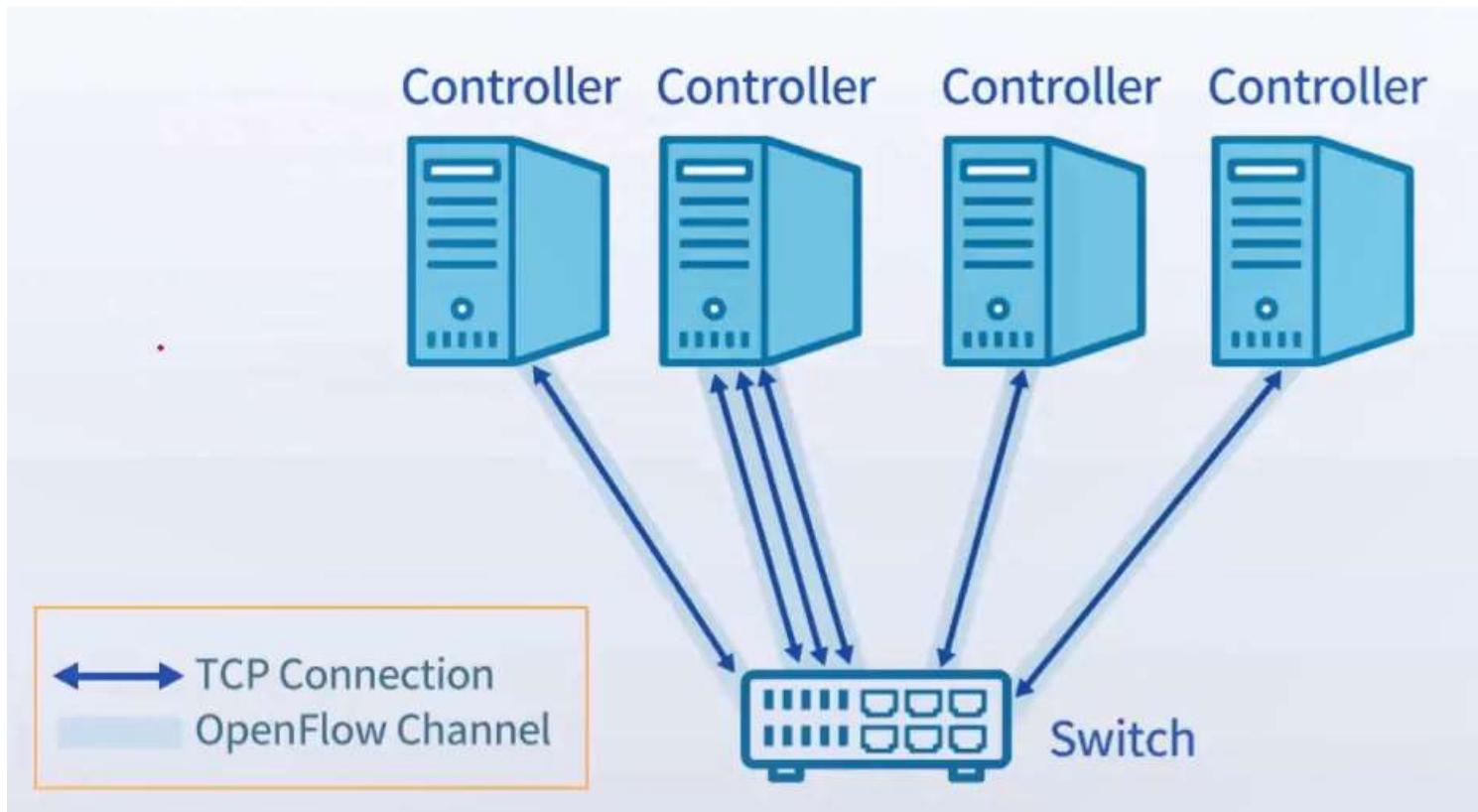


No recovery



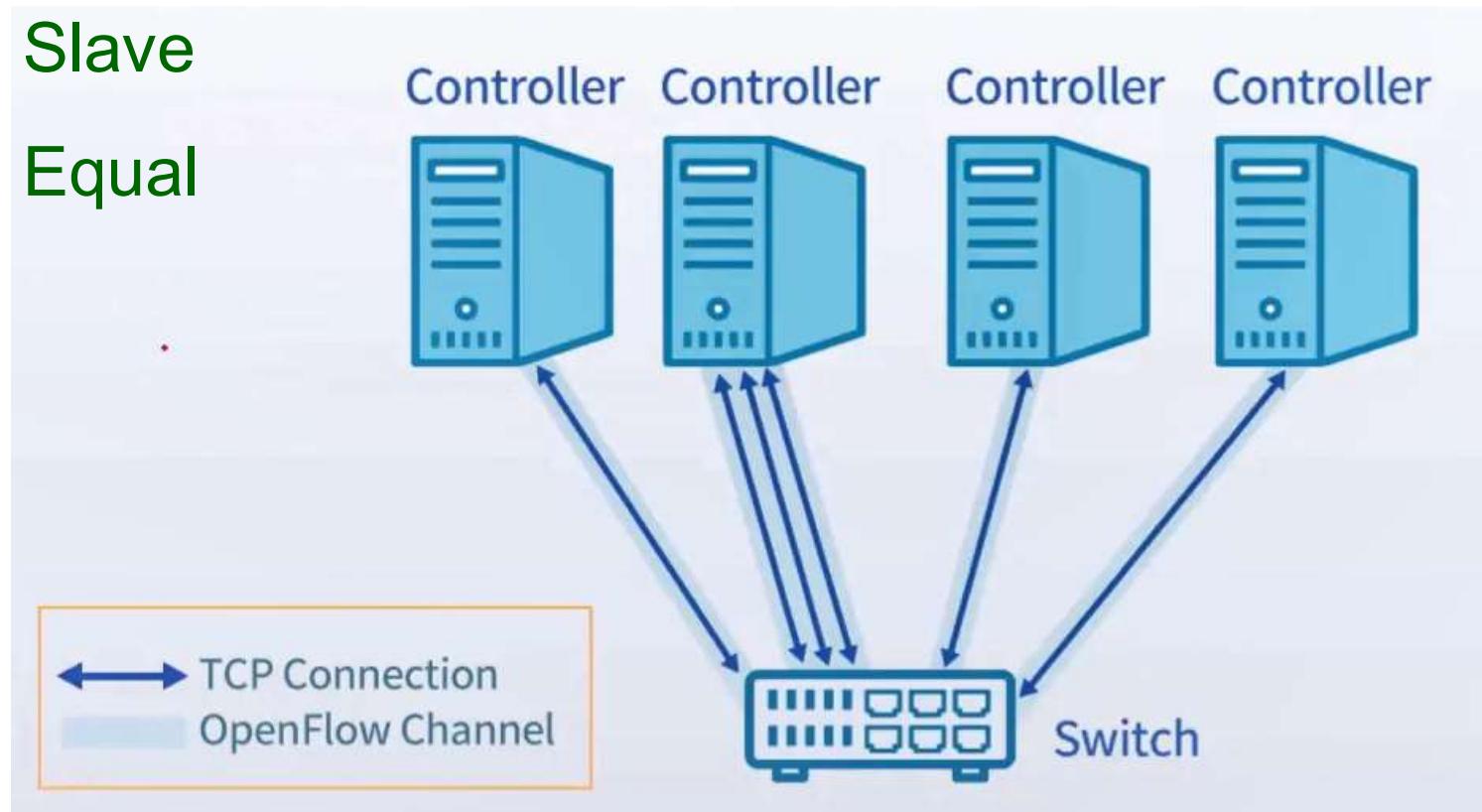
Controller connection failure

- Fail secure -> old flow table
- Fail standalone -> legacy (non-openflow)



