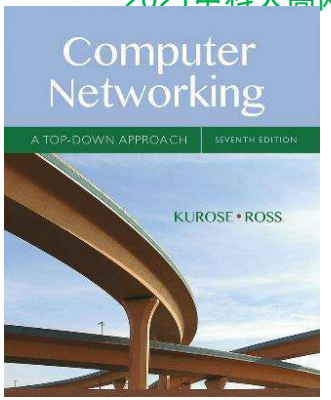
SDN：软件定义网络

中国科学技术大学
自动化系 郑烃
改编自Jim kurose, Keith Ross



Computer Networking: A Top Down Approach

7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

提纲

1. SDN导论
2. SDN数据平面
   - match
   - action
   - OpenFlow examples of match-plus-action in action
3. SDN控制平面

网络层功能：数据平面和控制平面

网络层功能：

- *转发：* 对于从某个端口到来的分组转发到合适的输出端口
- *路由：* determine 决定分组从源端到目标端的路径
  - *路由算法*

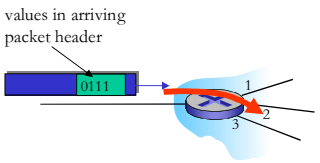
类比：旅行

- *转发：* 一个多岔路口的进入和转出过程
- *路由：* 规划从源到目标的旅行路径

网络层：数据平面和控制平面

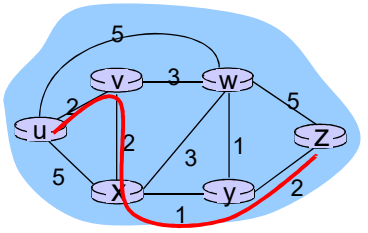
数据平面

- 本地的、每个路由器的功能
- 决定某个从某个端口进入的分组从从哪个端口输出
- 转发功能



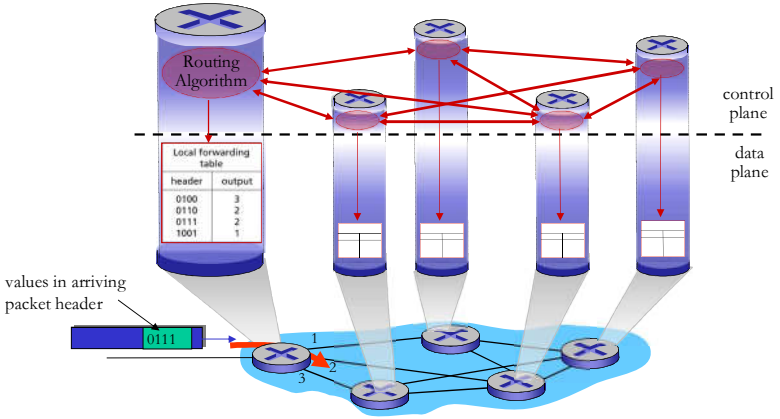
控制平面

- 网络范围的逻辑
- 决定分组端到端穿行于各个路由器的路径



传统路由器的控制平面

- 控制平面功能和数据平面功能垂直集成在每个路由器上
- 每个路由器都部分实现了路由功能，控制平面功能分布式实现
- 转发表：控制平面和数据平面功能的结合剂



其他控制功能分布式实现的网络设备

- 传统方式下实现的路由器网络层功能
  - IP转发：对于到来的分组按照路由表决定如何转发，数据平面
  - 路由：决定路径，计算**路由表**；处在控制平面
- 其他网络设备（中间盒）：种类繁多
  - 交换机；防火墙；NAT；IDS；负载均衡设备
  - 不断增加的需求和相应的网络设备，需要不同的设备去实现不同的网络功能
    - 每台设备集成了控制平面和数据平面的功能
    - 控制平面分布式地实现了各种控制平面功能
    - 升级（改变它们的工作方式）和部署新网络设备非常困难

