软件定义网络(SDN)

张喆

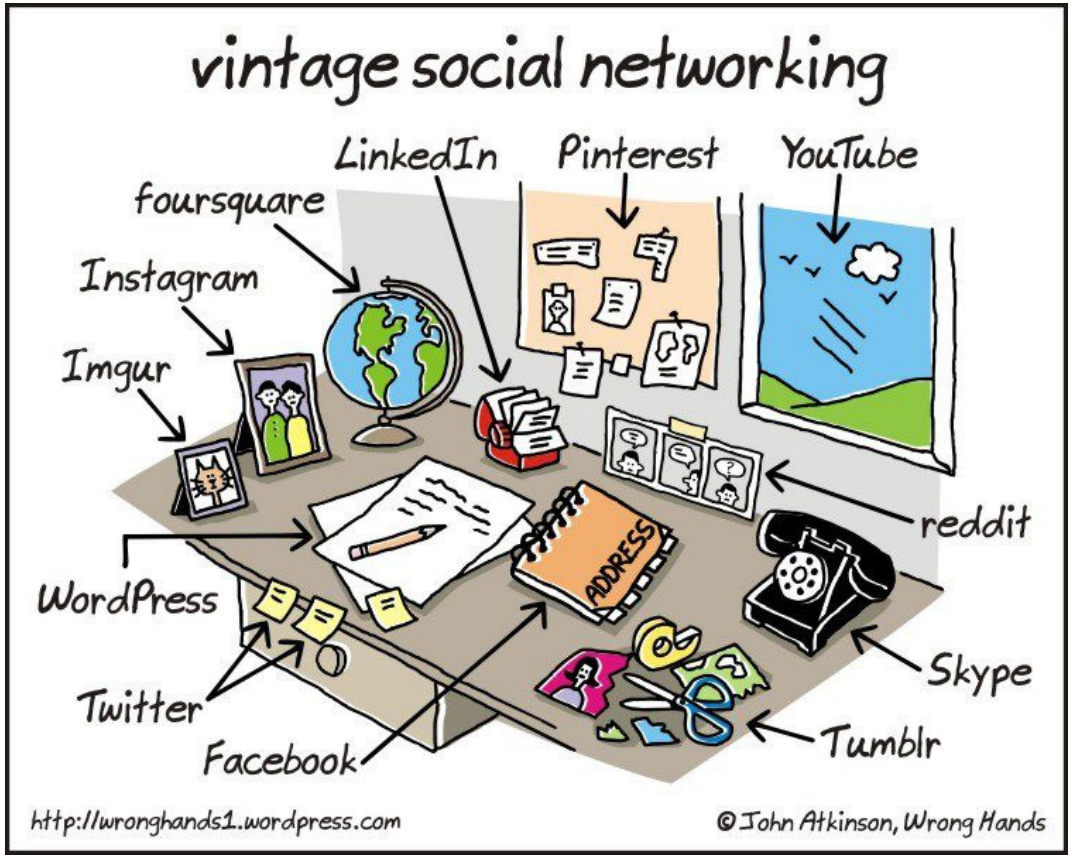
zhezhang@njupt.edu.cn

通信与信息工程学院

目　录

传统网络的局限性

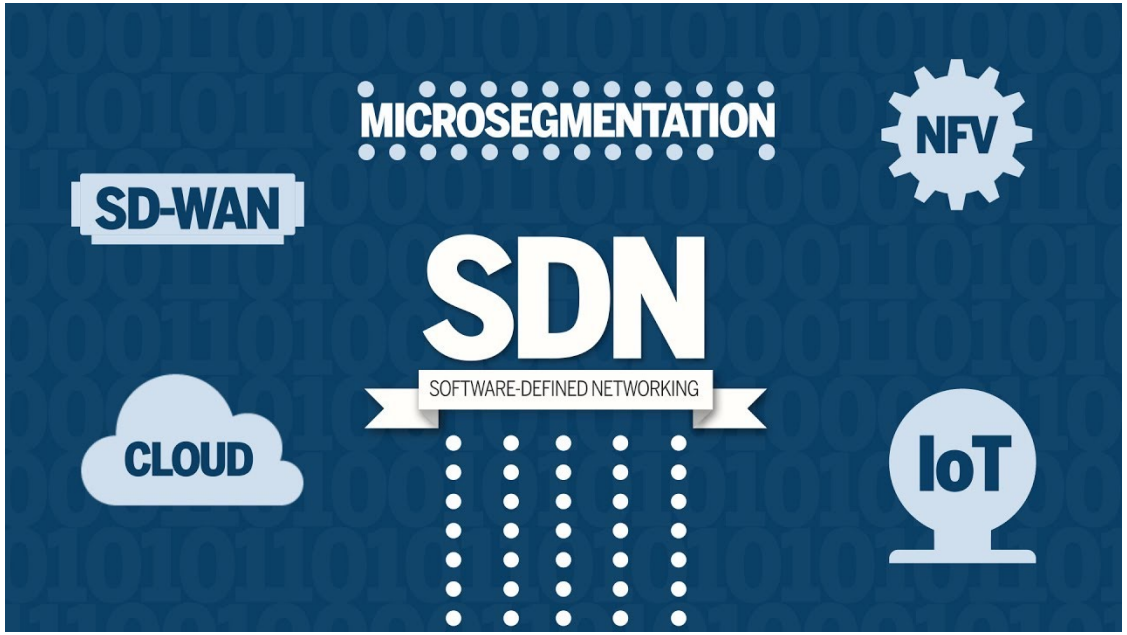
- 垂直集成和硬件依赖
- 缺乏灵活性和可编程性
- 管理和配置复杂性
- 缺乏全局视野和集中控制
- 限制快速部署与创新
- 难以适应动态变化和需求