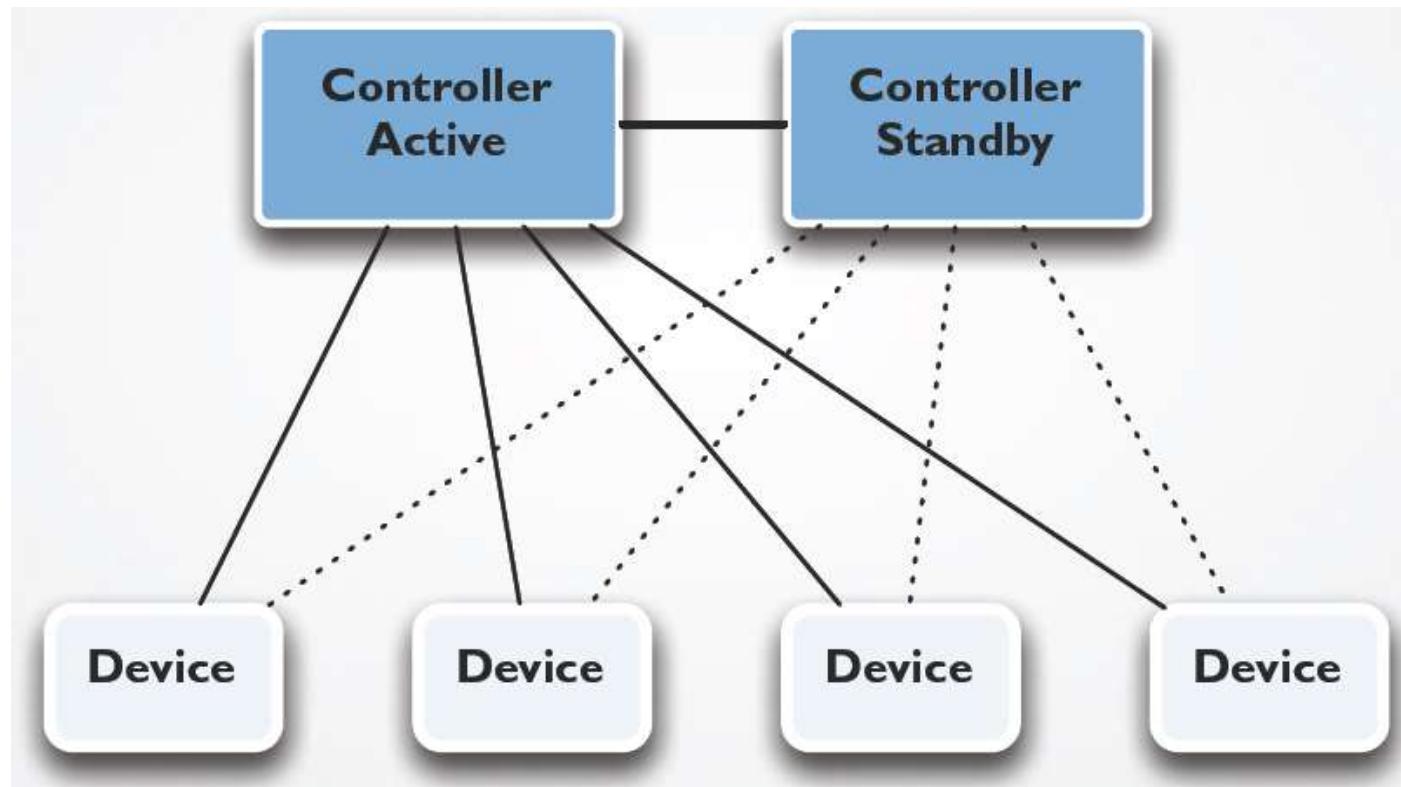
Multiple controller

- Master
- Slave
- Equal



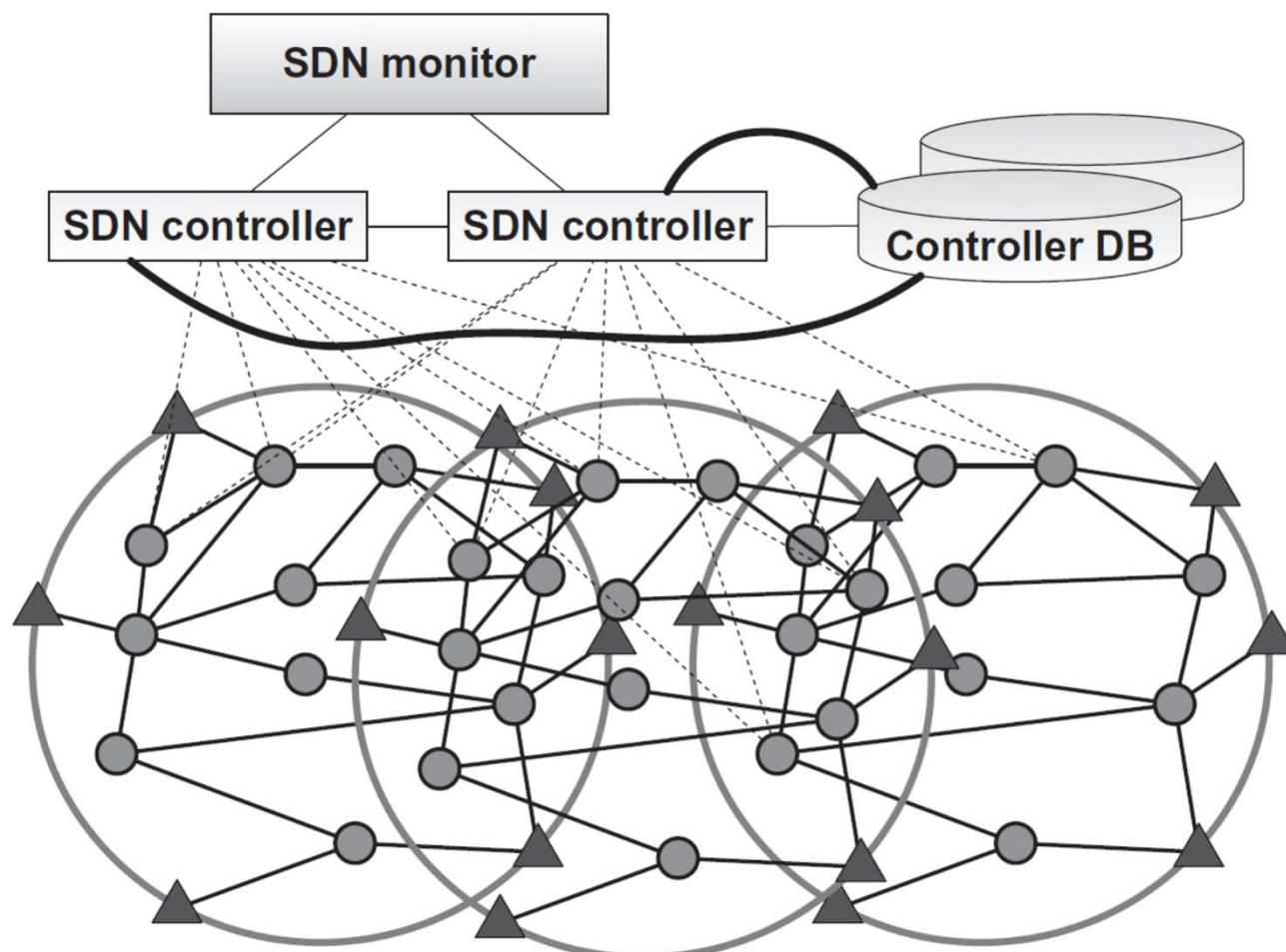
Controller deployment

- Centralized control in practice



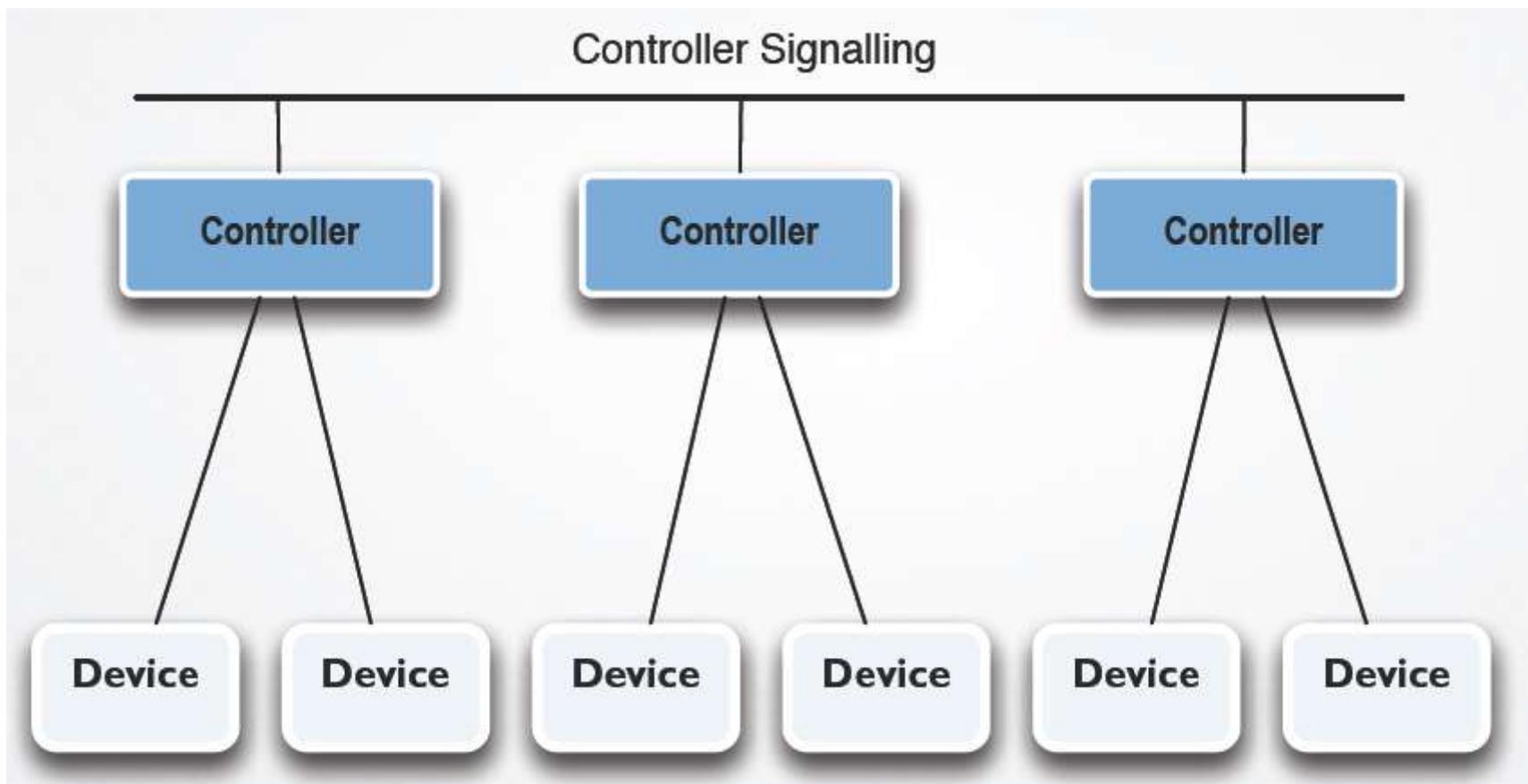
Controller deployment

- Controller high availability and monitoring



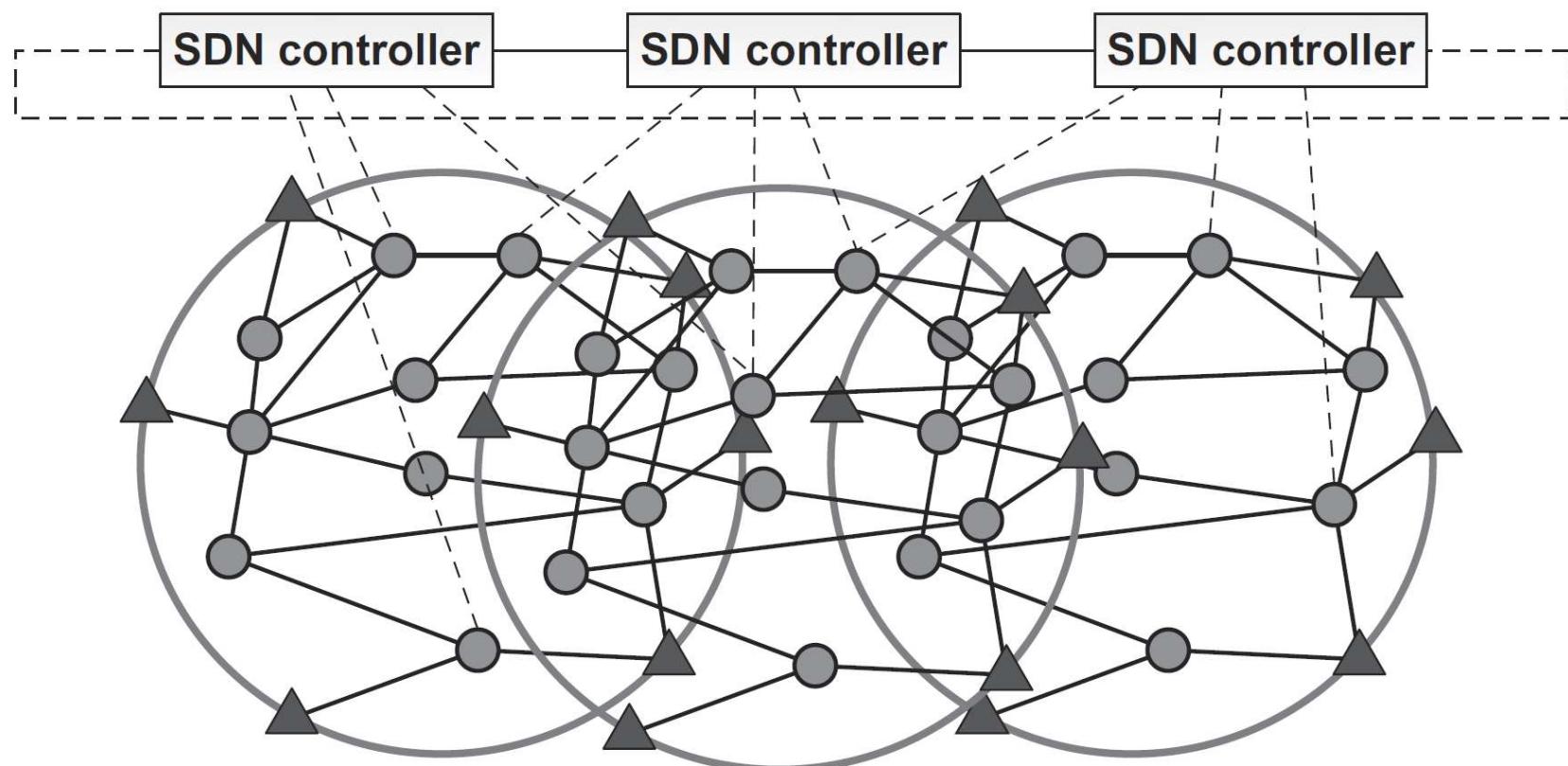
Controller deployment

- Distributed



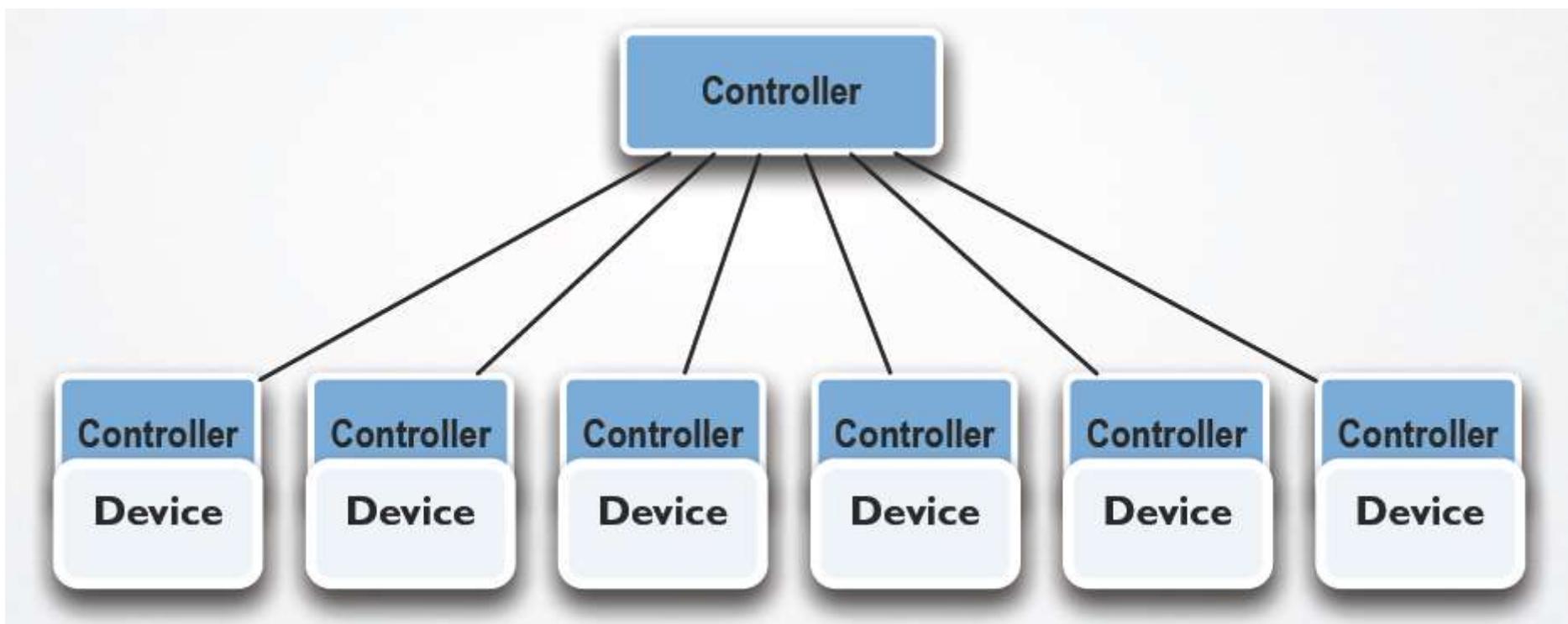
Controller deployment

- Distributed