网络设备控制平面的分布式实现方式特点

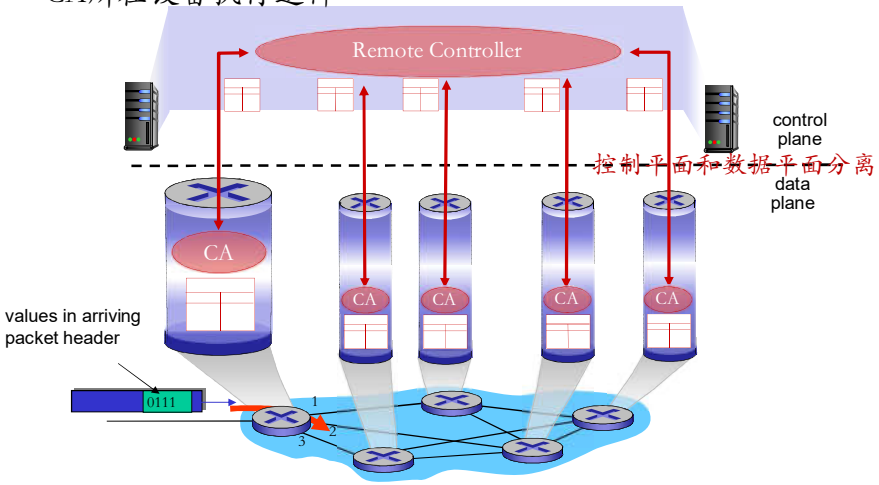
- 传统方式下的网络设备：分布式、每个设备的方法来实现（数据平面和控制平面）
  - *垂直集成：每个*路由器等网络设备包含：
    - 1）硬件、在私有的操作系统
    - 2）运行互联网标准协议(IP, RIP, IS-IS, OSPF, BGP)私有实现
    - 3）全部由一个厂商提供
  - 每个设备实现了数据平面和控制平面的事情，控制平面的功能是分布式实现的
  - 设备按照固定方式工作，控制逻辑固化
  - 不同的网络层功能需要不同的“middleboxes”：防火墙、负载均衡设备、NAT boxes, ..
- 集成->分布->固化

传统方式实现网络功能的问题

- 垂直集成->昂贵、不便于创新的生态
- 分布式、固化
  - > 固化：无法改变路由等工作逻辑，无法实现流量工程等高级特性
  - > 设备种类繁多：要增加新的网络功能，需要设计、实现以及部署新的特定设备
  - > 管理困难：配置错误影响全网运行，升级和维护会涉及到全网设备
- ~2005: 开始重新思考网络控制平面的处理方式
  - 集中：远程的控制器集中实现控制逻辑，好管理，升级，可编程
  - 远程：数据平面和控制平面的分离，不再和数据平面功能集成在一台设备上

SDN方式：逻辑上集中的控制平面

- 一个远程的控制器与网络设备相独立，与设备上控制代理交互
- 控制器决定分组转发逻辑（可编程），下发流表给CA
- CA所在设备执行逻辑



SDN的主要思路

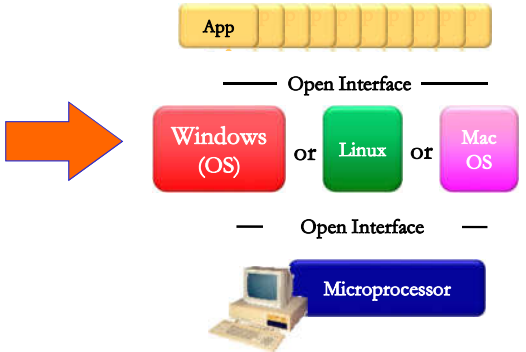
- 网络设备数据平面和控制平面分离
- 数据平面-分组交换机
  - 将路由器、交换机和目前大多数网络设备的功能进一步**抽象**成：按照流表（由控制平面设置的控制逻辑）进行PDU（帧、分组）的动作（包括转发、丢弃、拷贝、泛洪、阻塞）
  - **统一化**设备功能：SDN交换机（分组交换机），执行控制逻辑
- 控制平面-控制器+网络应用
  - 分离、集中
  - 计算和下发控制逻辑：流表