根本原因：网络设备独立决策，缺乏全局视野

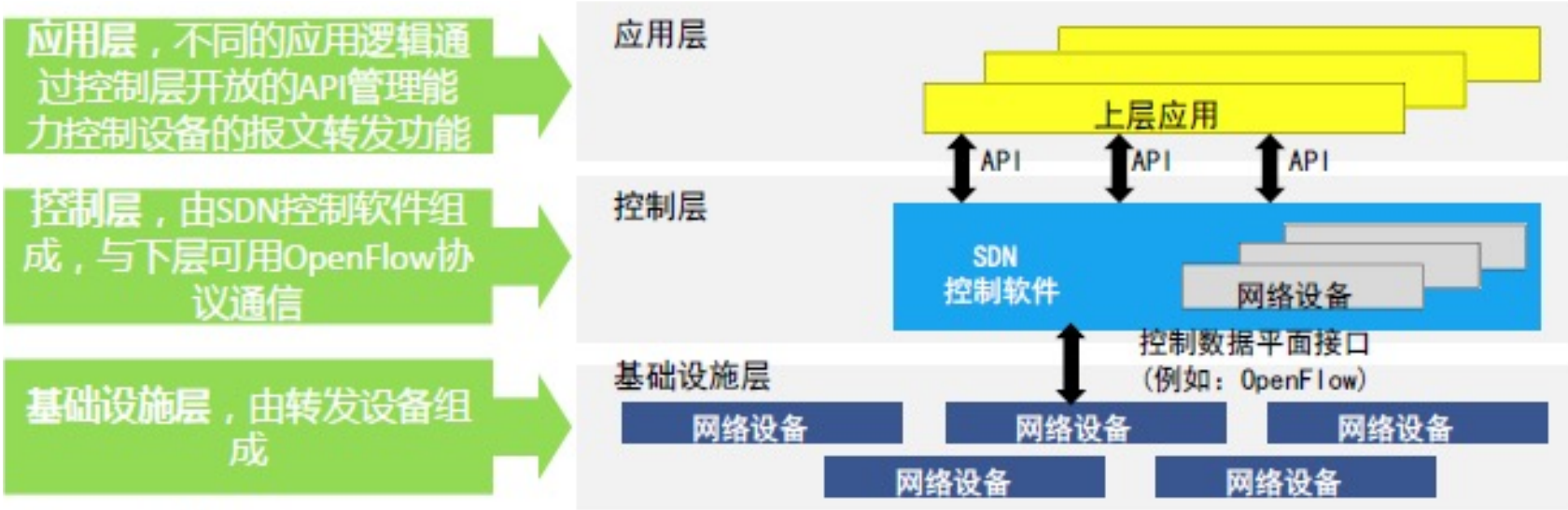
软件定义网络（SDN）

- Solution：设置一个具有全局视野的控制器
  - SDN controller
- 网络设备：无需再独立做决策
  - 需要将控制面与数据面解耦



软件定义网络（SDN）

- 软件定义网络（Software Defined Networking, SDN）是一种新型的网络技术，其设计理念是将网络的控制平面与数据转发平面进行分离，并实现可编程化的集中控制。
- 传统网络设备紧耦合的网络架构被分拆成应用、控制、转发三层分离的架构。控制功能被转移到了服务器，上层应用、底层转发设施被抽象成多个逻辑实体。



传统网络 VS SDN

特点	传统网络	SDN
控制方式	分布式	集中式
网络设备	复杂，具有自主运行的能力	简单，决策由SDN controller完成
网络管理和配置	独立且分散的，复杂性极高	管理简单，由SDN controller统一管理
灵活性和可编程性	网络功能通常固化在设备上，难以快速扩展	灵活，可编程
创新和快速部署	设备升级严重依赖设备商，部署速度慢	有利于创新，能够实现快速部署

SDN特征

集中控制

✓集中控制使得全局优化成为可能，比如流量工程、负载均衡
✓集中控制使得整个网络可以当作一台设备进行维护，设备零配置即插即用，大大降低运维成本，类似的技术:

开放接口

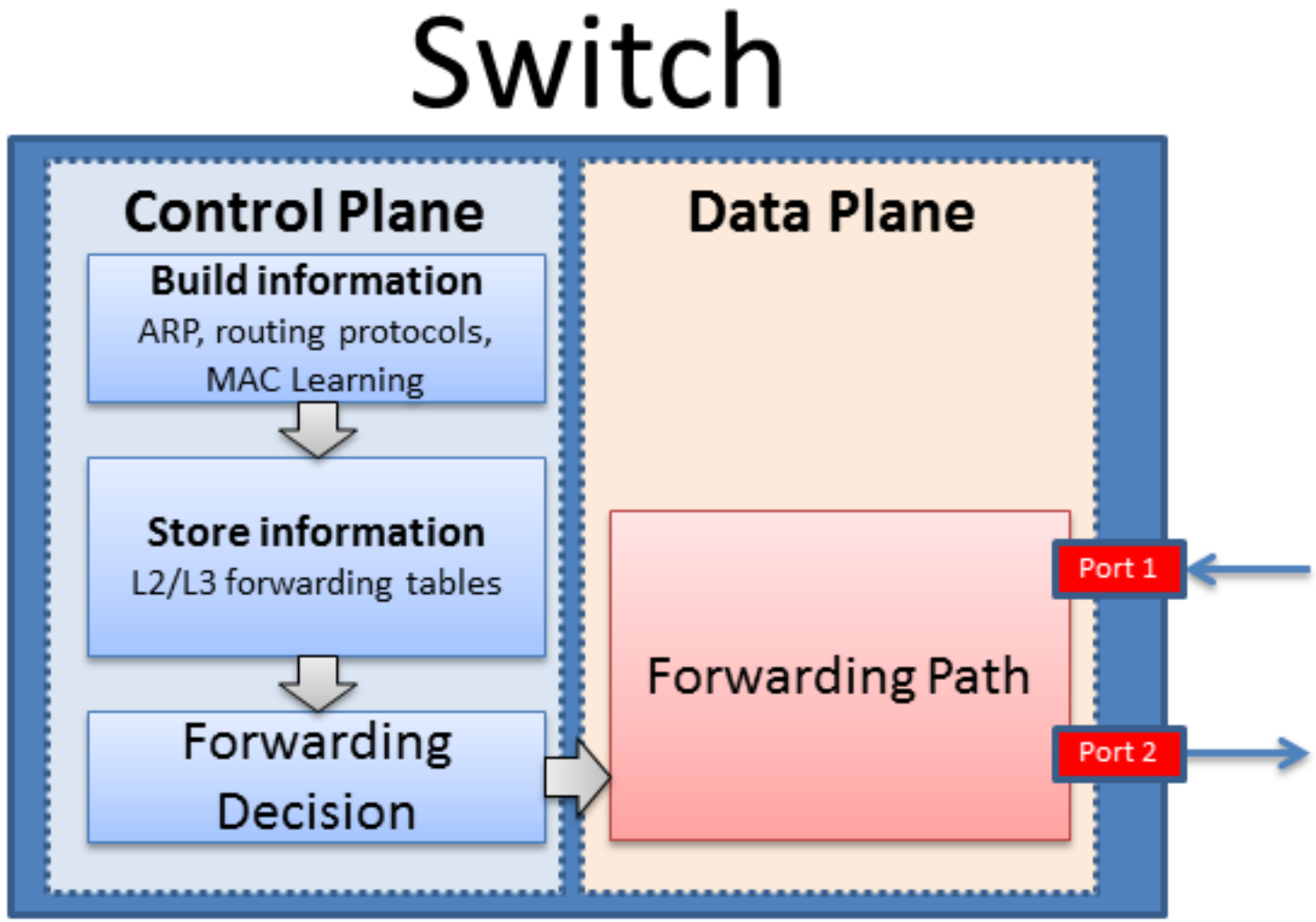
✓应用和网络的无缝集成，应用告诉网络如何运行才能更好地满足应用的需求，比如业务的带宽、时延需求，计费对路由的影响等
✓用户可以自行开发网络新功能，加快新功能面世周期
✓理论上NOS和转发硬件的开放标准接口使得硬件完全PC化