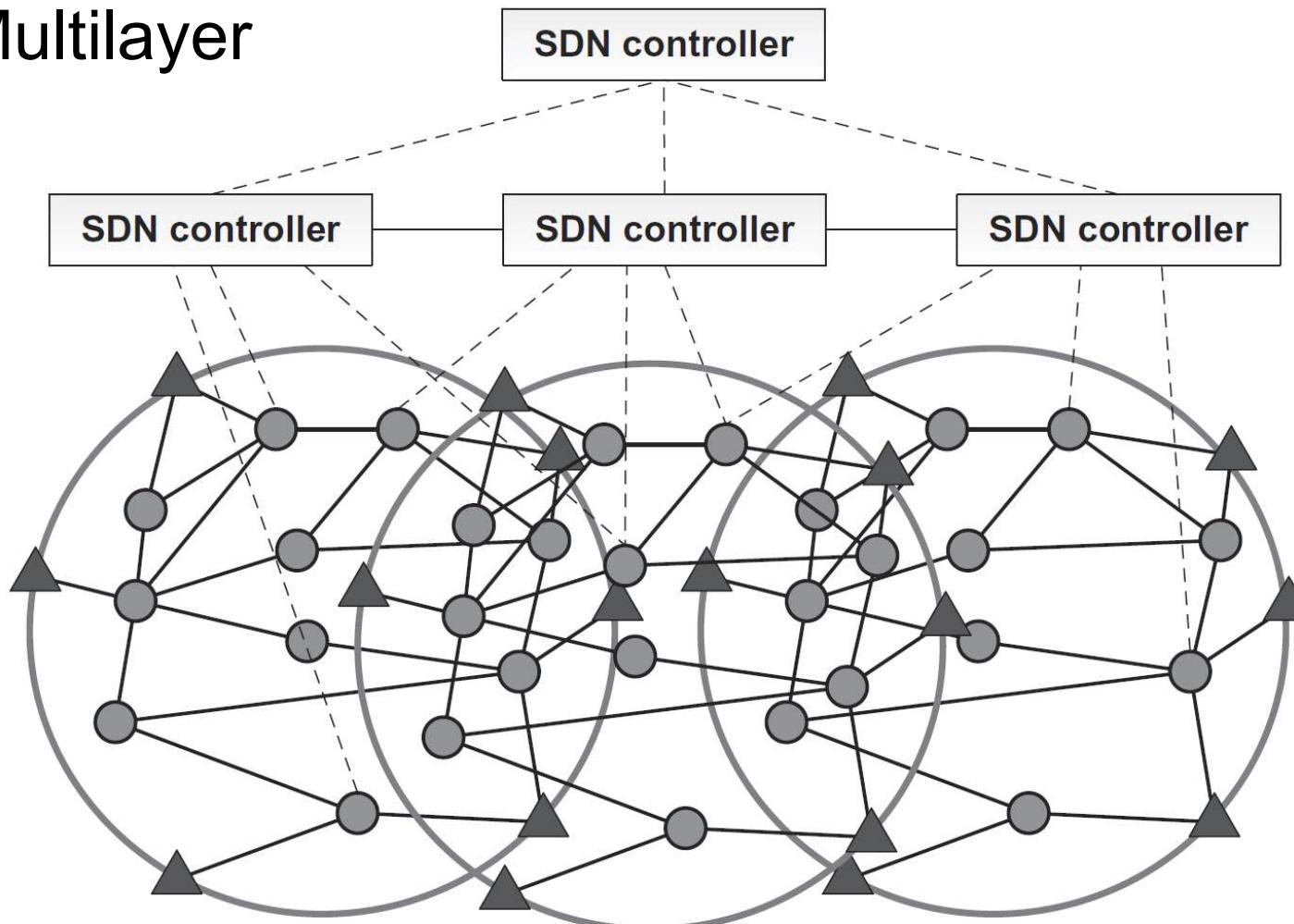
Controller deployment

- Multilayer

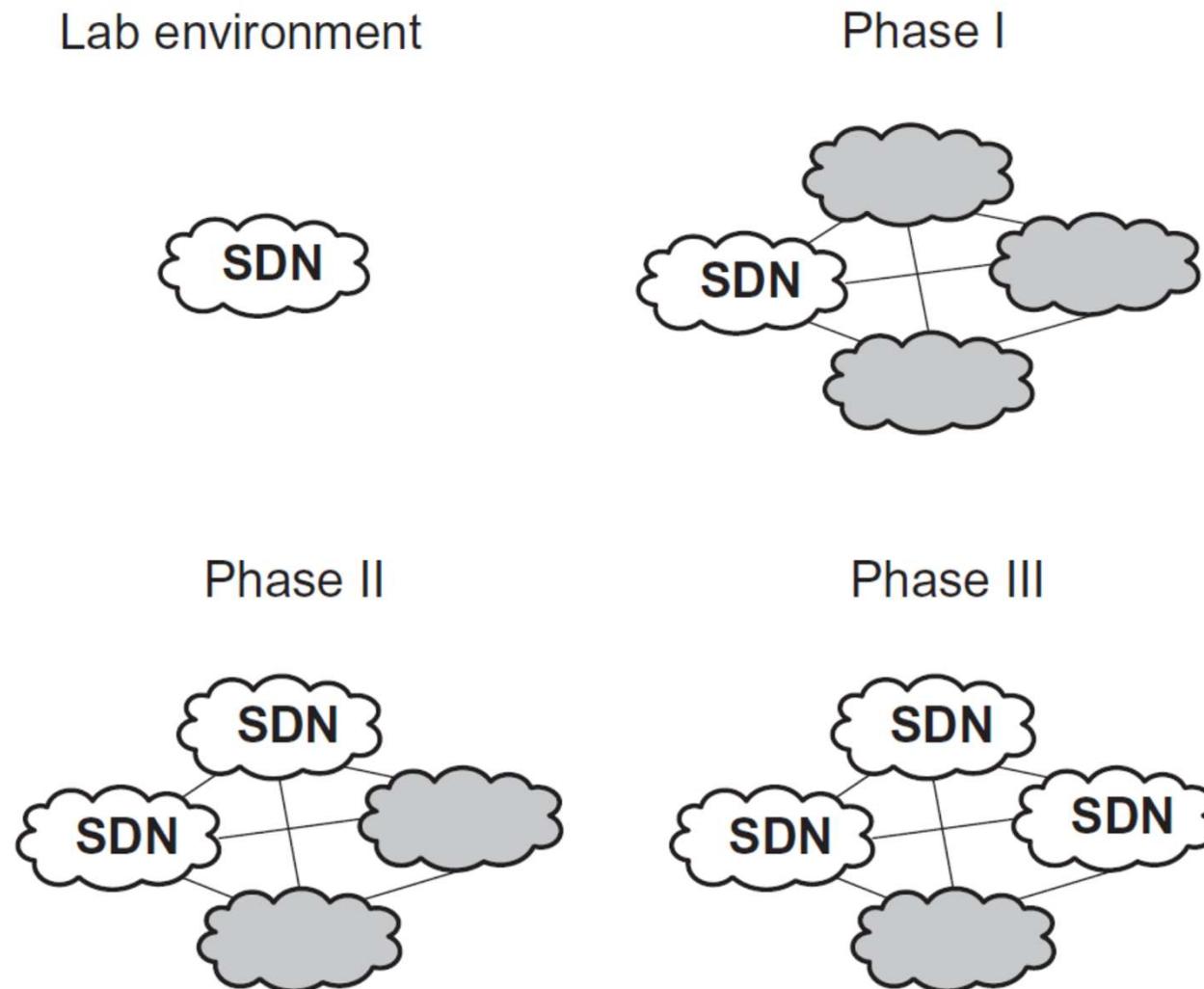


Controller deployment

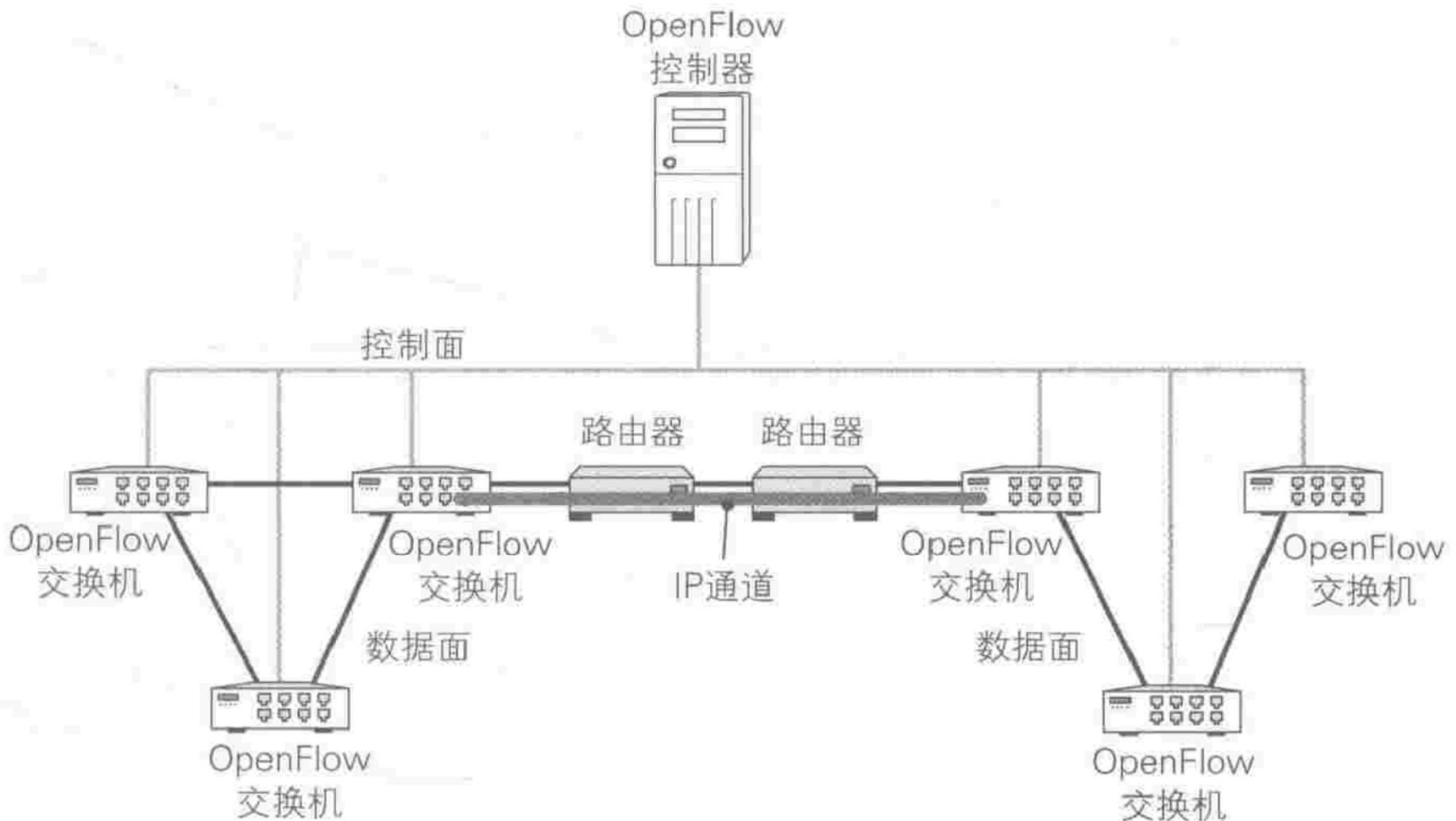
- Multilayer



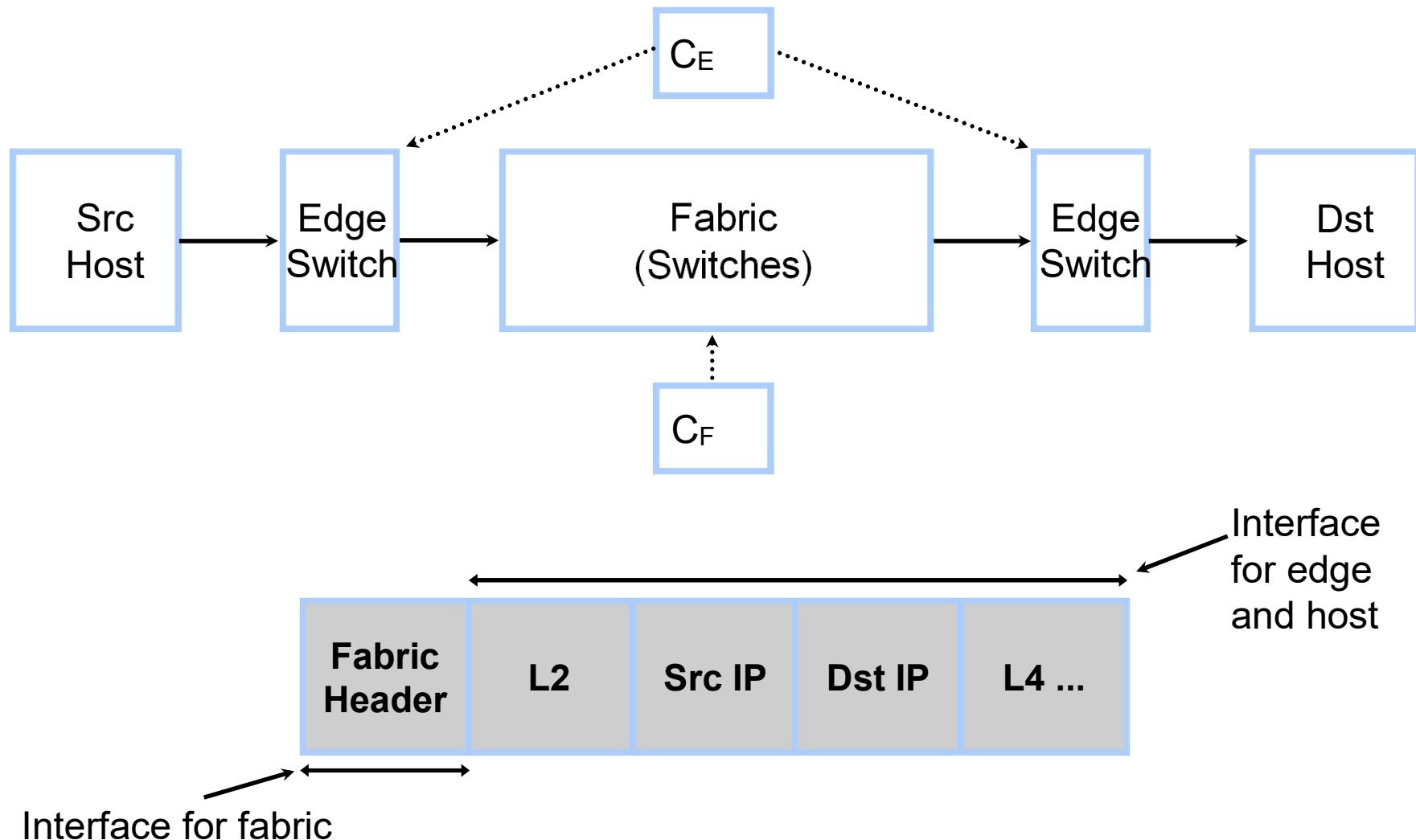
Phased deployment of SDN



Hybrid deployment



SDN + MPLS = “Fabric”

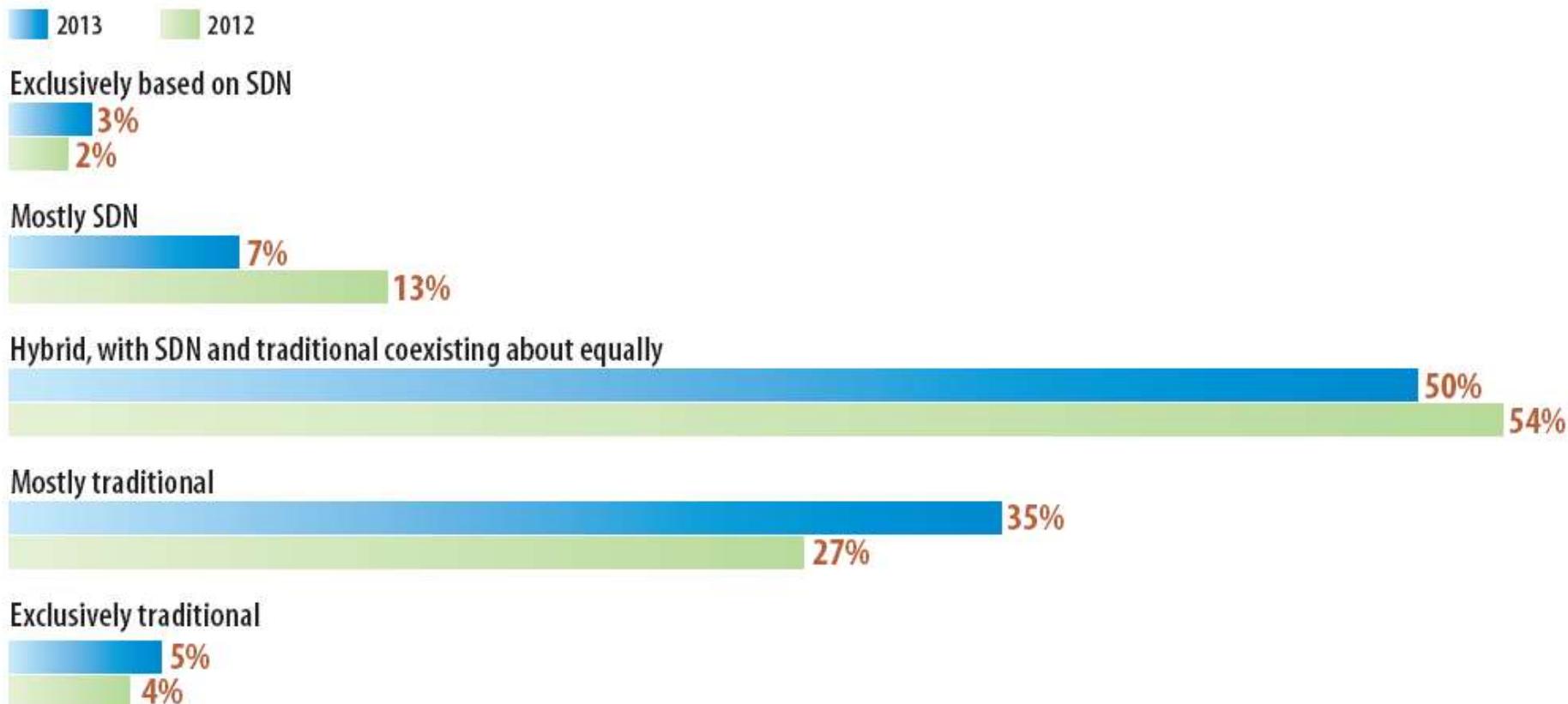


Hybrid network

InformationWeek
:: reports

Predicted Network Format

In five years, I believe my organization's networks will be:



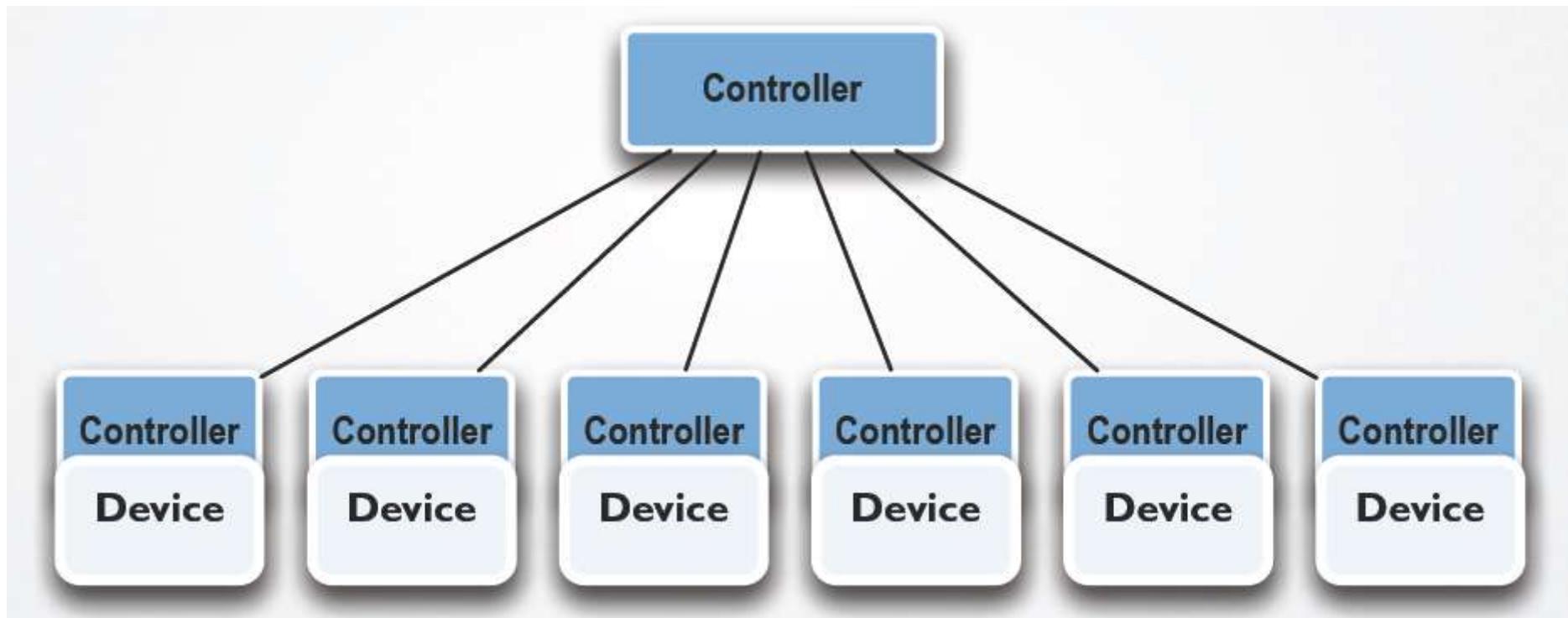
Base: 267 respondents in July 2013 and 250 in July 2012

Data: InformationWeek Software-Defined Networking Survey of business technology professionals

R7240813/4

116

OpenFlow-enabled switch



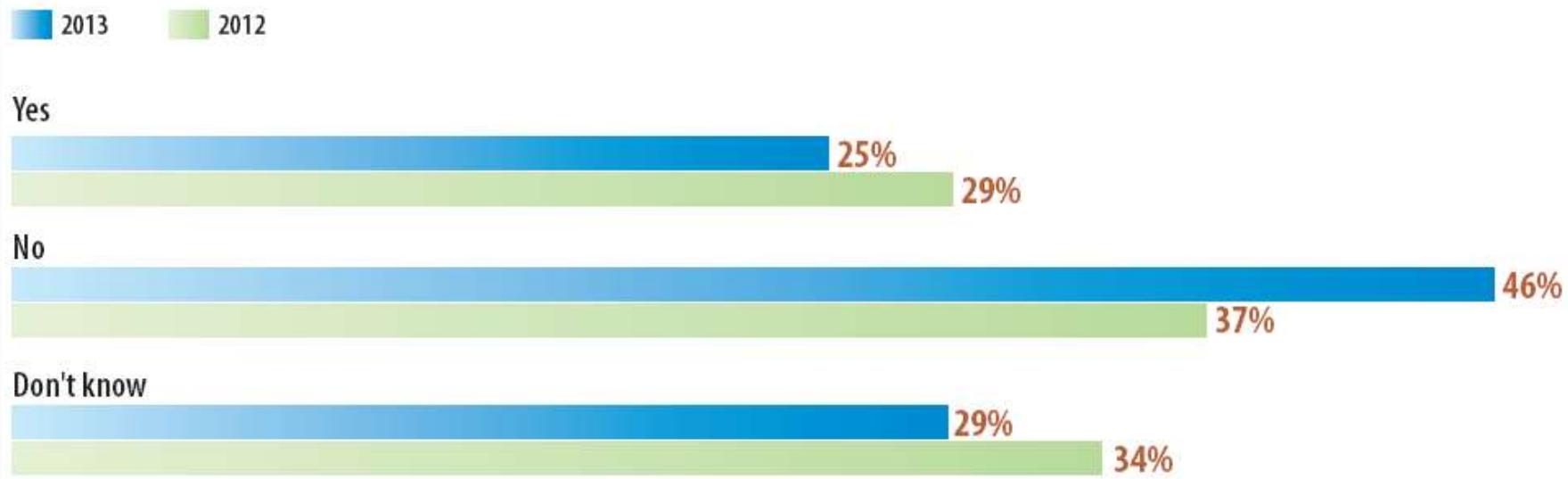
OpenFlow-enabled switch

- Two kinds of switches/routers
 - OpenFlow-enabled /hybrid
 - OpenFlow-only/pure OpenFlow

'Dumbing Down' of Switches and Routers?

InformationWeek
:: reports

Do you believe that SDN will relegate switches and routers to being just relatively dumb forwarding engines?



Base: 267 respondents in July 2013 and 250 in July 2012

R7240813/12

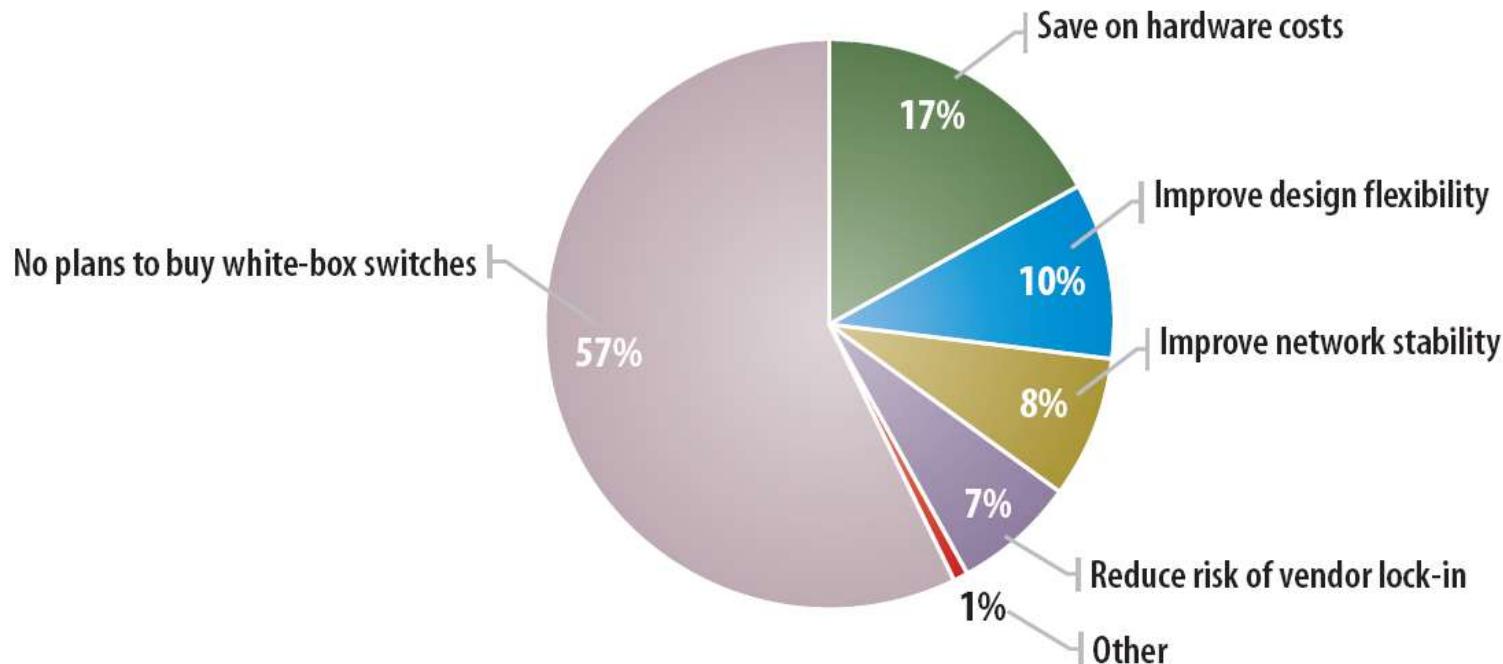
Data: InformationWeek Software-Defined Networking Survey of business technology professionals

OpenFlow-only switch

InformationWeek
:: reports

Reasons for Buying 'White-Box' Switches

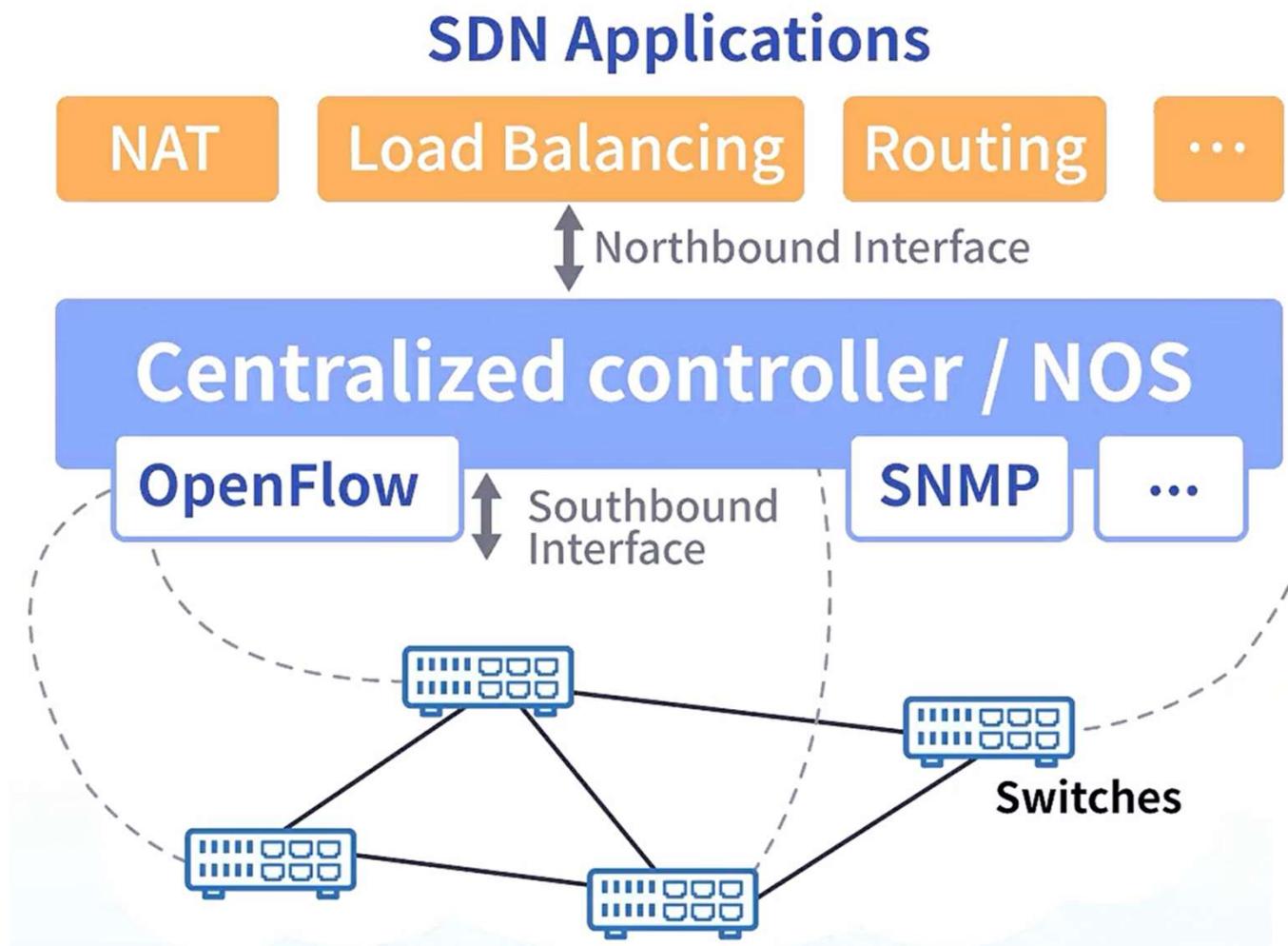
If you currently or plan to buy generic "white-box" switches, what's your top motivator?