SDN控制平面和数据平面分离的优势

- 水平集成控制平面的开放实现（而非私有实现），创造出好的产业生态，促进发展
  - 分组交换机、控制器和各种控制逻辑网络应用app可由不同厂商生产，专业化，引入竞争形成良好生态
- 集中式实现控制逻辑，网络管理容易：
  - 集中式控制器了解网络状况，编程简单，传统方式困难
  - 避免路由器的误配置带来的影响
- 基于流表的**匹配+行动**的工作方式允许“可编程的”分组交换机
  - 实现流量工程等高级特性
  - 在此框架下实现各种新型（未来）的网络设备
- 运营商一旦部署完毕不必要再增加新的网络设备

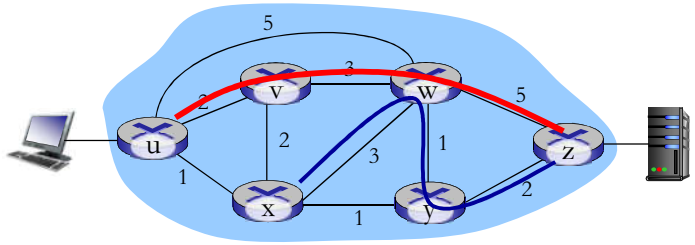
类比：主框架到PC的演变

垂直集成，昂贵
封闭，私有
创新缓慢
产业规模小



水平集成、竞争、便宜
开放接口
快速创新
产业巨大

流量工程: 传统路由器比较困难

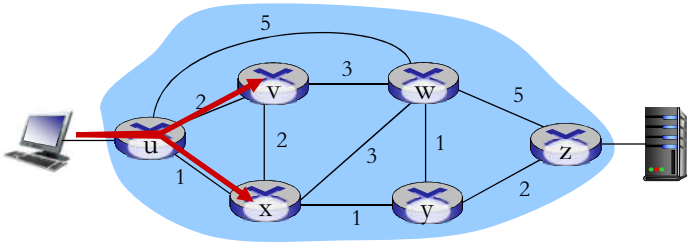


Q: 网管如果需要u到z的流量走uvwz,x到z的流量走xwyz，怎么办?

A: 需要定义链路的代价，流量路由算法以此运算（IP路由面向目标，无法操作）(或者需要新的路由算法)!

2021中科大高网

流量工程：困难

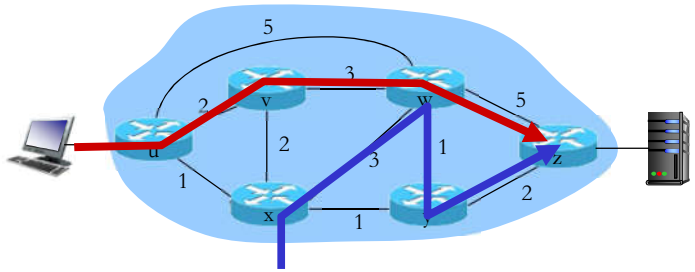


Q: 如果网管需要将u到z的流量分成2路：uvwz 和uxyz（负载均衡)，怎么办?（IP路由面向目标）

A: 无法完成(在原有体系下只有使用新的路由选择算法，而在全网部署新的路由算法是个大的事情，固化)