网络虚拟化

✓逻辑网络和物理网络的分离，逻辑网络可以根据业务需要配置、迁移，不受物理位置的限制
✓多租户支持，每个租户可以自行定义带宽需求和私有编址

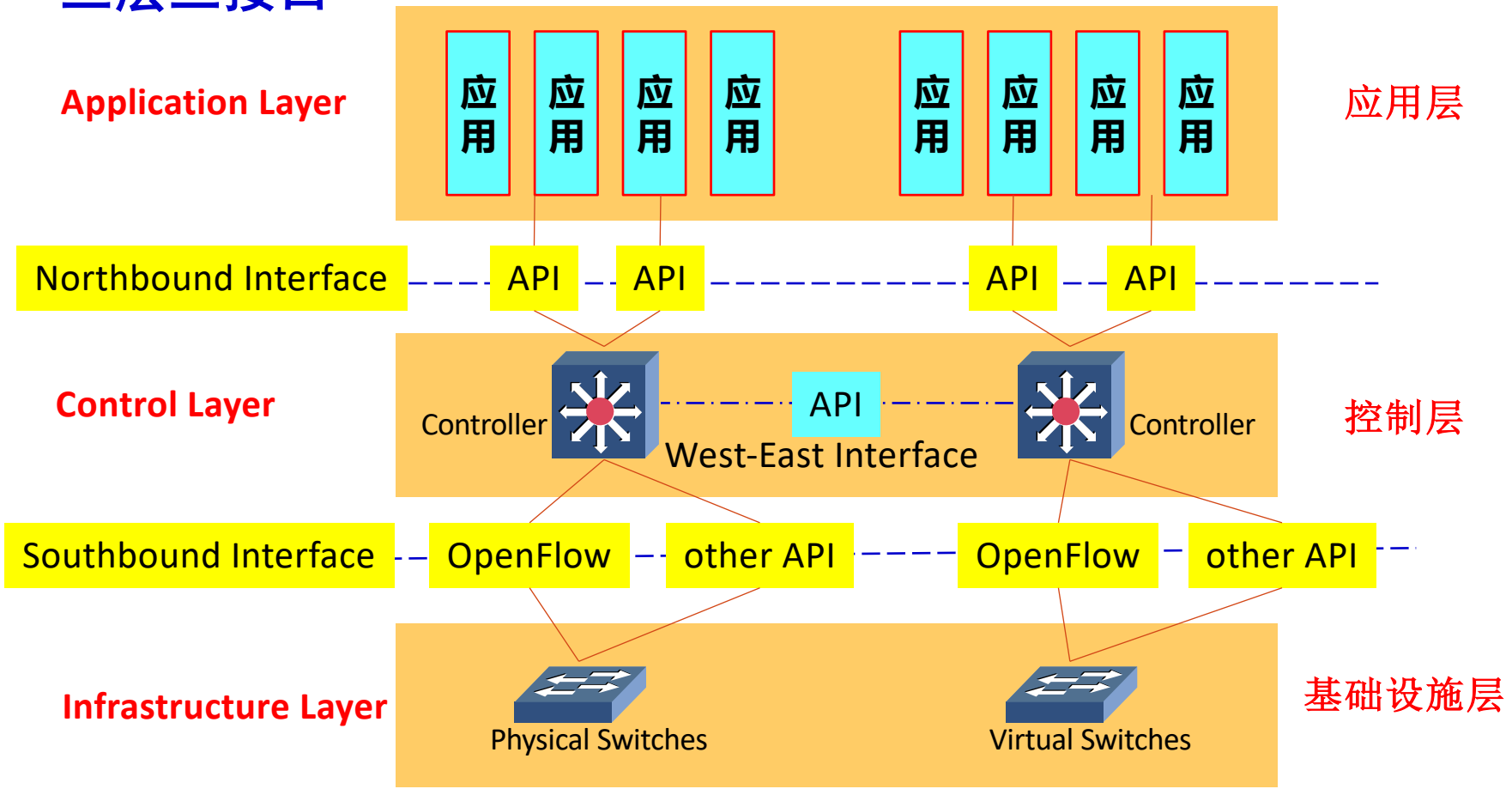
目　录

数据面与控制面耦合的传统网络设备



SDN网络架构

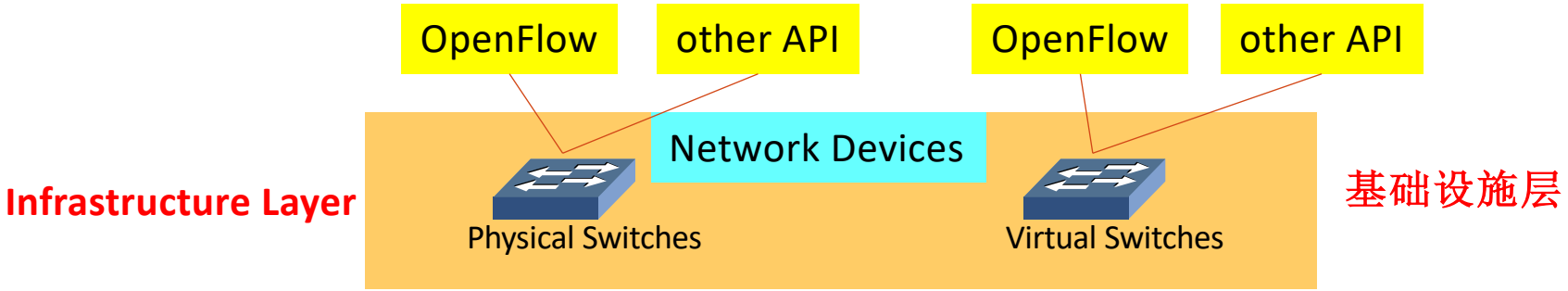
三层三接口



SDN 网络架构-基础设施层

功能：抽象为转发（Forwarding）==转发面/数据面
设备：物理的硬件交换机/虚拟的软件交换机/路由器
任务：存储流表== 根据流表 处理/转发用户报文

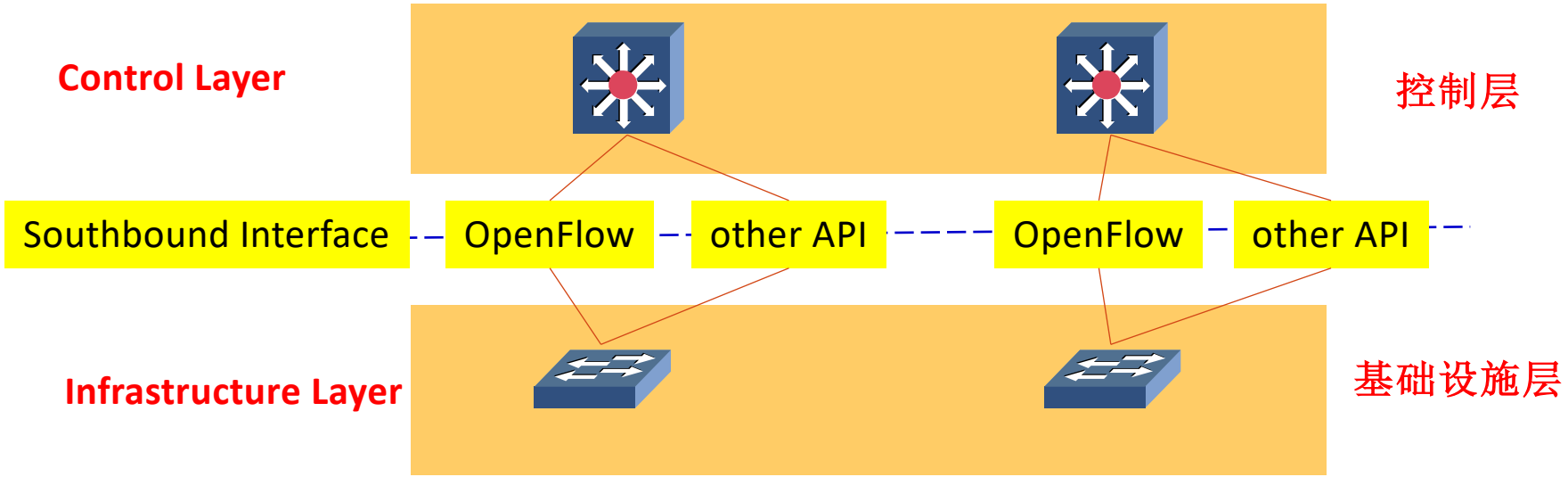
通过南向接口接收Controller 发过来的指令，配置位于交换机内的转发表项
通过南向接口主动上报一些事件给Controller