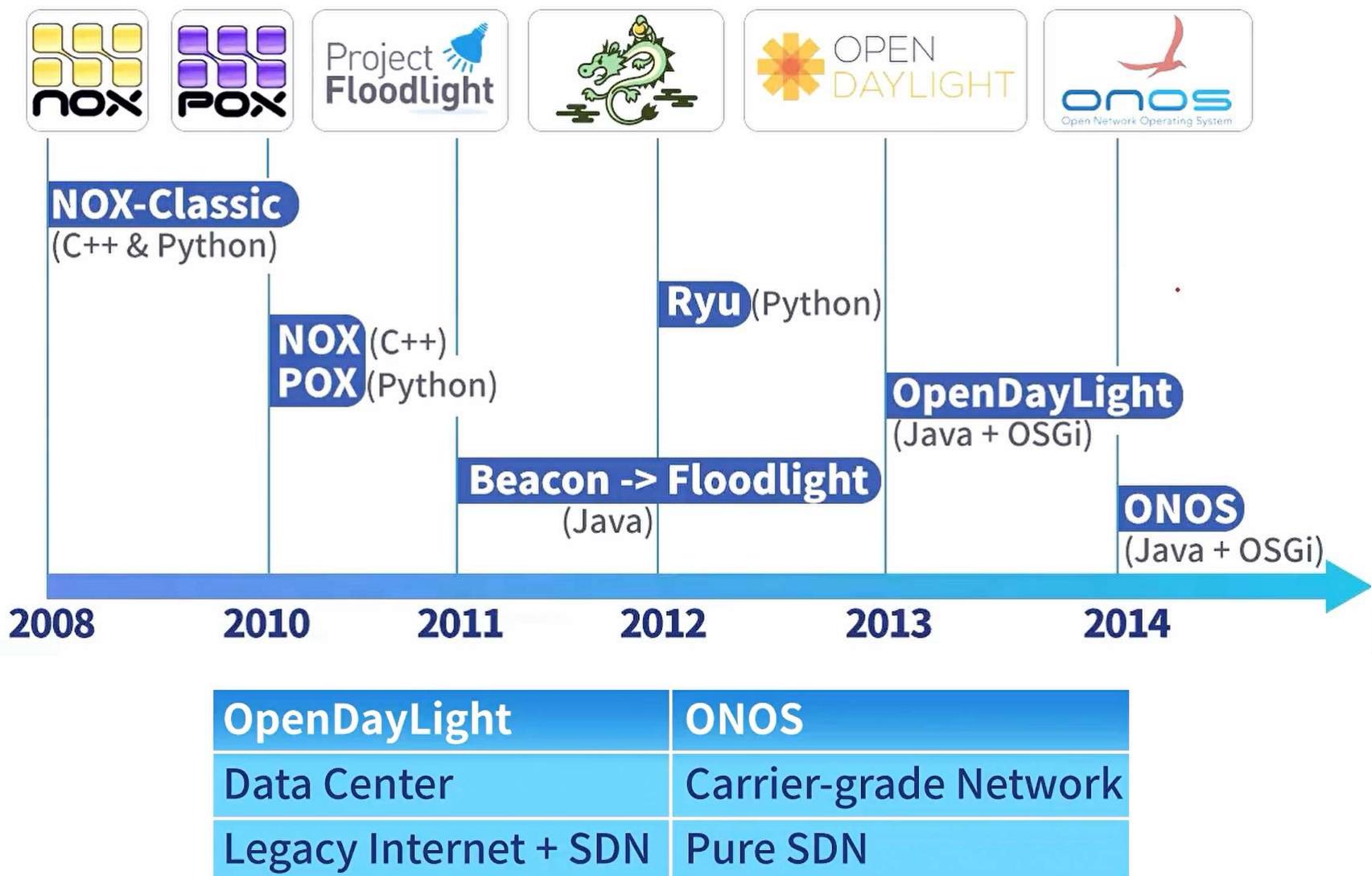
Data: InformationWeek 2013 Software-Defined Networking Survey of 267 business technology professionals, July 2013

R7240813/21

SDN controller



Mainstream controller software



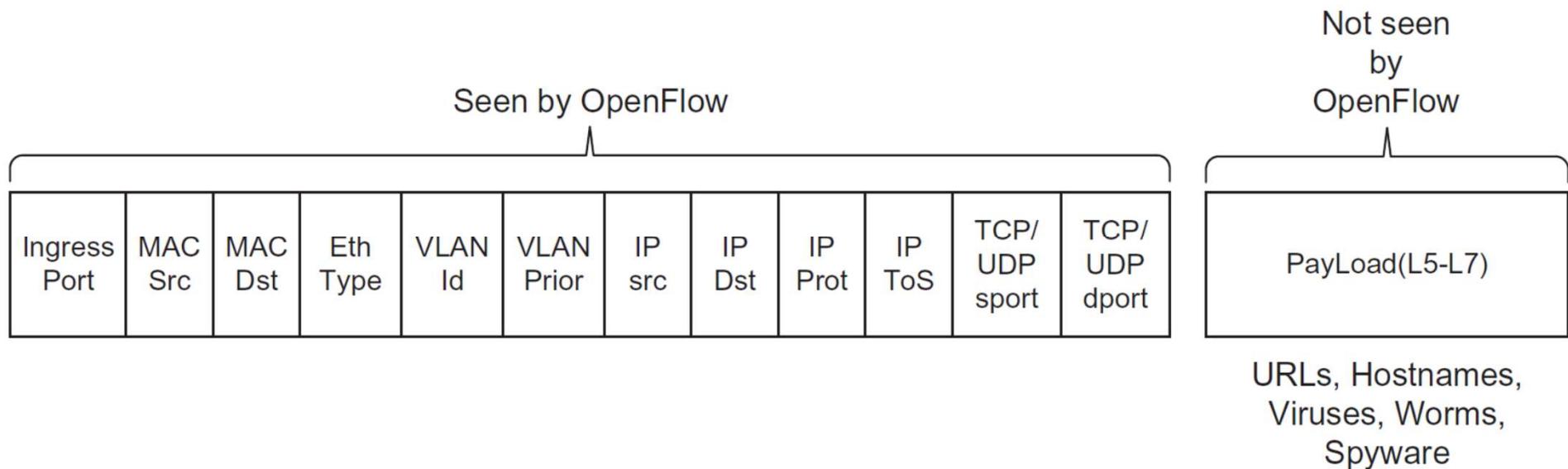
Today's agenda

- Introduction to Software Defined Networking
 - What is SDN? Why SDN?
 - History of SDN
 - SDN standardization
 - SDN deployment models
 - OpenFlow
 - SDN applications