2021中科大高网

流量工程：困难

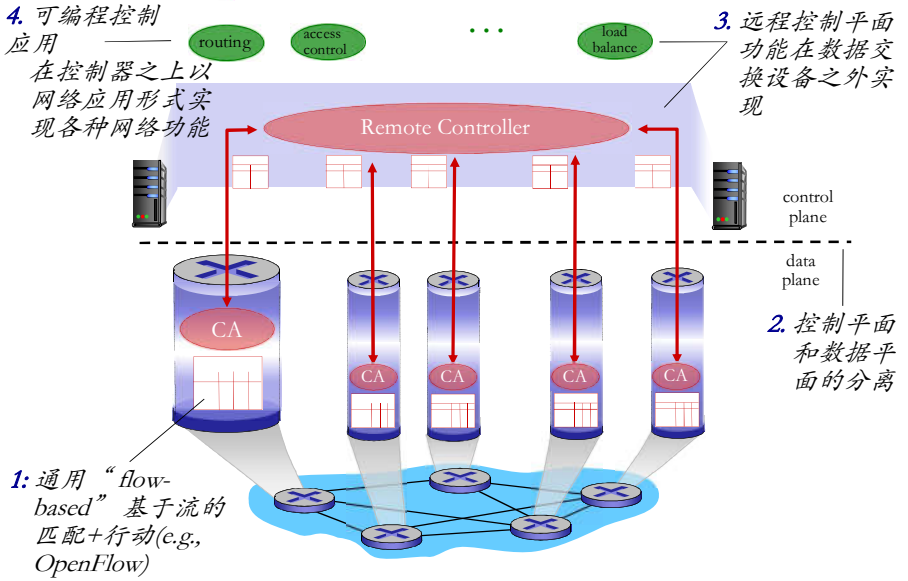


Q: 如果需要w对蓝色的和红色的流量采用不同的路由，怎么办?

A: 无法操作 (基于目标的转发，采用LS, DV 路由)

2021中科大高网

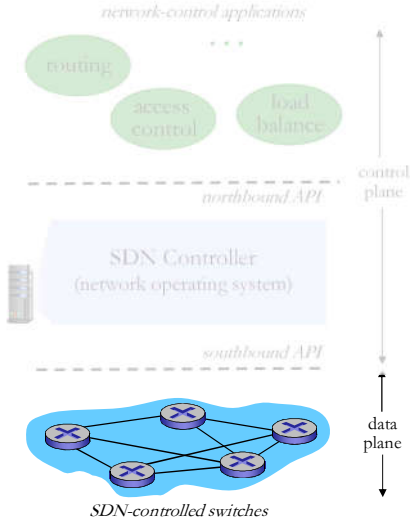
SDN特性



SDN 架构: 数据平面交换机

数据平面交换机

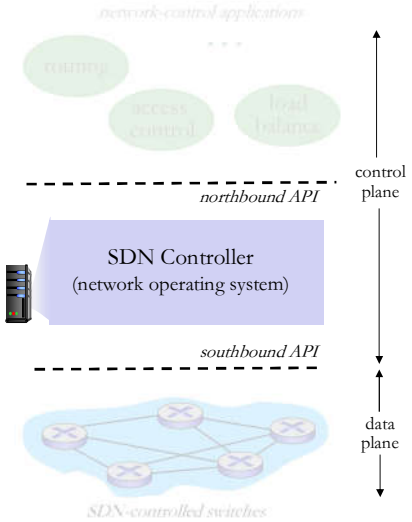
- 高速、简单、商业化交换设备采用硬件实现通用转发功能
- 流表被控制器计算和安装
- 分组交换机 和 控制器之间符合 openflow 南向接口协议
  - 上报sdn交换机状态
  - 接收来自控制的流表修改。



SDN 架构: SDN控制器

SDN 控制器(网络OS):

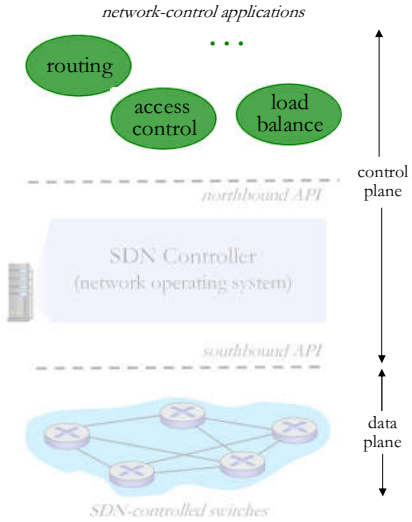
- 维护网络状态信息
- 通过上面的北向API和网络控制应用交互
- 通过下面的南向API和网络交换机交互
- 逻辑上集中，但是在实现上通常由于性能、可扩展性、容错性以及鲁棒性采用分布式方法



SDN 架构: 控制应用

网络控制应用:

- 控制的大脑：基于下层提供的服务（SDN控制器提供的API)，实现各种网络功能
  - 路由器 交换机
  - 接入控制 防火墙
  - 负载均衡
  - 未来新功能
- *非绑定:*
  - 可以由第三方提供
  - 与控制器厂商以通常上不同，
  - 与分组交换机厂商也可以不同