参考：Software-Defined Networking:The New Norm for Networks， ONF White PaperApril 13, 2012

SDN 网络架构-南向接口

SDN控制器-------→SDN交换机：下发流表
SDN控制器←-------SDN交换机：上报信息
====➔网络的可编程，资源的优化利用，提升网络管控效率
OpenFlow协议
OVS（Open vSwitch，开放虚拟交换标准）

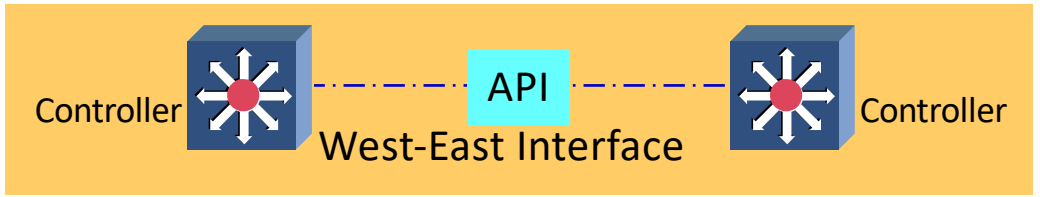


参考：Software-Defined Networking:The New Norm for Networks， ONF White PaperApril 13, 2012

SDN 网络架构-控制层

1个SDN 网络=====可多个Controller
Controller ←==→ Controller　　1主多备；对等关系
1 Controller ==→多台Switch
1 Switch←=====多个Controller
通常Controller 运行在一台独立的服务器上

Control Layer

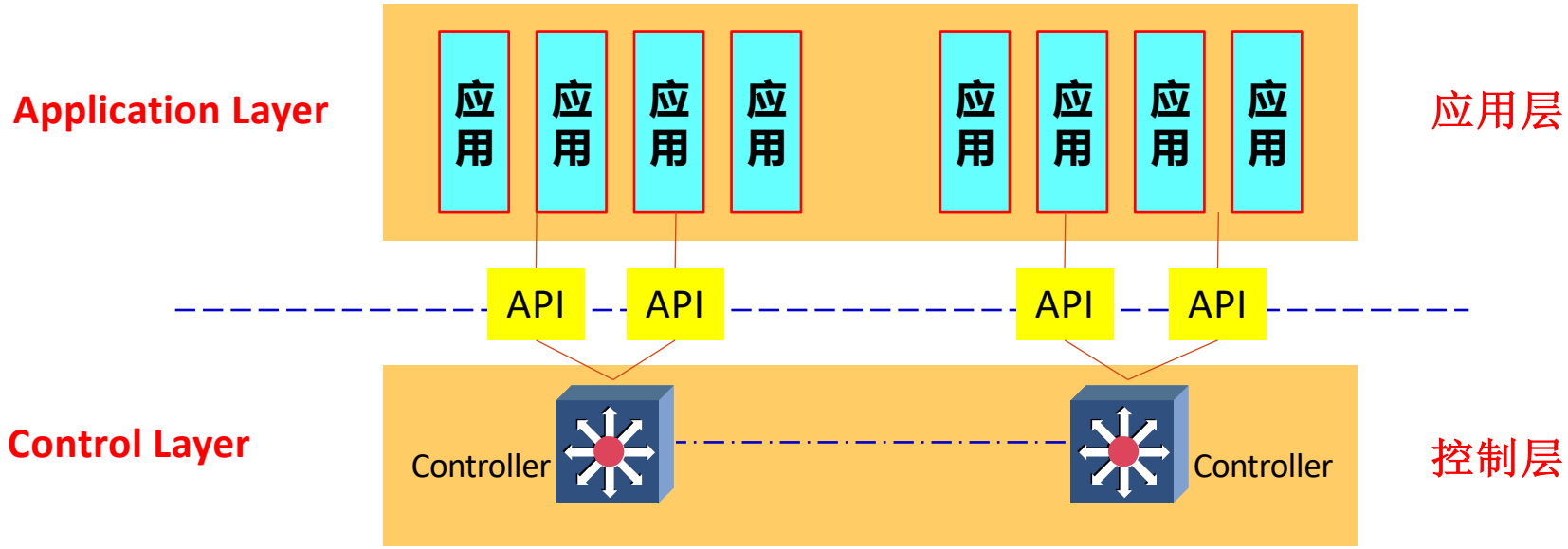


控制层

SDN 网络的核心

参考：Software-Defined Networking:The New Norm for Networks， ONF White PaperApril 13, 2012

SDN 网络架构-北向接口



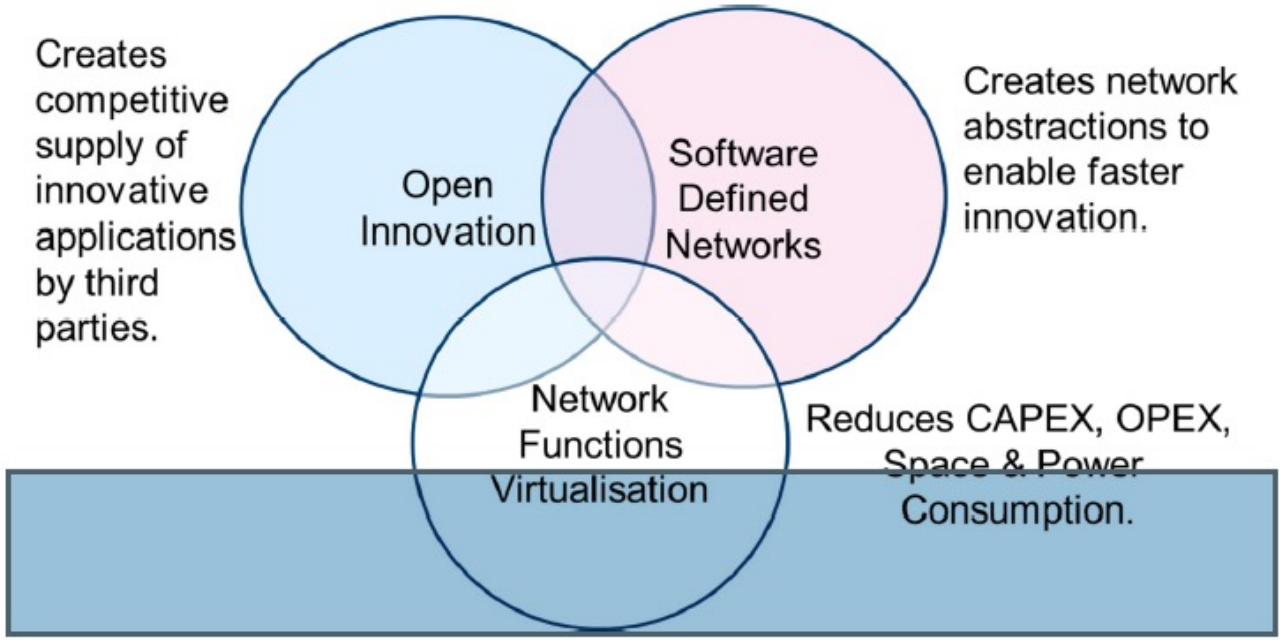
SDN应用的多样性
北向接口的复杂性
REST API 接口
Intent 接口