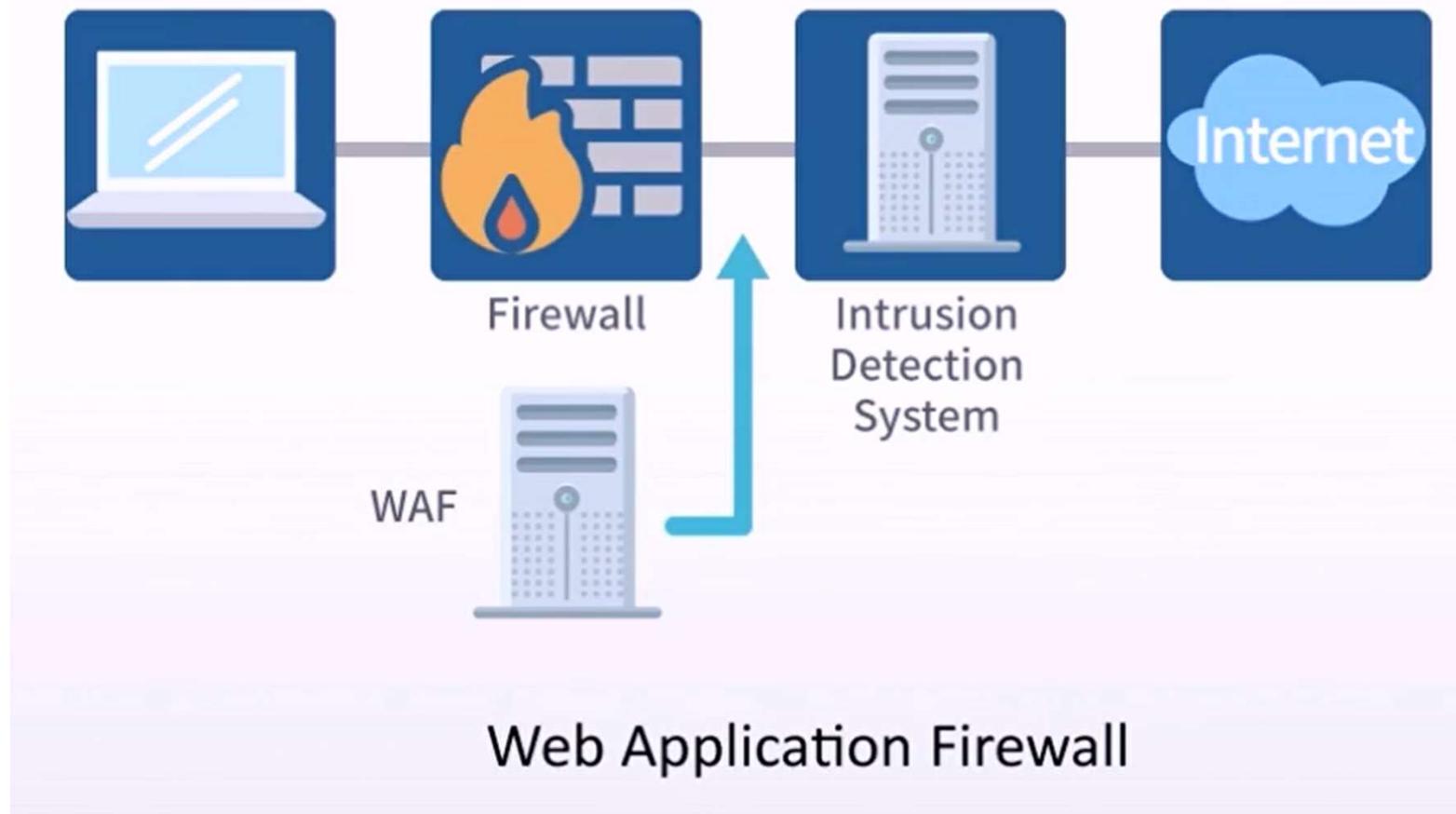
SDN applications

- SDN+NFV
- Wireless SDN
- SDN for Data center
- SDWAN

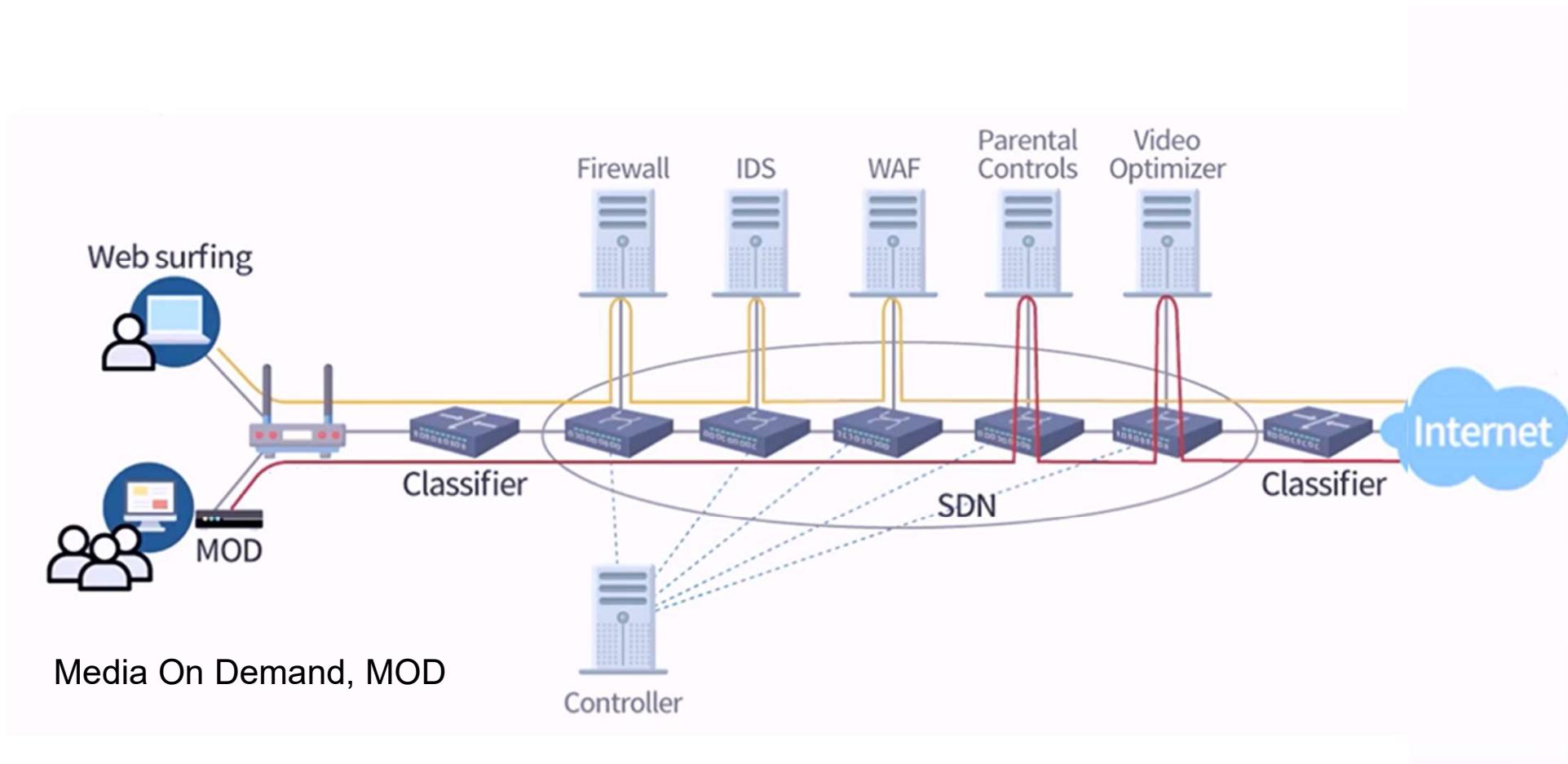
Deep packet inspection



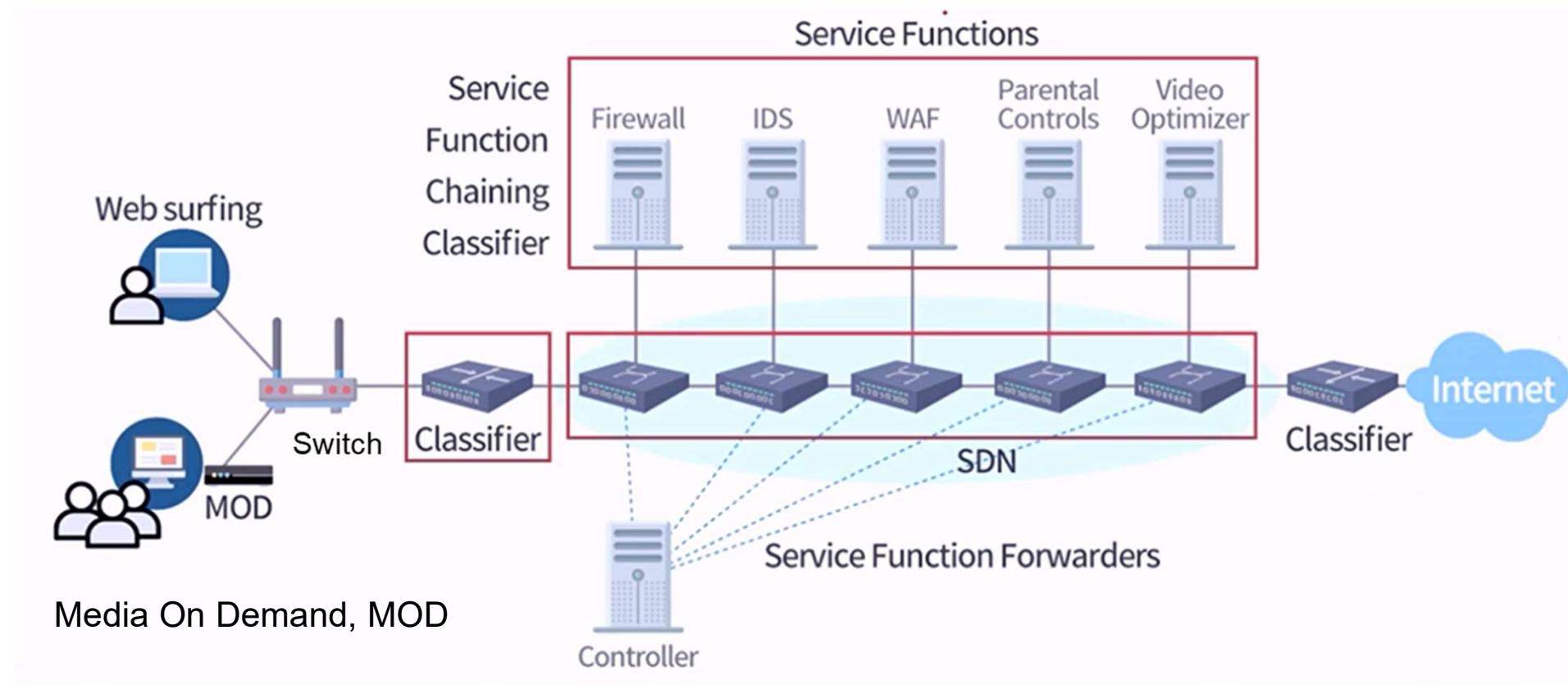
Traditional Middlebox



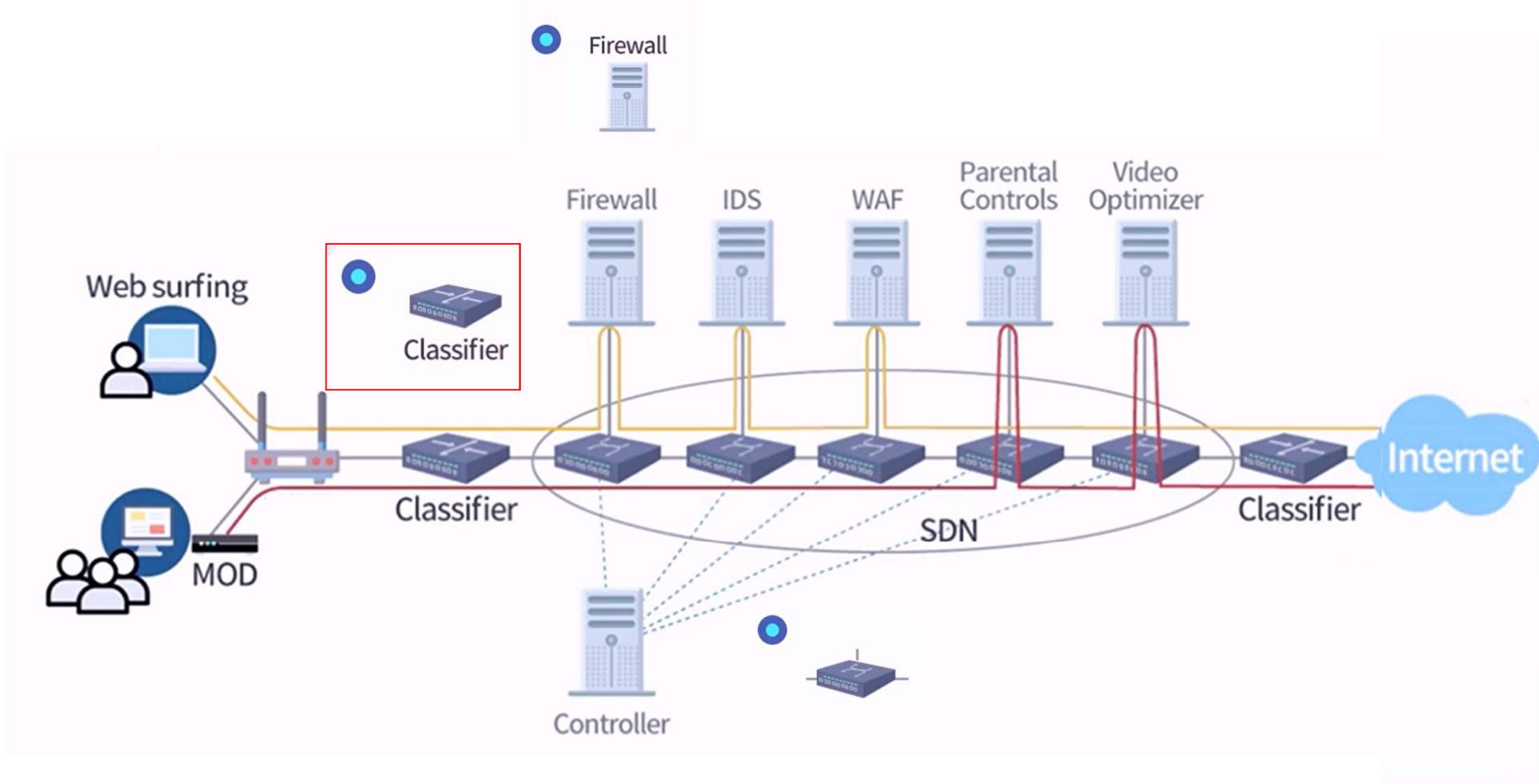
SDN+NFV



SDN+NFV



SDN+NFV

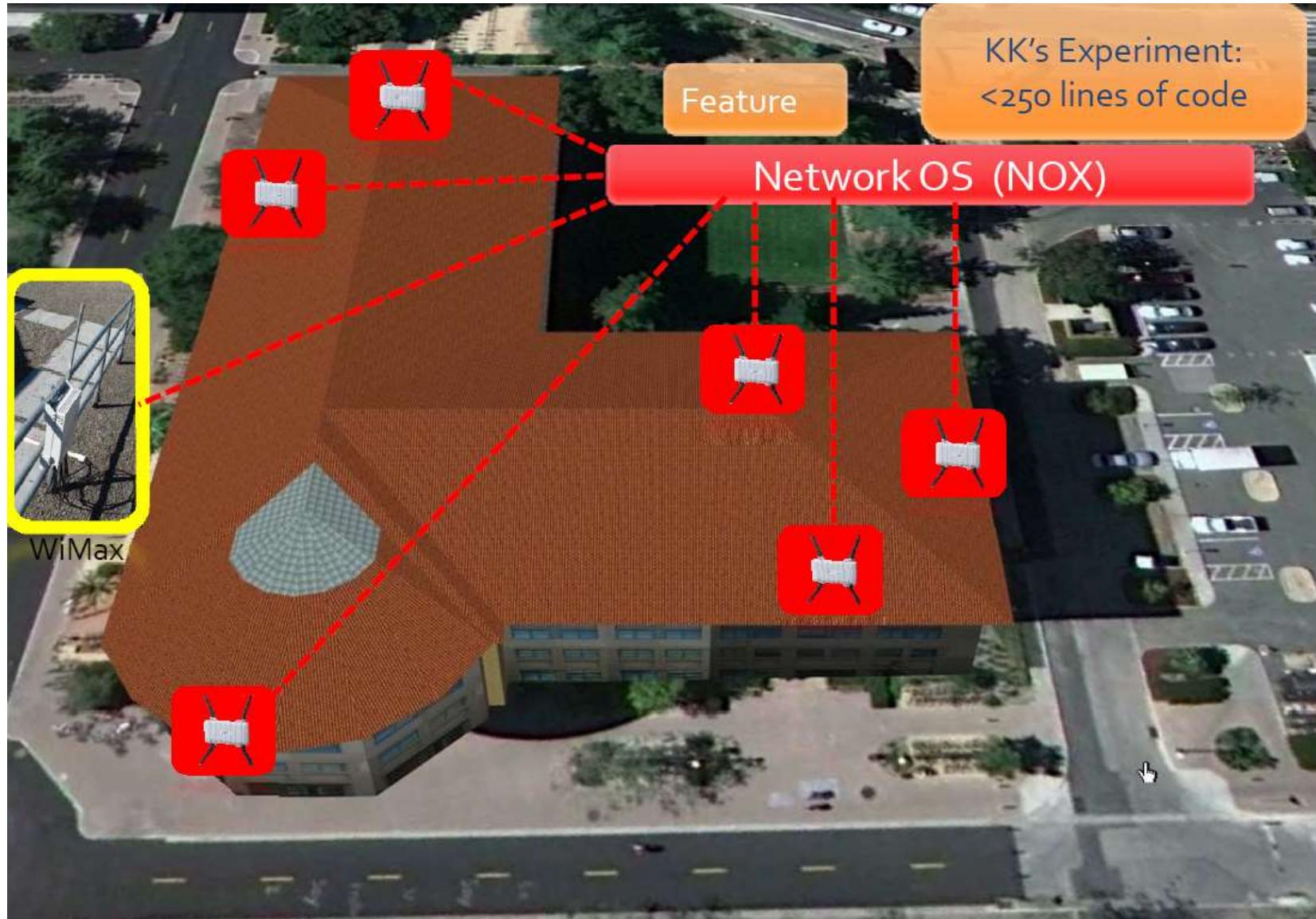


SDN applications