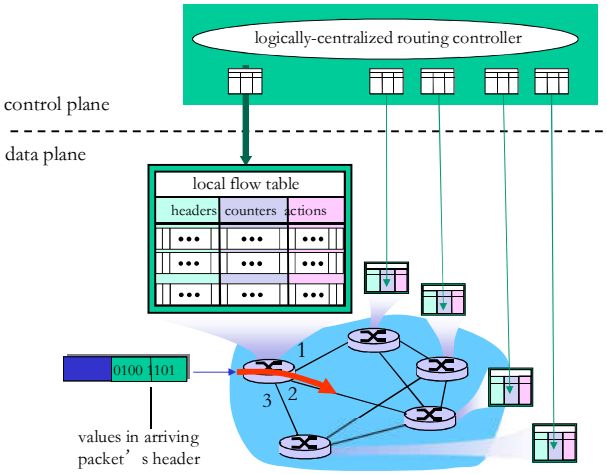


提纲

1. SDN导论
2. SDN数据平面
   - match
   - action
   - OpenFlow examples of match-plus-action in action
3. SDN控制平面

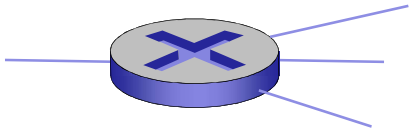
通用转发和SDN

每个路由器包含一个流表（被逻辑上集中的控制器计算和分发）



OpenFlow 数据平面抽象

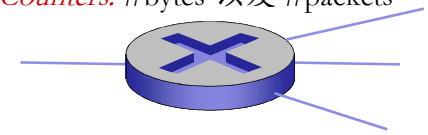
- *流*: 由分组（帧）头部字段所定义
- 通用转发: 简单的分组处理规则
  - *模式:* 将分组头部字段和流表进行匹配
  - *行动：对于匹配上的分组，可以是丢弃、转发、修改、将匹配的分组发送给控制器*
  - *优先权Priority*: 几个模式匹配了，优先采用哪个，消除歧义
  - *计数器Counters:* #bytes 以及 #packets



路由器中的流表定义了路由器的匹配+行动规则（流表由控制器计算并下发）

OpenFlow 数据平面抽象

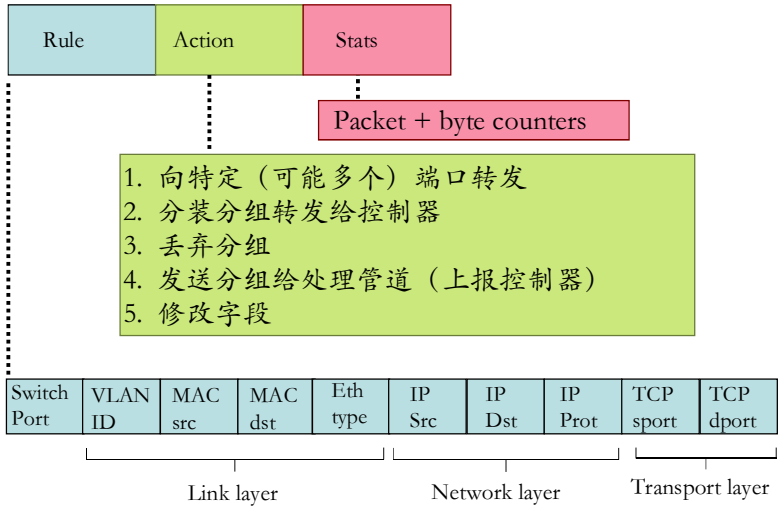
- *流*: 由头部字段所定义
- 通用转发: 简单的分组处理规则
  - *模式Pattern:* 将分组头部字段和流表进行匹配
  - *行动：对于匹配上的分组，可以是丢弃、转发、修改、将匹配的分组发送给控制器*
  - *优先权Priority*: 几个模式匹配了，优先采用哪个，消除歧义
  - *计数器Counters:* #bytes 以及 #packets



* : wildcard

1. src=1.2.*.*, dest=3.4.5.* → drop
2. src = *.*.*.*, dest=3.4.*.* → forward(2)
3. src=10.1.2.3, dest=*.*.*.* → send to controller