参考：Software-Defined Networking:The New Norm for Networks， ONF White PaperApril 13, 2012

网络虚拟化（NFV）

- 网络功能虚拟化（Network Function Virtualization，NFV）是一种将传统的专用硬件网络功能（如防火墙、负载均衡、路由器等）虚拟化为软件实例的技术。NFV旨在通过将网络功能从专用硬件中解耦，将其以软件的形式部署在通用服务器上，从而提供更灵活、可扩展和经济高效的网络架构。

NFV与SDN的关系



- NFV与SDN的基础都是通用服务器、云计算以及虚拟化技术
- NFV与SDN存在互补性，二者相互独立，没有依赖关系，SDN不是NFV的前提
- NFV侧重于网络功能软件化，SDN侧重于控制与转发的分离；NFV增加了功能部署的灵活性，SDN可进一步推动NFV功能部署的灵活性和方便性

目　录

SDN编程模型

- SDN controller架构：
  - 负责控制管理网络设备
- 数据面编程：
  - 包括定义流表项、匹配规则和操作指令等
- 控制面编程：
  - 通过编写应用程序或脚本来控制和管理网络设备行为

SDN编程语言和工具

- Python：Ryu与POX controller都是由Python编写
  - Pyretic：基于Python的SDN编程框架
- P4：（Programming Protocol-Independent Packet Processors）是一种可编程数据平面语言，用于定义网络设备（如交换机、路由器）上数据包的处理逻辑。

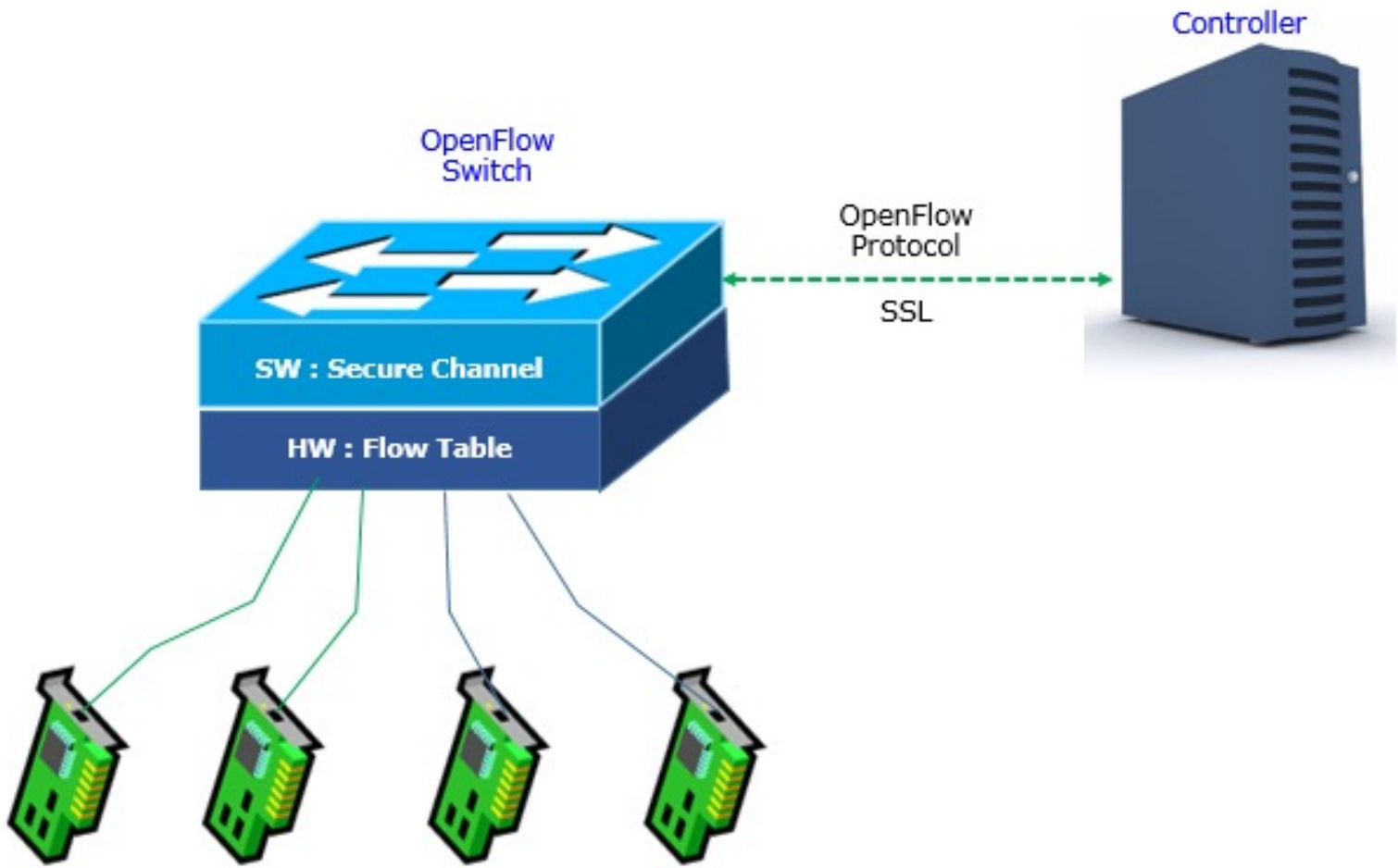
SDN编程语言和工具

- SDN controller：
  - 开源：Ryu，POX，OpenDaylight， ONOS（Open Network Operating System）等
  - 商用：
    - Cisco Application Policy Infrastructure Controller (APIC)
    - VMware NSX
    - Juniper Contrail Controller
    - Huawei CloudEngine SDN Controller
    - Nokia Nuage Networks Virtualized Services Platform (VSP)

OPENFLOW协议

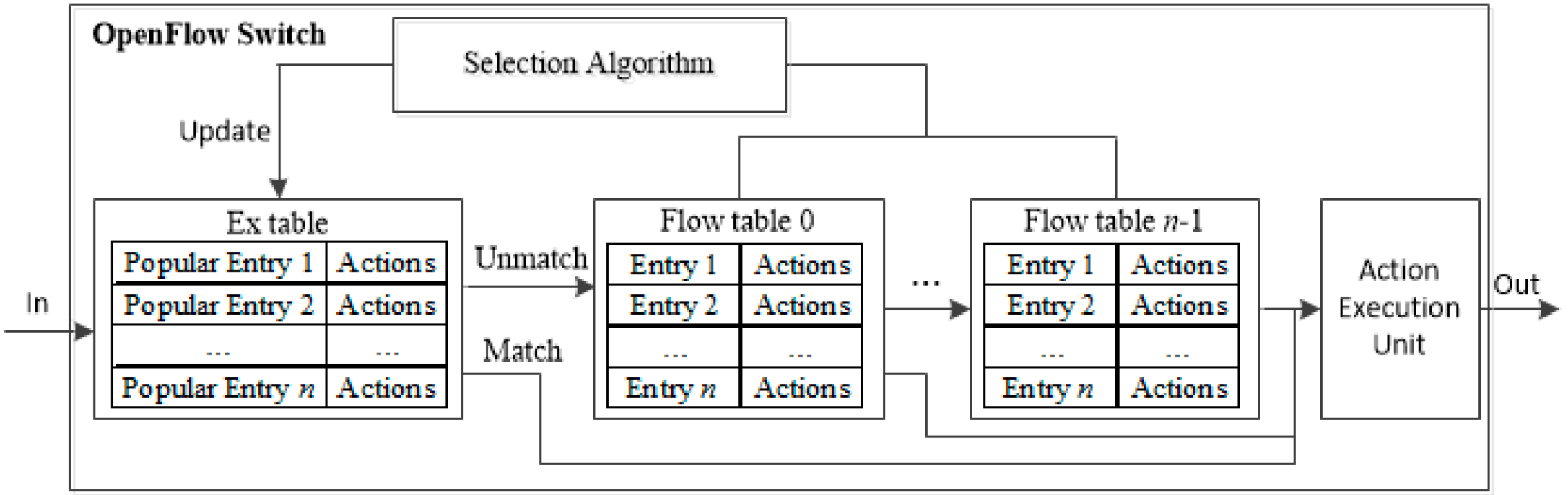
- OpenFlow：一种网络通信协议，属于数据链路层，能够控制网络交换器或路由器的转发平面（forwarding plane），借此改变网络数据包所走的网络路径。
- OpenFlow将转发面抽象为一个由多级流表组成的转发模型，网络控制器通过Openflow协议下发Openflow流表到具体交换机，从而定义、控制交换机的具体行为。

