



# 《软件定义网络技术》

## Software Defined Networking (SDN)

主讲人: 黄辉  
email: huanghui@wzu.edu.cn

# SDN的核心思想

- 解耦（Decoupling）；
- 抽象（Abstraction）；
- 可编程（Programmable）

# 数据平面与控制平面的解耦

- 通过解耦合，控制平面负责上层的控制决策，数据平面负责数据的交换转发，双方遵循一定的开放接口进行通信；
- 实现网络逻辑集中控制的前提；
- 两个平面独立完成体系结构和技术的发展演进，有利于网络的技术创新与发展。

# 控制与转发分离

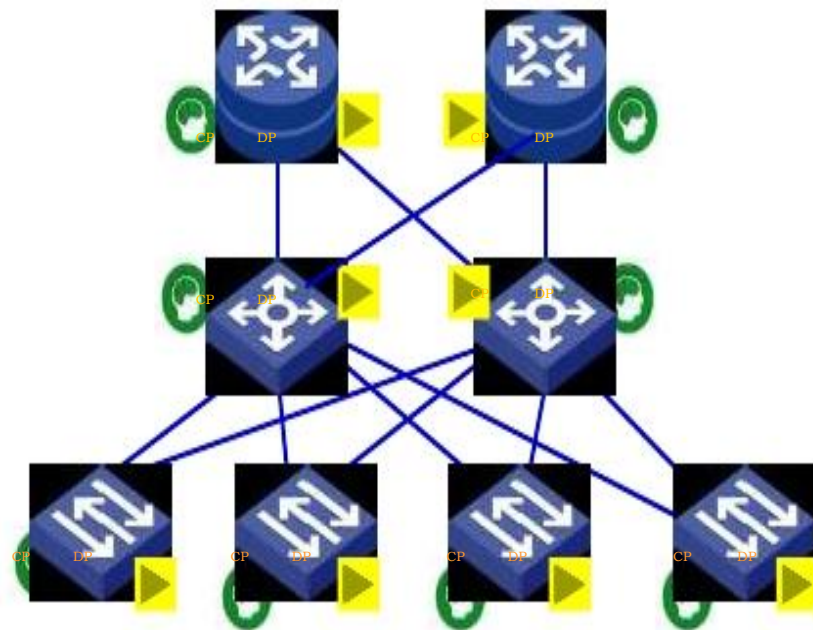


控制平面 (CP)



转发平面 (DP)

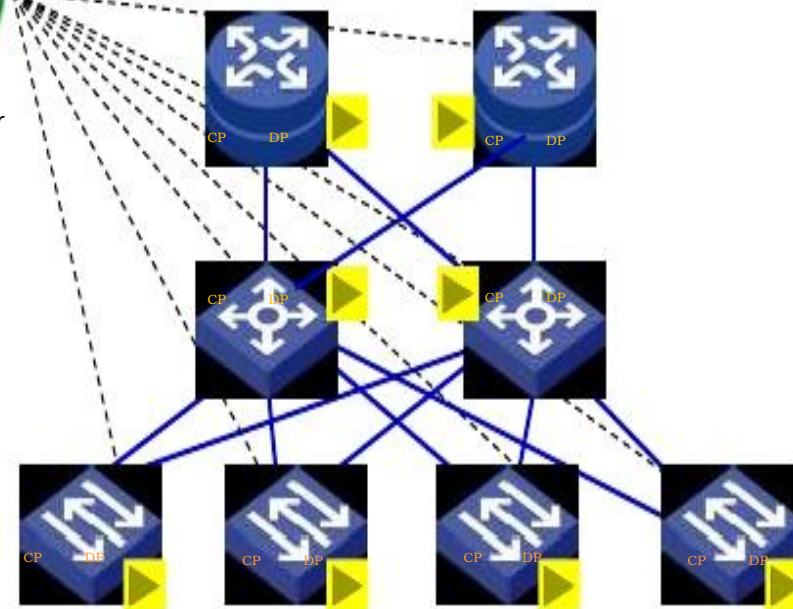
- 传统网络设备的CP与DP 不分离；
- 设备之间通过控制协议交互转发信息；



- SDN 将网络设备的控制平面集中上收到Controller;
- 网络设备上只保留转发平面（转发表项）；
- 软件可以实现灵活的控制面功能满足用户多元化需求
- 硬件成为简单哑资源，专注转发；



Controller



# 解耦带来的问题与挑战

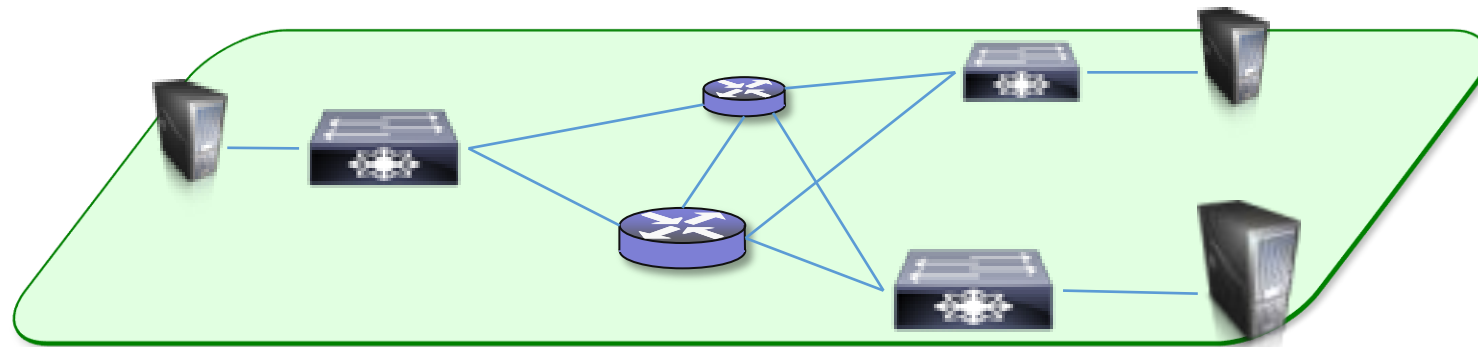
- 网络规模的扩大，单一控制器成为网络性能的瓶颈；
- 保持分布式网络节点状态的一致性，是一个重要的挑战；
- 响应延迟，导致数据平面的可用性问题。