OpenFlow: 流表的表项结构



例子

基于目标的转发

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	51.6.0.8	*	*	*	port6

IP 数据报目标地址是51.6.0.8
应该被通过端口6转发

防火墙:

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

不转发（阻塞）所有具有目标TCP端口号是22的分组

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

所有由128.119.1.1发送的分组都应该被阻塞

例子

基于层2目标的转发:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	22:A7:23:11:E1:02	*	*	*	*	*	*	*	*	port3

所有层2源MAC地址是 22:A7:23:11:E1:02都应该被
向端口6转发

OpenFlow 抽象

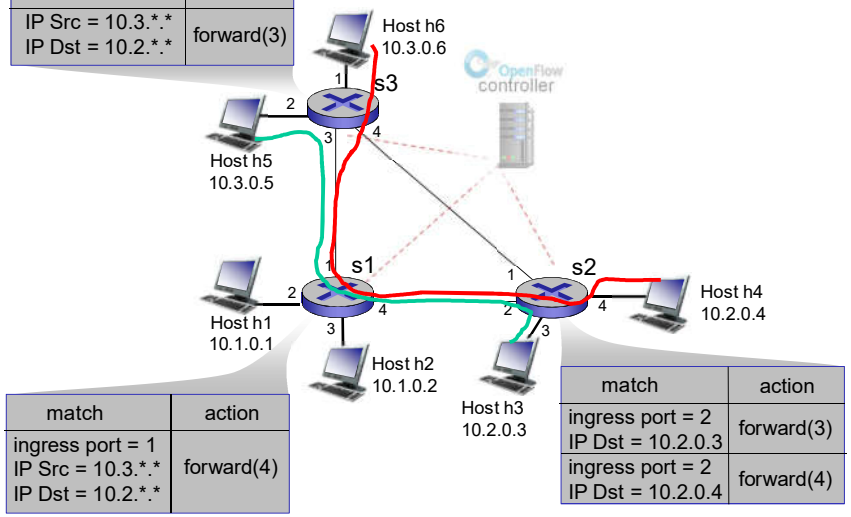
- *match+action:* 统一化各种网络设备的功能
- 路由器
  - *match:* 最长前缀匹配
  - *action:* 通过一条链路转发
- 交换机
  - *match:* 目标MAC地址
  - *action:* 转发或者泛洪
- 防火墙
  - *match:* IP地址和TCP/UDP端口号
  - *action:* 允许或者禁止
- NAT
  - *match:* IP地址和端口号
  - *action:* 重写地址和端口号

目前几乎所有的网络设备都可以在这个 匹配+行动模式 框架进行描述，具体化为各种网络设备包括未来的网络设备

OpenFlow 例子

例子: 来自H5和H6的数据报应该被发向H3或者H4通过 s1然后经由 s2

match	action
IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(3)



match	action
ingress port = 1 IP Src = 10.3.*.* IP Dst = 10.2.*.*	forward(4)

match	action
ingress port = 2 IP Dst = 10.2.0.3	forward(3)
ingress port = 2 IP Dst = 10.2.0.4	forward(4)

提纲

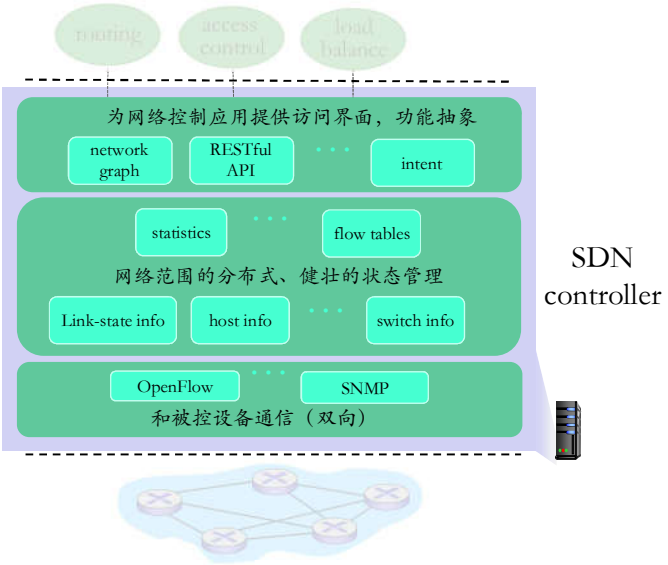
1. SDN导论
2. SDN数据平面
   - match
   - action
   - OpenFlow examples of match-plus-action in action
3. SDN控制平面