OPENFLOW协议


Controller
OpenFlow
Switch
OpenFlow
Protocol
SSL
SW : Secure Channel
HW : Flow Table

流表（FLOW TABLE）的基本结构

- **流表是OpenFlow对网络设备的数据转发功能的抽象**



表项（FLOW ENTRY）的基本结构

■**流表是OpenFlow对网络设备的数据转发功能的抽象**

–表项（Flow Entry）包括了网络中各个层次的网络配置信息

包头域	计数器	动作

–**包头域**：用于对交换机接收到的数据包的包头内容进行匹配

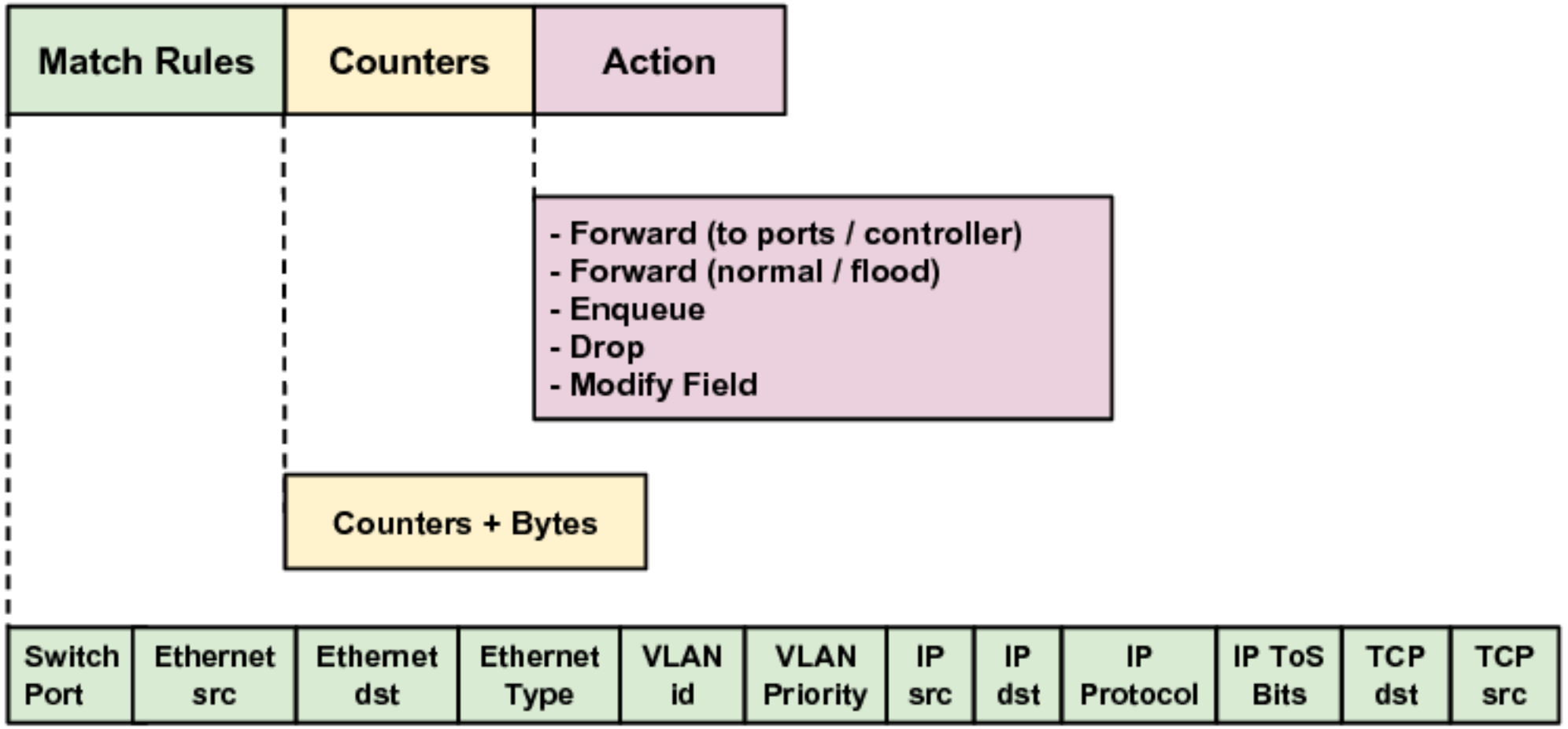
–**计数器**：用于统计数据流量相关信息，可以针对交换机中的每张流表、每个数据流、每个设备端口、每个转发队列进行维护

–**动作（action）**：用于指示交换机在收到匹配数据包后如何对其进行处理

流表包头域（匹配域）

- **用于匹配交换机接收到的数据包的包头内容，OpenFlow 1.1及后续版本将“包头域”更名为“匹配域”**
  - OpenFlow 1.0包头域包含12个元组（tuple）
    - 涵盖ISO网络模型中第二至第四层的网络配置信息
    - 每一个元组中的数值可以是一个确定的值或者是“ANY”