- SDN+NFV
- Wireless SDN
- SDN for Data center
- SDWAN

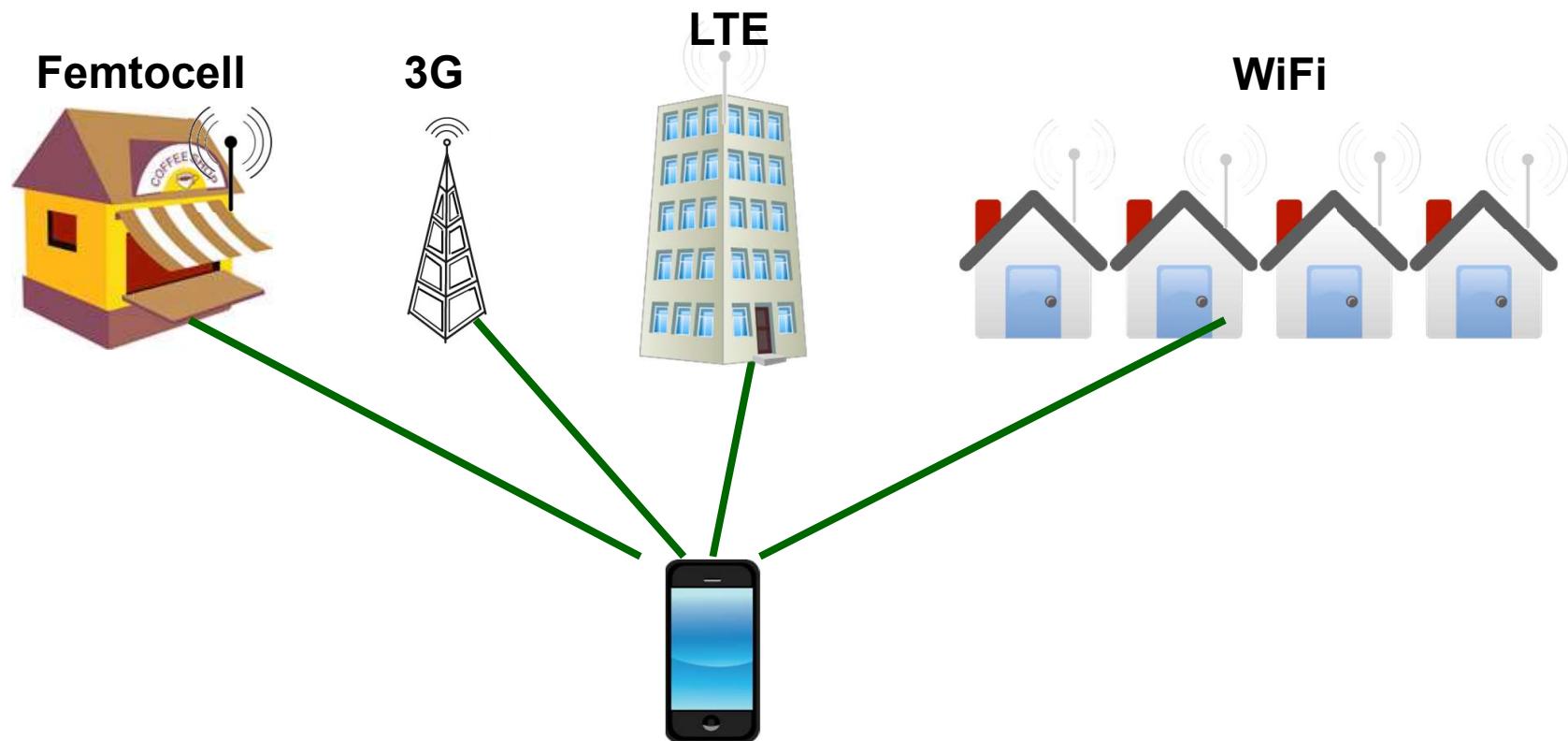
Wireless SDN

- Using all the wireless capacity around us

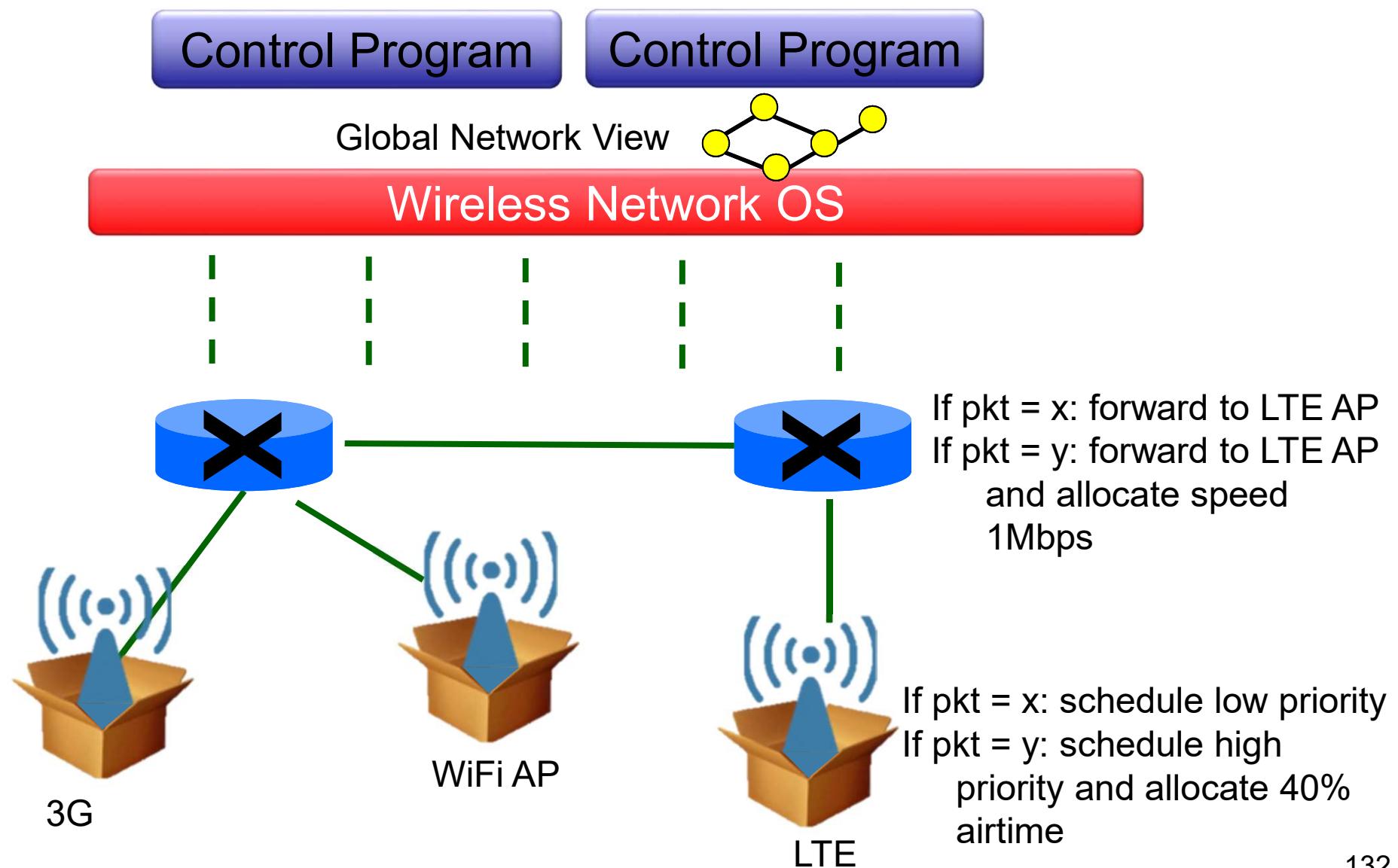


Wireless SDN

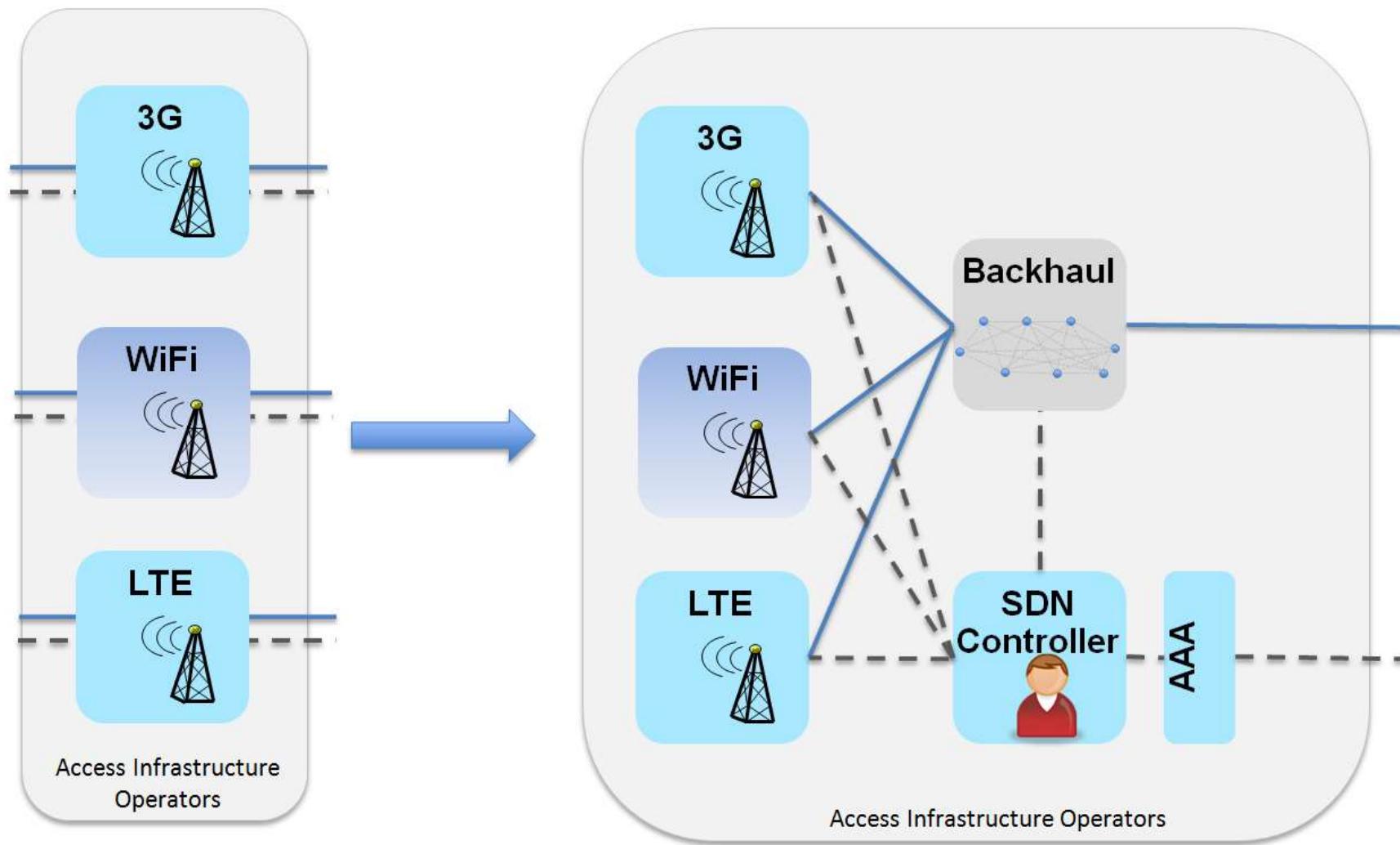
- Why can't I seamlessly connect me to the best AP available?
- Why can't I seamlessly connect to multiple APs if I want more speed?