SDN控制器里的元件

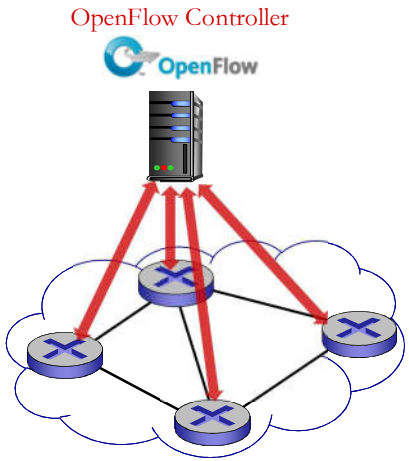
网络控制应用的**界面层** 抽象API

网络范围的**状态管理层**: 网络链路、交互设备和服务的状态: *分布式数据库*

通信层: SDN控制器和SDN交换机之间进行通信



OpenFlow 协议

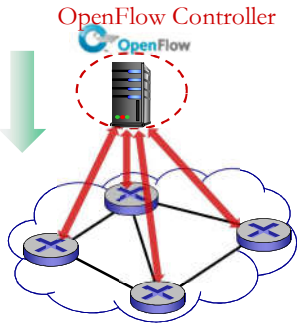


- 控制器和SDN交换机交互的协议
- 采用TCP 来交换报文
  - 加密可选
- 3种OpenFlow报文类型
  - 控制器>交换机
  - 异步（交换机>控制器）
  - 对称 (misc)

OpenFlow: 控制器->交换机报文

一些关键的控制器到交换机报文

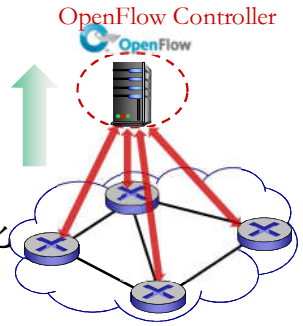
- *特性：* 控制器查询交换机特性，交换机应答
- *配置：* 交换机查询/设置交换机的配置参数
- *修改状态：* 增加删除修改OpenFlow表中的流表
- *packet-out：* 控制器可以将分组通过特定的端口发出



OpenFlow: 交换机到控制器的报文

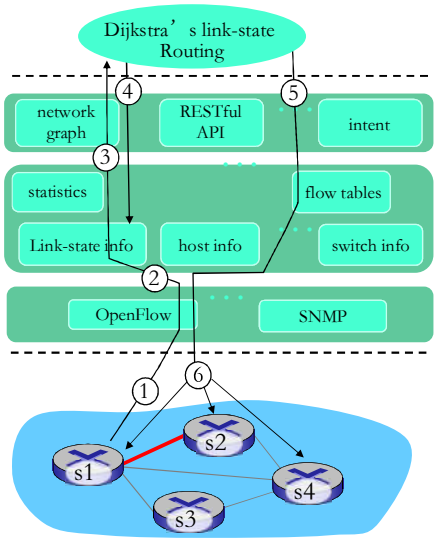
一些关键的交换机到控制器报文

- *分组进入:* 将分组（和它的控制）传给控制器，见来自控制器的packet-out报文
- *流移除:* 在交换机上删除流表项
- *端口状态:* 通告控制器端口的变化