FLOW ENTRY结构

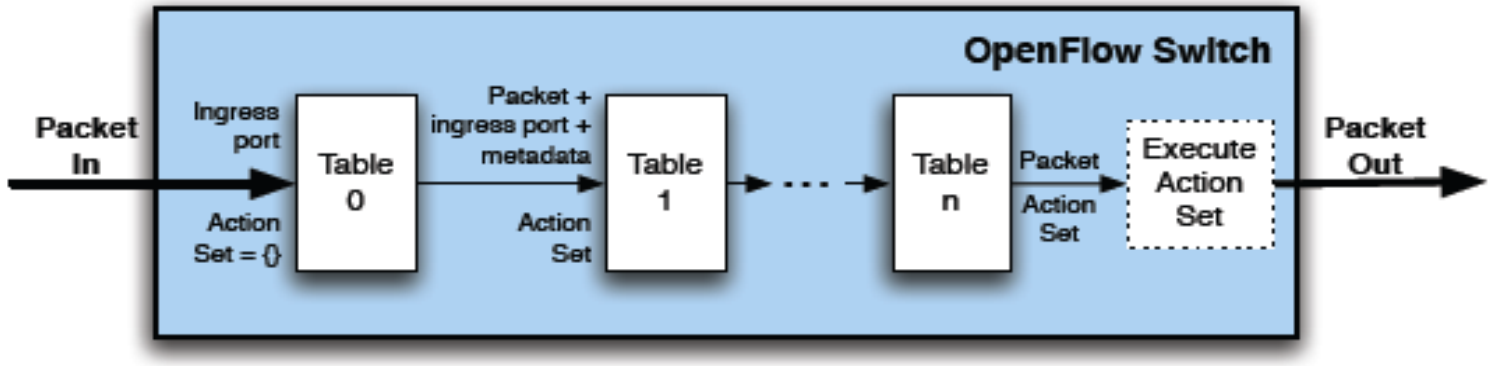


表项（FLOW ENTRY）的动作集

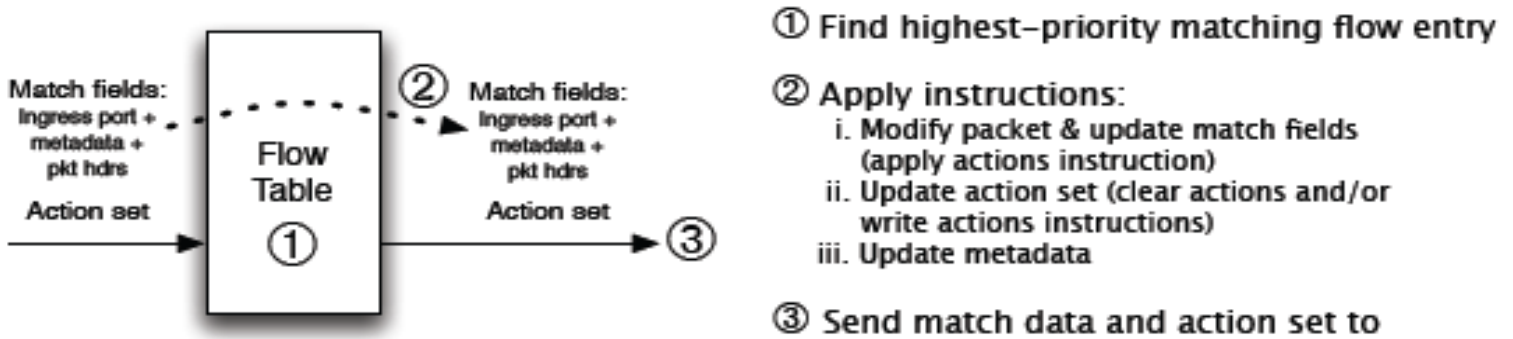
- Flow Entry的Action可以包括转发数据包到指定端口、修改数据包的头部信息、设置数据包的QoS优先级等。可以根据需要指定多个操作，交换机将按顺序依次执行这些操作。
- Flow Entry的Action通常具有更高的优先级，它覆盖了Flow Table的Action。即如果数据包匹配到了具体的Flow Entry，则会执行Flow Entry的Action，而不会执行Flow Table的Action。

SDN转发流程

PIPELINE
的处理



(a) Packets are matched against multiple tables in the pipeline

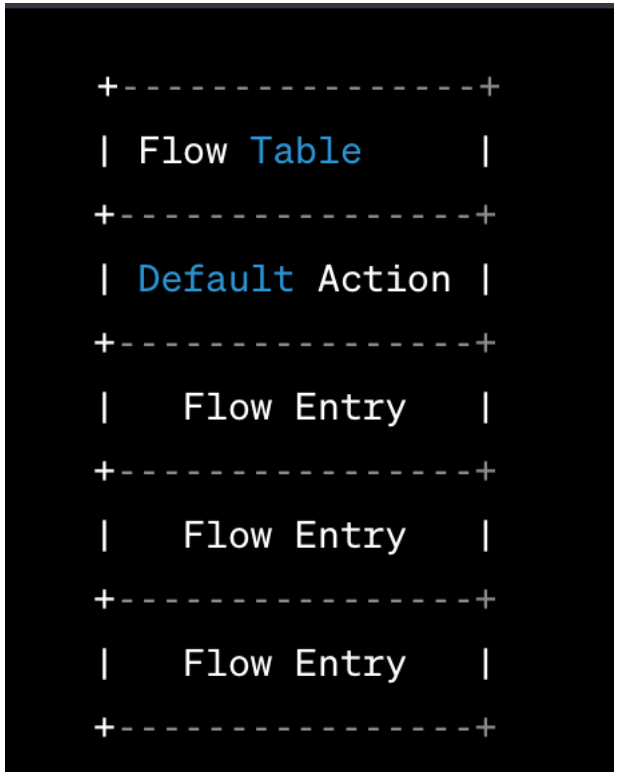


(b) Per-table packet processing

- 每个openflow switch的pipeline包含多个flowtable，每个flowtable包含多个flowentry
- 每个flowentry的处理结果只能交给序号更大的表进一步处理

流表（FLOW TABLE）的动作

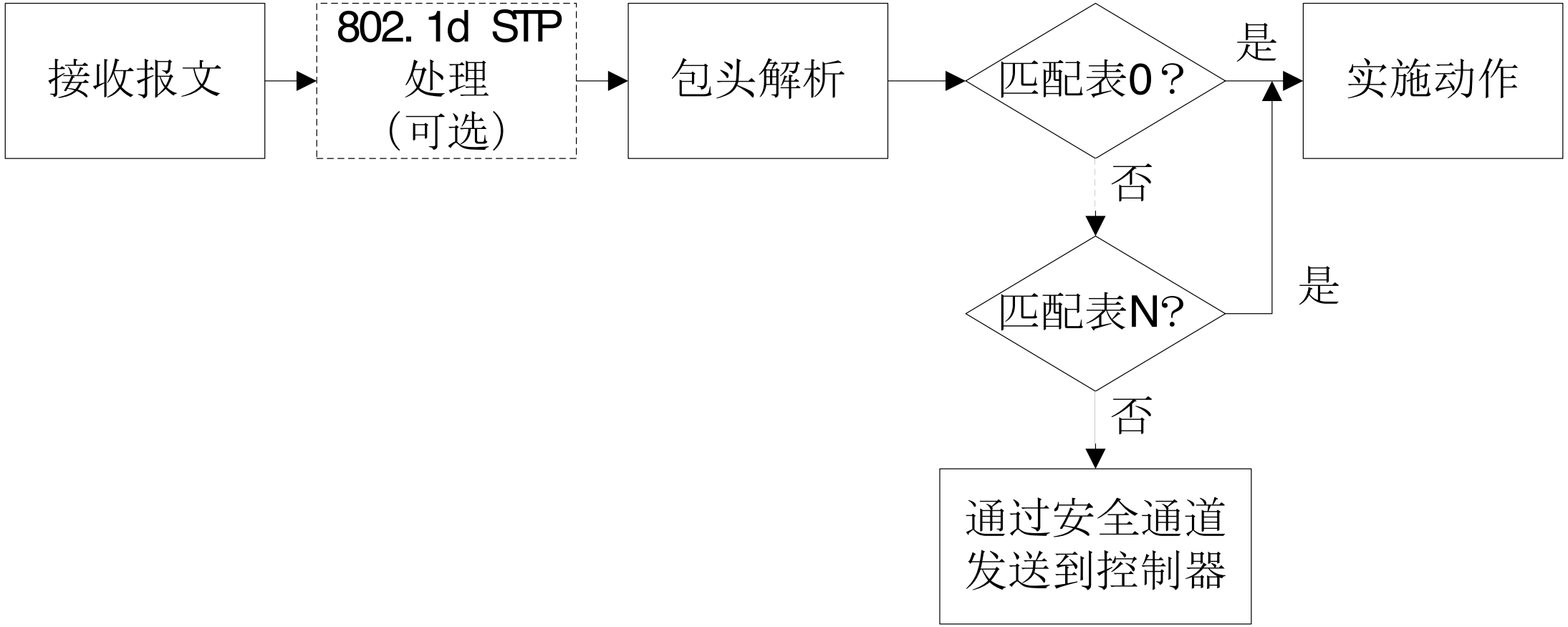
- Flow Table中的动作为默认动作，即数据包未匹配到Flow Entry时采取的动作。
- 因此，Flow Entry的动作优先级更高。