Wireless SDN



Wireless SDN

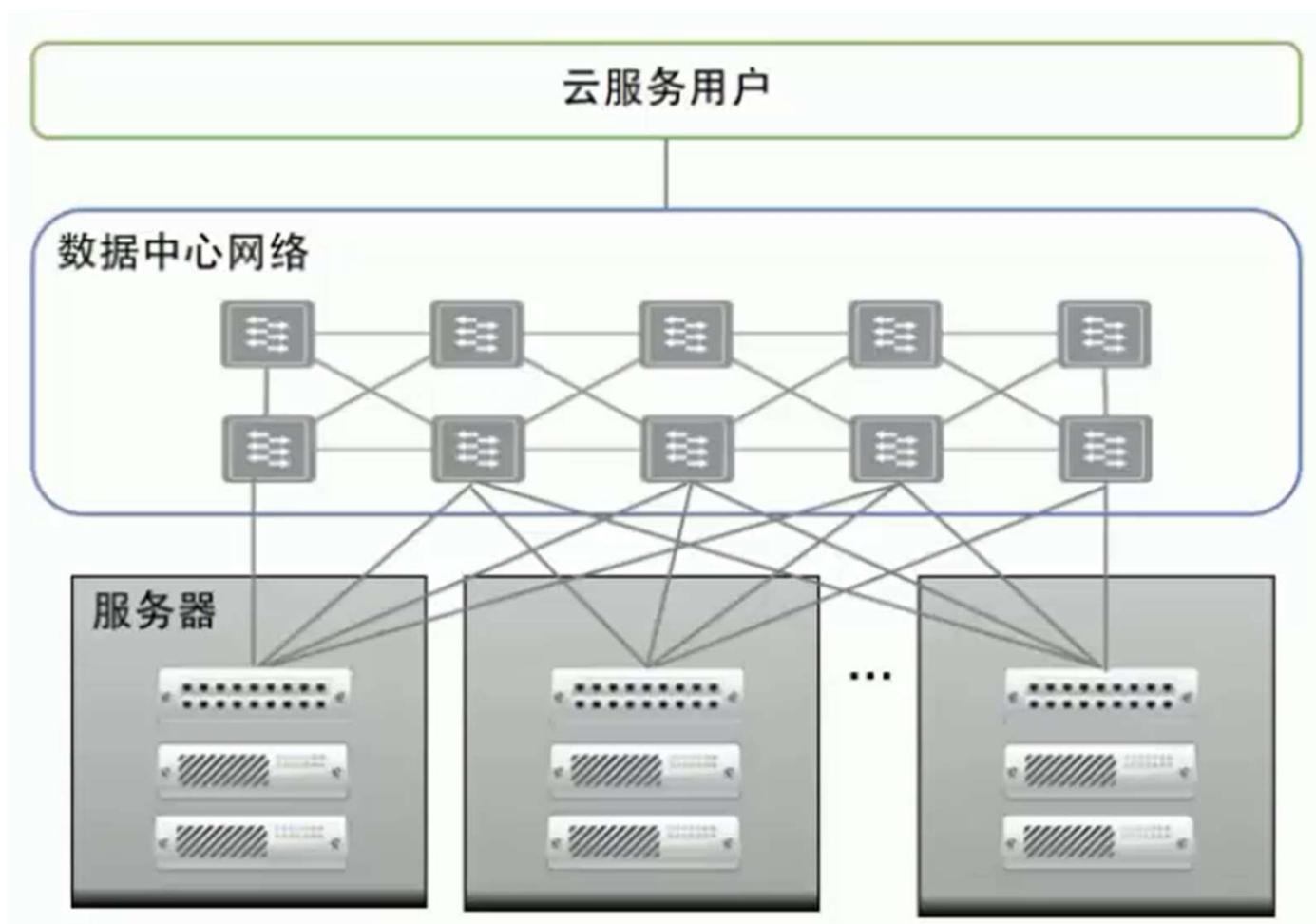


— Data path
- - - - Control path 133

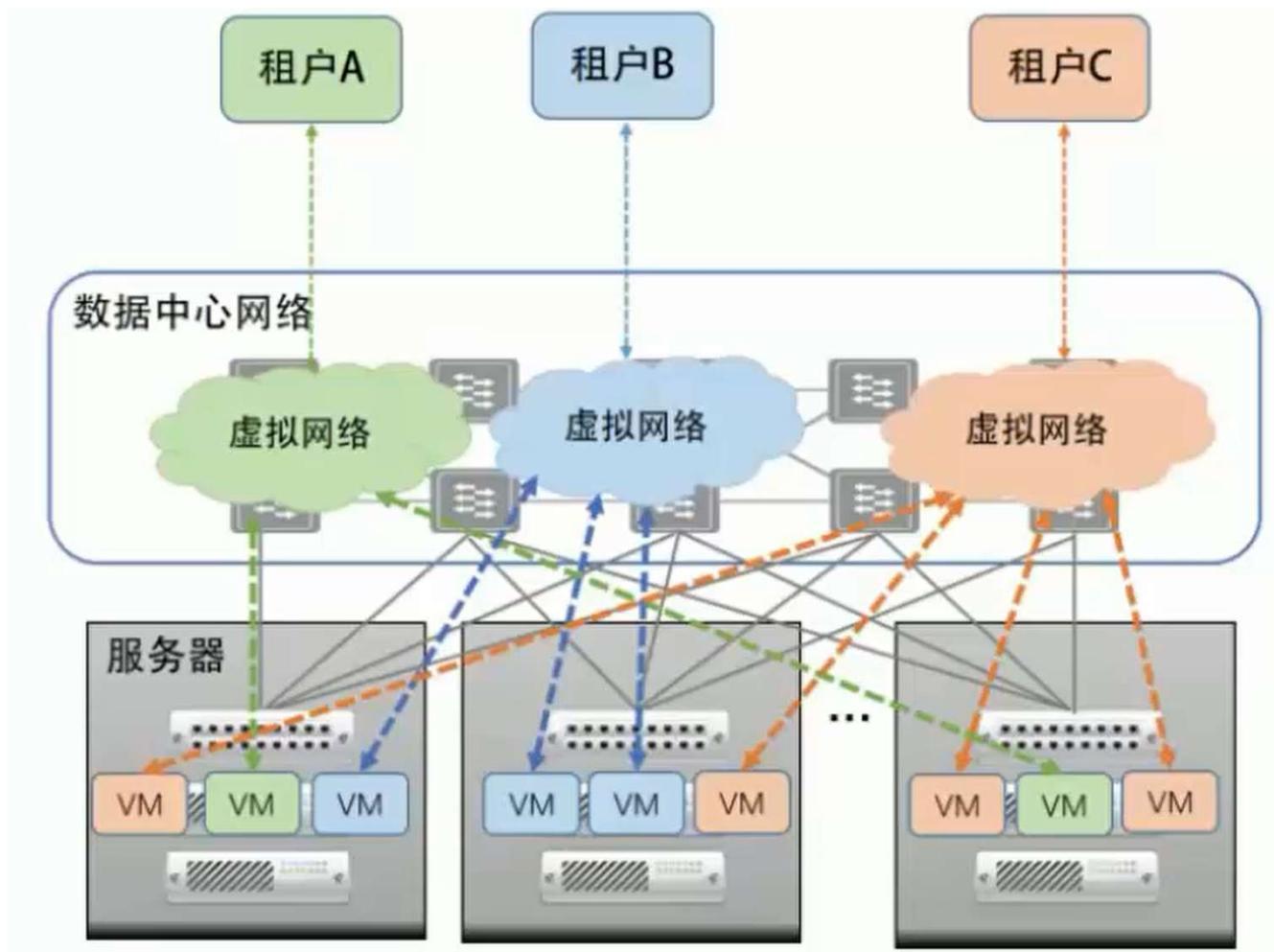
SDN applications

- SDN+NFV
- Wireless SDN
- **SDN for Data center**
- SDWAN

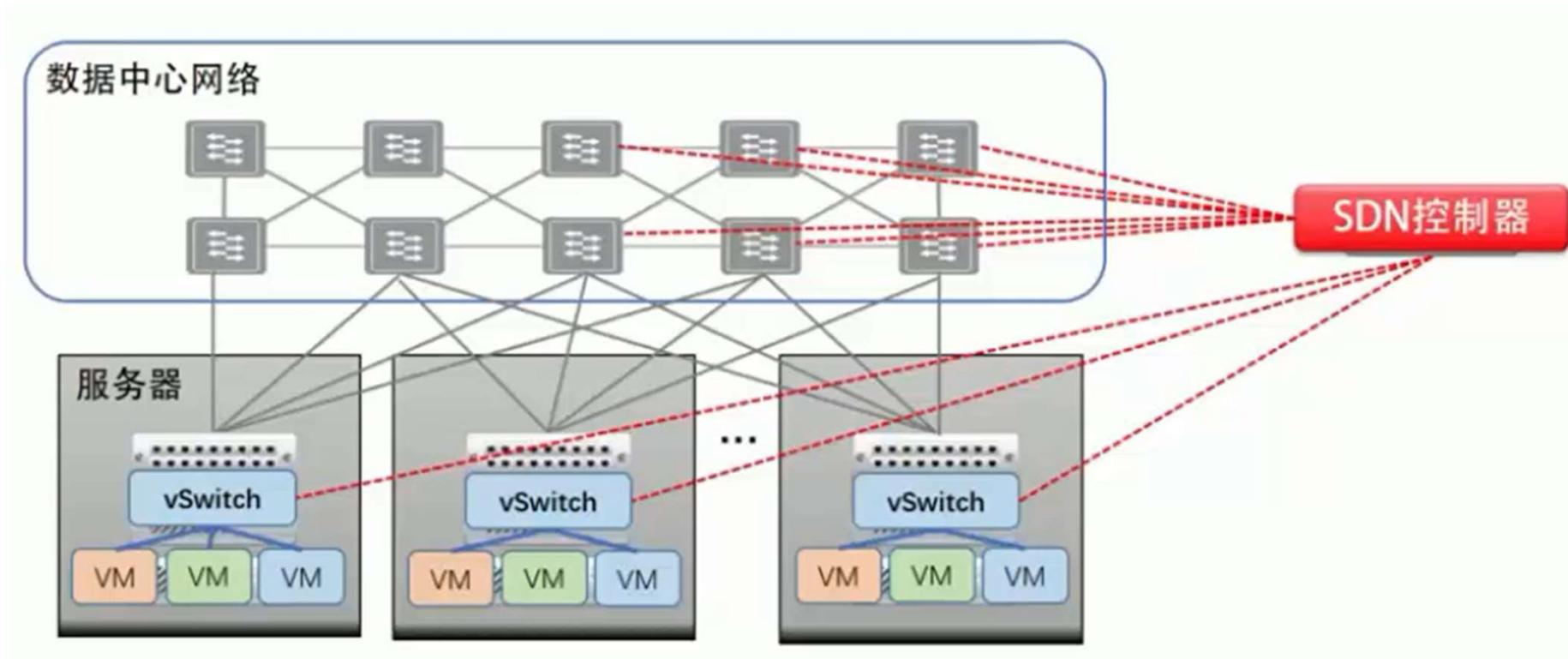
Data Center



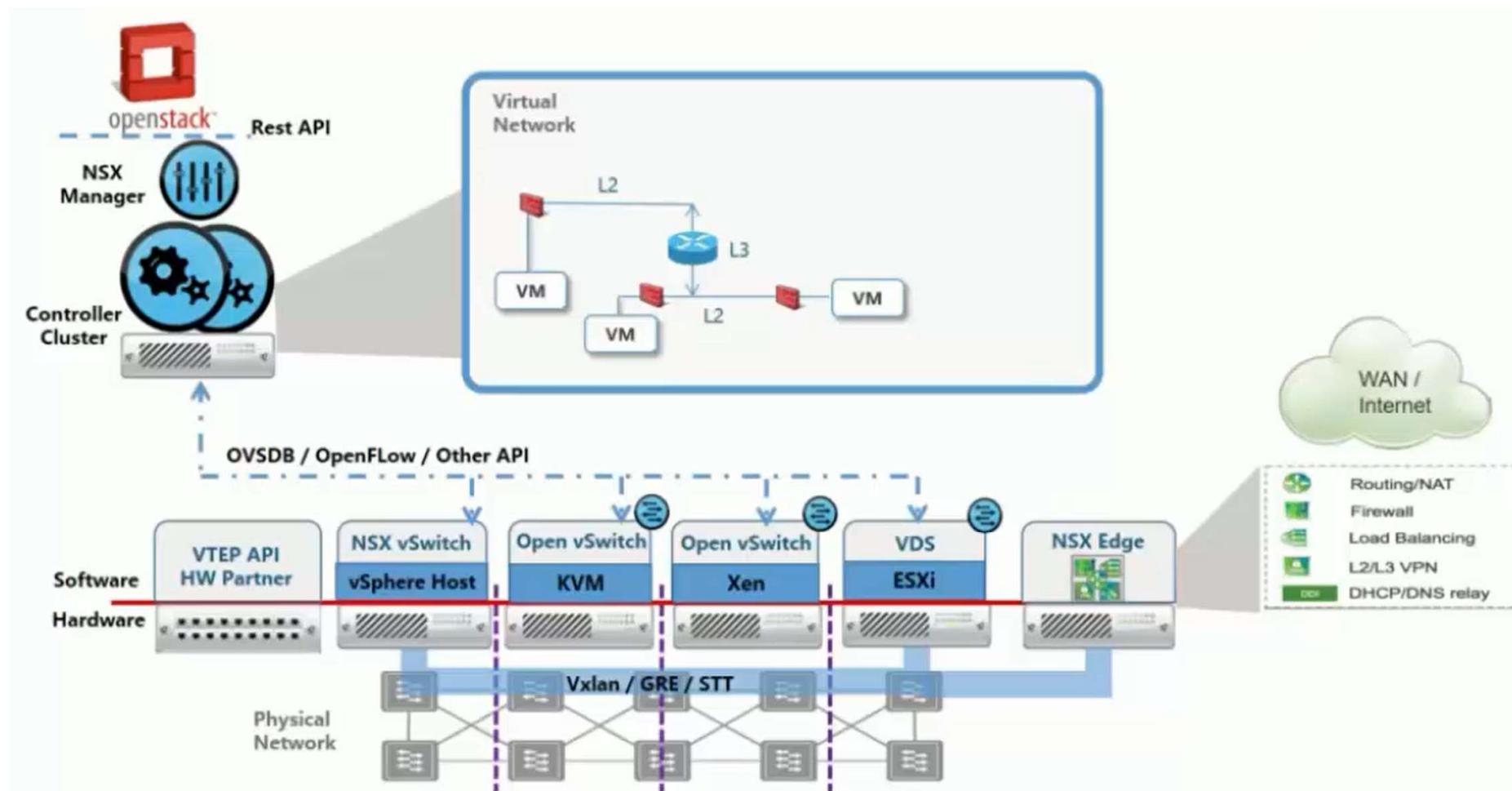
Multi-tenant applications



SDN for DC



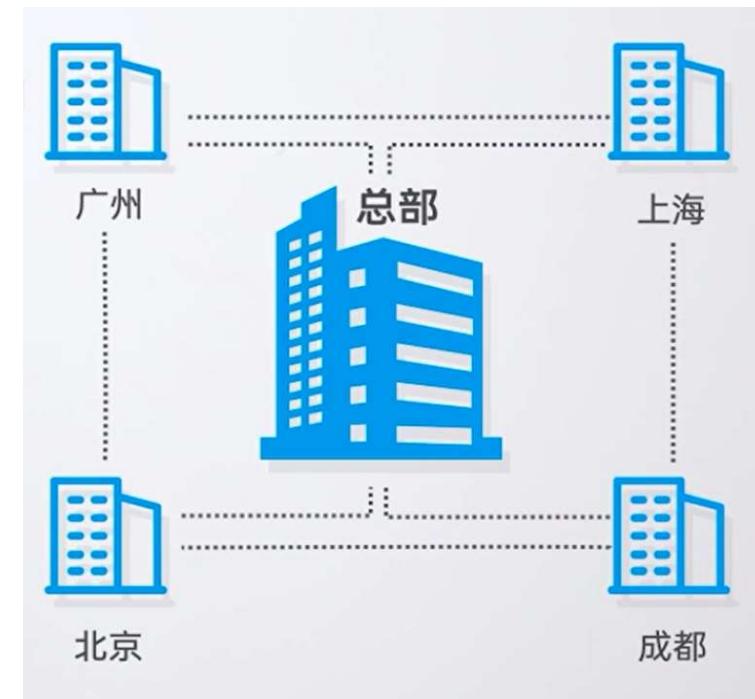
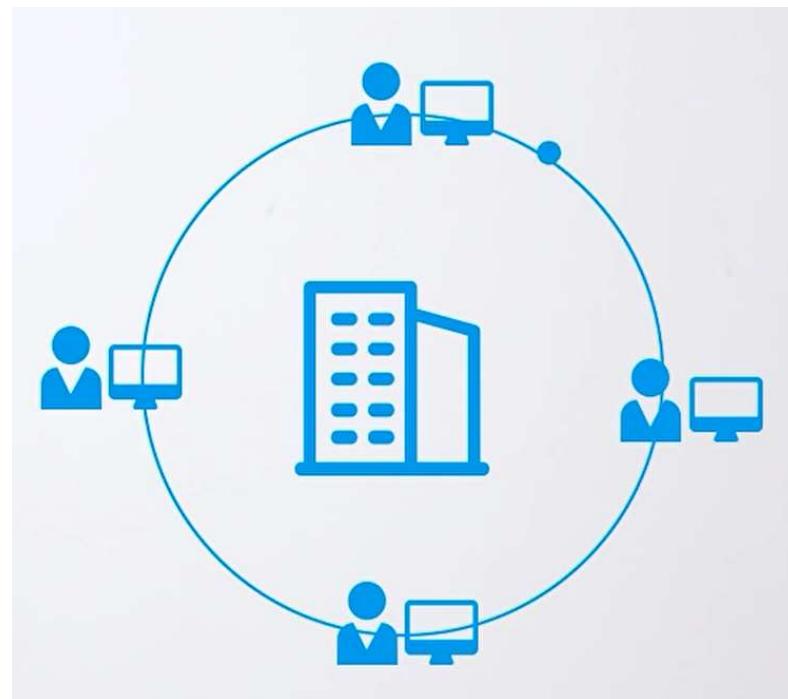
Vmware NSX



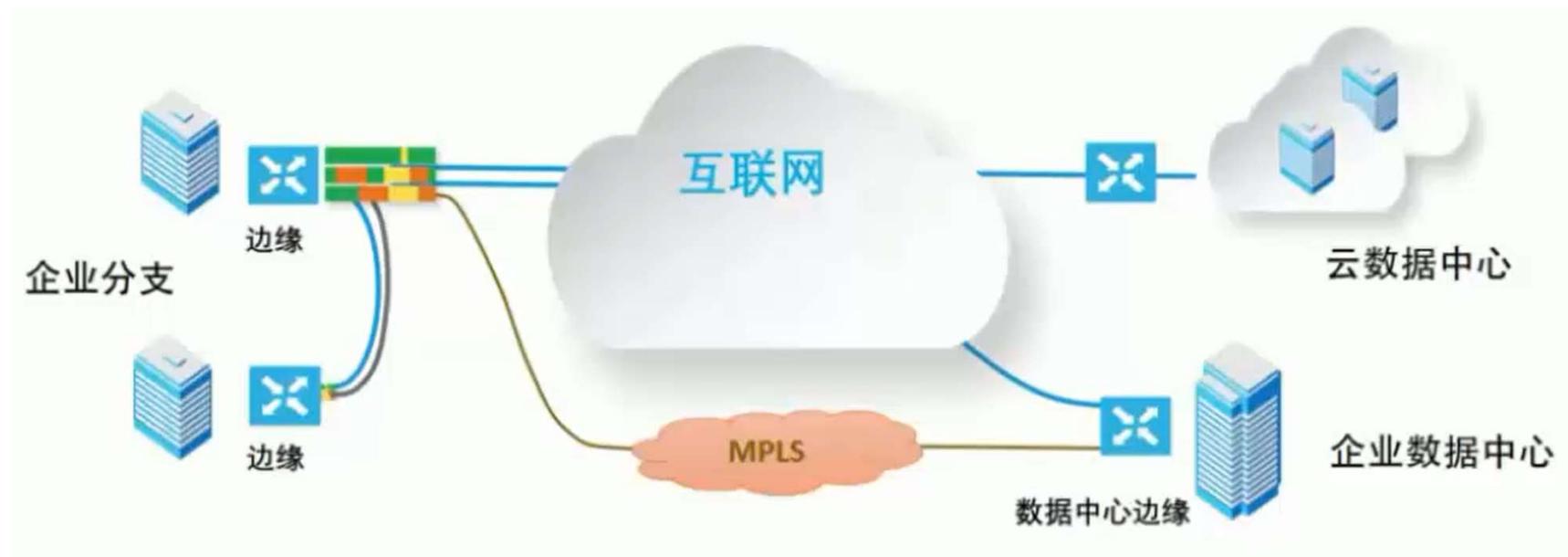
SDN applications

- SDN+NFV
- Wireless SDN
- SDN for Data center
- SDWAN

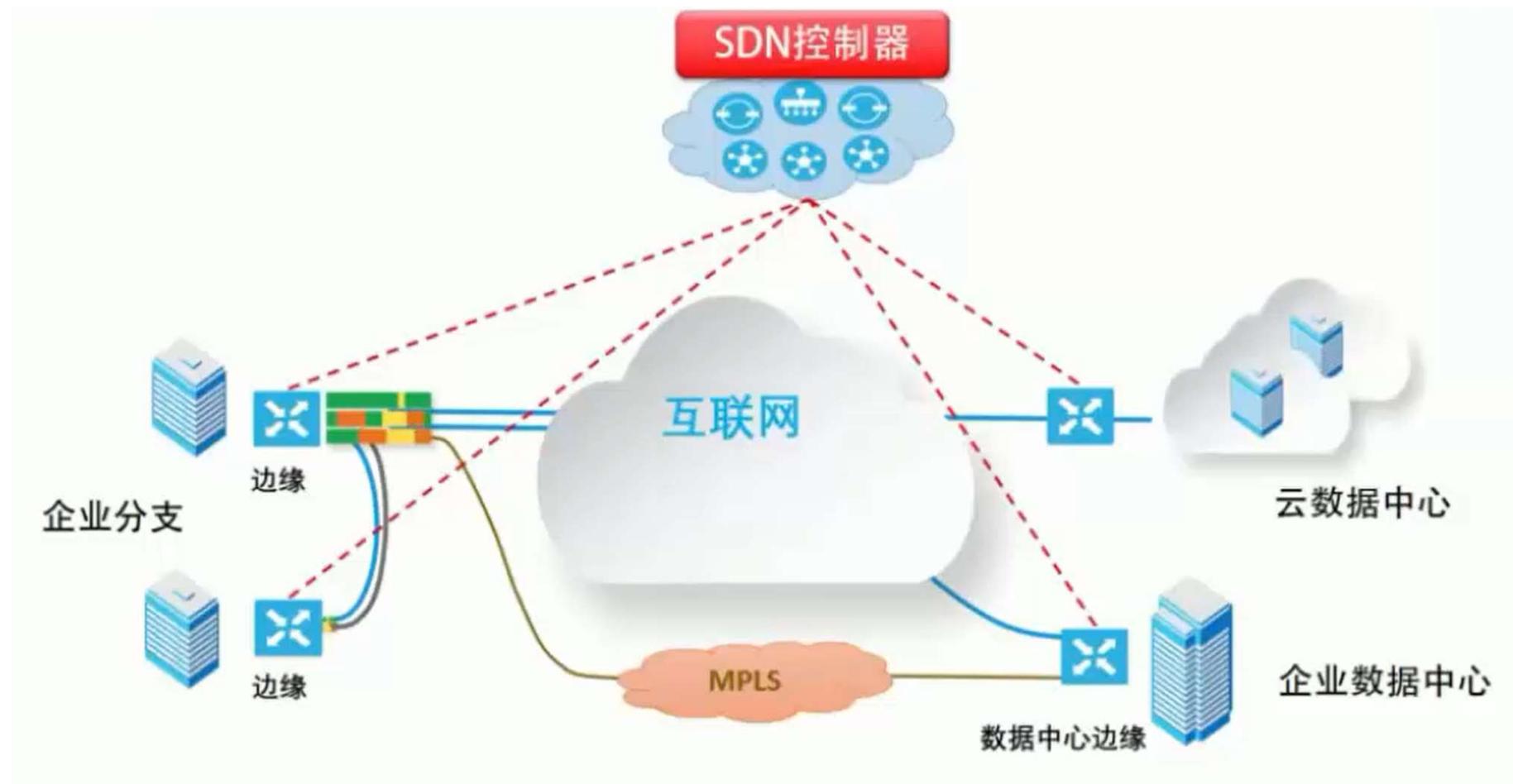
LAN vs. WAN



Wide Area Network (WAN)



Software-defined WAN



Software-defined WAN