幸运的是，网络管理员不需要直接通过创建/发送流表来编程交换机，而是采用在控制器上的app自动运算和配置

SDN: 控制/数据平面交互的例子



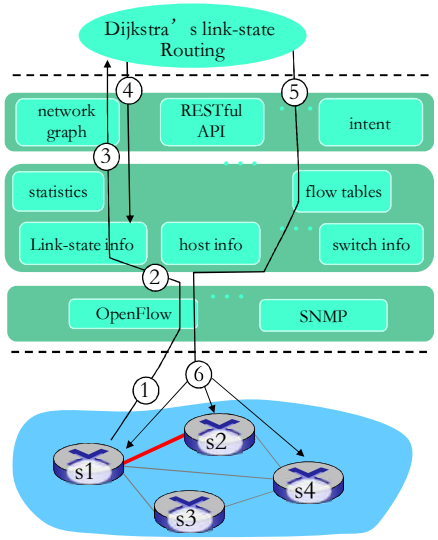
① S1, 经历了链路失效，采用OpenFlow报文通告控制器:
端口状态报文

② SDN 控制器接收OpenFlow报文，更新链路状态信息

③ Dijkstra路由算法应用被调用（前面注册过这个状态变化消息）

④ Dijkstra路由算法访问控制器中的网络拓扑信息，链路状态信息计算新路由

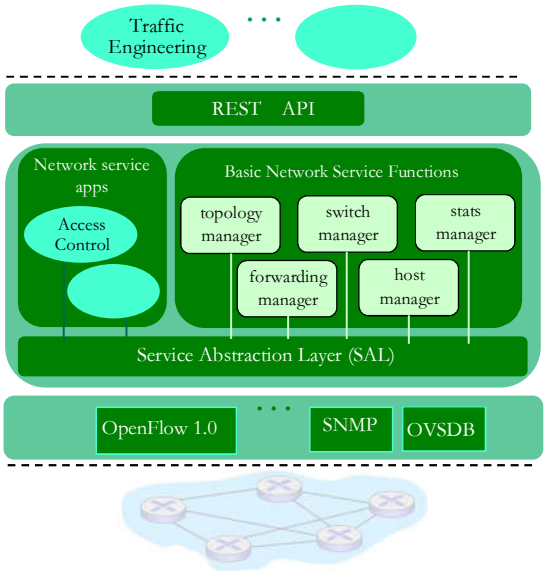
SDN: 控制/数据平面交互的例子



⑤ 链路状态路由app和SDN控制器中流表计算元件交互，计算出新的所需流表

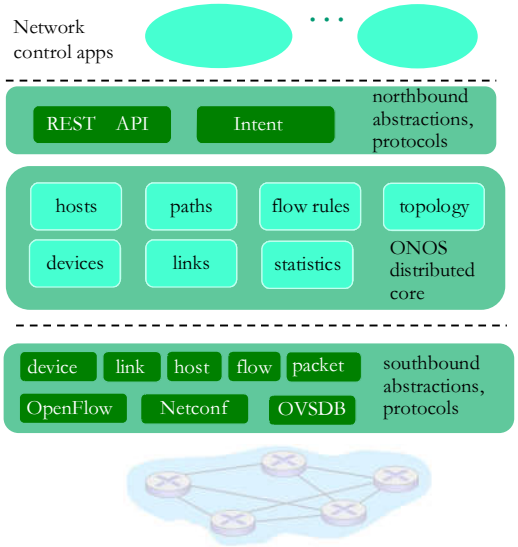
⑥ 控制器采用OpenFlow在交换机上安装新的需要更新的流表

OpenDaylight (ODL) 控制器



- ODL Lithium 控制器
- 网络应用可以在SDN控制内或者外面
- 服务抽象层SAL: 和内部以及外部的应用以及服务进行交互

ONOS 控制器



- 控制应用和控制器分离（应用app在控制器外部）
- 意图框架：服务的高级规范：描述什么而不是如何
- 相当多的重点聚焦在分布式核心上，以提高服务的可靠性，性能的可扩展性

2021中科大高网

SDN: 面临的挑战

- 强化控制平面：可信、可靠、性能可扩展性、安全的分布式系统
  - 对于失效的鲁棒性：利用为控制平面可靠分布式系统的强大理论
  - 可信任，安全：从开始就进行铸造
- 网络、协议满足特殊任务的需求
  - e.g., 实时性、超高可靠性、超高安全性
- 互联网络范围内的扩展性
  - 而不是仅仅在一个AS的内部部署，全网部署