流表动作列表

类型	名称	说明
必备动作	转发 (Forward)	交换机必须支持将数据包转发给设备的物理端口及如下的一个或多个虚拟端口： ➢ALL：转发给所有出端口，但不包括入端口 ➢CONTROLLER：封装数据包并转发给控制器 ➢LOCAL：转发给本地的网络栈 ➢TABLE：对packet_out消息执行流表的动作 ➢IN_PORT：从入端口发出
	丢弃 (Drop)	交换机对没有明确指明处理动作的流表项，将会对与其所匹配的所有数据包进行默认的丢弃处理
可选动作	转发 (Forward)	交换机可选支持将数据包转发给如下的虚拟端口： ➢NORMAL：利用交换机所能支持的传统转发机制（例如二层的MAC、VLAN信息或者三层的IP信息）处理数据包 ➢FLOOD：遵照最小生成树从设备出端口洪泛发出，但不包括入端口
	排队 (Enqueue)	交换机将数据包转发到某个出端口对应的转发队列中，便于提供QOS支持
	修改域 (Modify-Field)	交换机修改数据包的包头内容，具体可以包括： ➢设置VLAN ID、VLAN优先级，剥离VLAN头 ➢修改源MAC地址、目的MAC地址 ➢修改源IPv4地址、目的IPv4地址、ToS位 ➢修改源TCP/IP端口、目的TCP/IP端口

OPENFLOW交换机数据包处理流程



安全通道

- **OpenFlow的集中控制架构对控制器与交换机之间信息传送通道提出了极高的要求**
  - 控制器与交换机之间的数据通路必须确保安全
    - 采用TLS（Transport Layer Security）技术

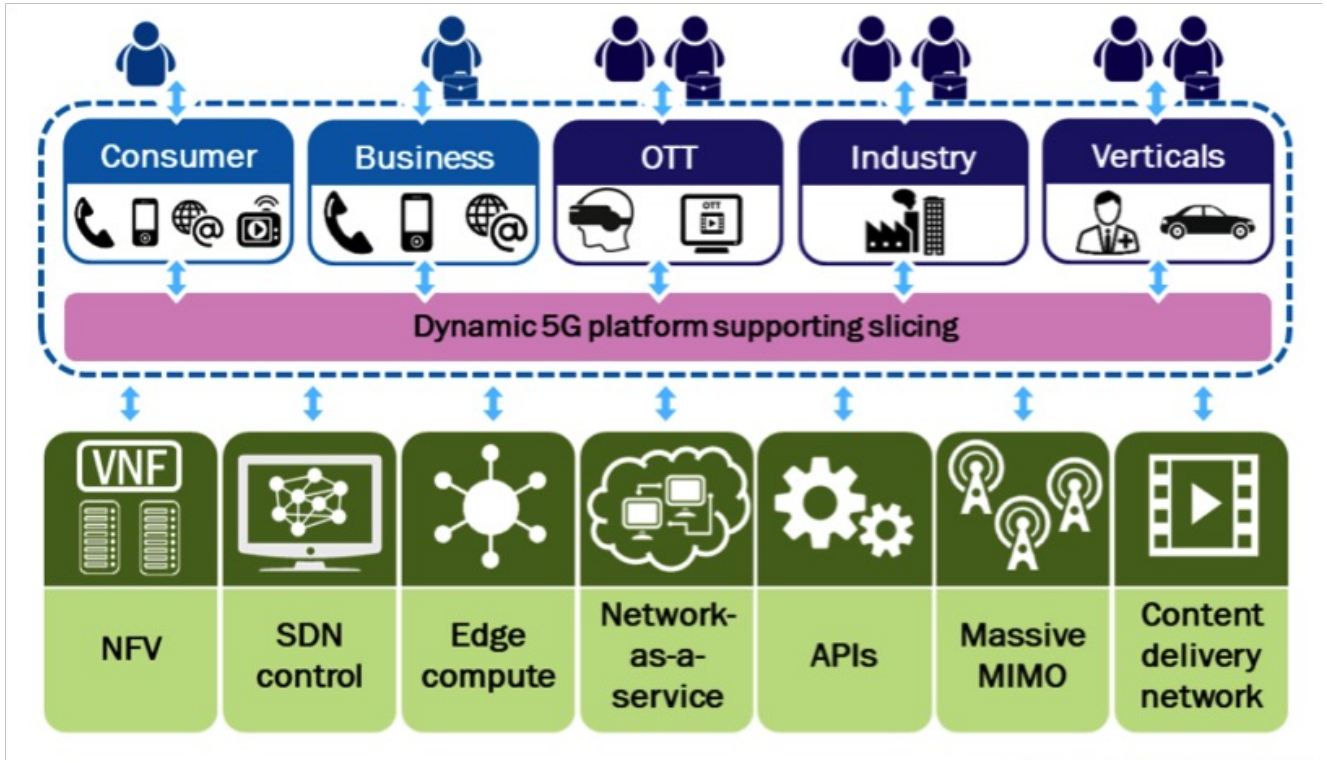
目　录

SDN应用


互联网/电信运营商
Google
Microsoft
Tencent 腾讯
verizon
T··Mobile·
德国电信
部分主流互联网运营商已成功商用；电信运营商开始进行试验。
解决方案/设备提供商
nicira
NICIRA
I OpenFlow
Bigswitch
CISCO
Juniper NETWORKS
NEC
hp invent
IBM
ERICSSON
HUAWEI
ZTE中兴
主流IT厂家与创新公司产品已成功商用；传统主流IP厂家态度由保守向积极转变。
芯片提供商
intel
BROADCOM.
MARVELL
主流芯片厂商态度积极，已推出商用产品，规模生产指日可待。

SDN应用

- 软件定义数据中心（SDDC）
- 软件定义广域网（SD-WAN）
- 5G和物联网



SDN未来发展趋势

- 边缘计算：SDN可以与边缘计算相结合，实现边缘网络的智能管理和资源协调，提供低延迟的计算和网络服务。
- 安全和隐私保护：SDN可以用于网络安全和隐私保护，通过集中管理和动态策略控制，提供更强大的安全性和隐私保护机制。
- 人工智能和机器学习：SDN与人工智能（AI）和机器学习（ML）相结合，可以实现智能的网络管理和优化，自动识别和适应不断变化的网络环境。

Thank You