# 网络功能的抽象

- ONF网络架构实现转发抽象、分布状态抽象和配置抽象：
  - 转发抽象（forwarding abstraction）：隐藏了底层的硬件实现，转发行为与硬件无关；
  - 分布状态抽象（distribution abstraction）：屏蔽分布式控制的实现细节，为上层应用提供全局网络视图；
  - 配置抽象（specification abstraction）：网络行为的表达通过网络编程语言实现，将抽象配置映射为物理配置。
- Overlay网络架构实现对基础网络设施的抽象。



Overlay网络



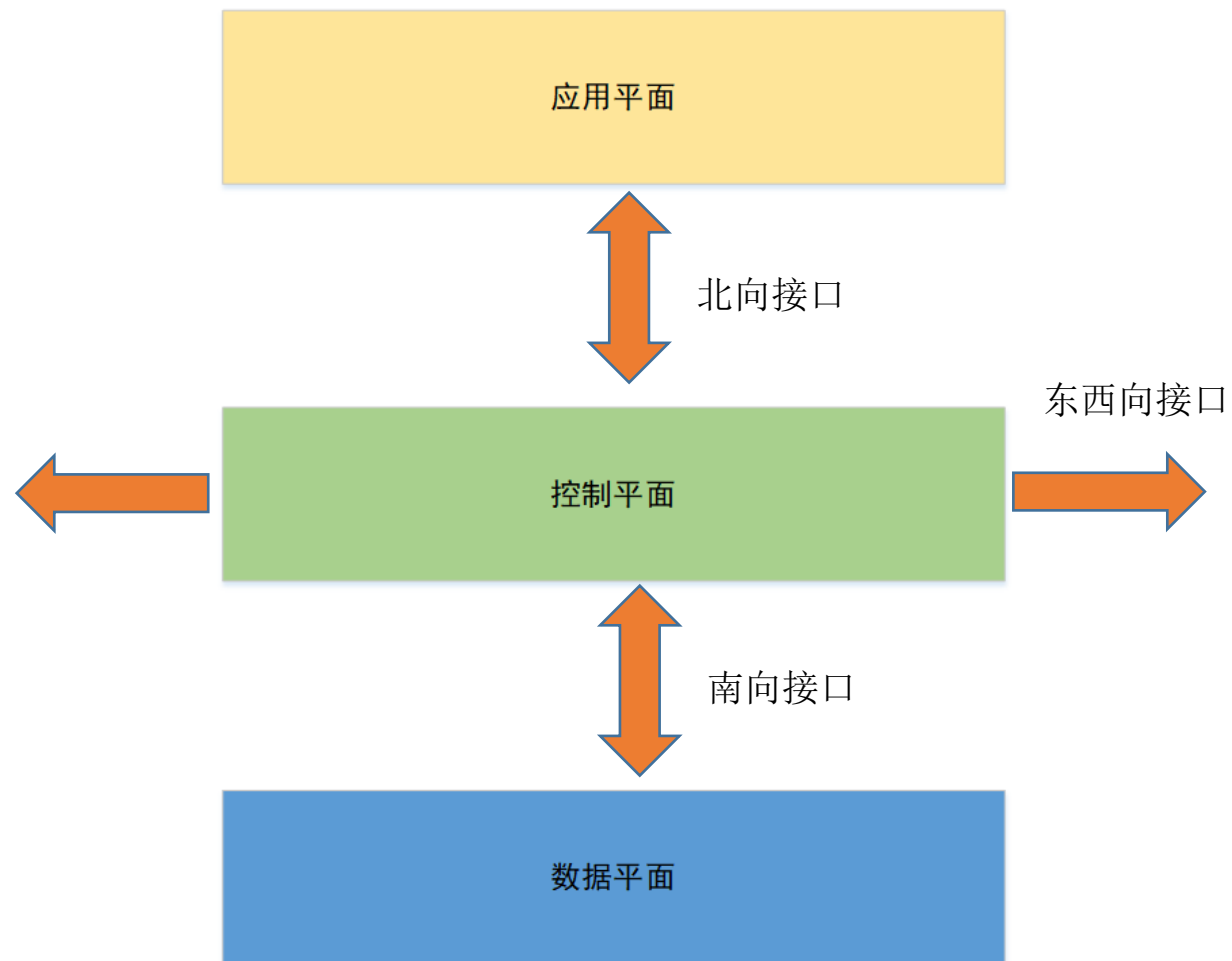
Underlay网络

# 网络可编程

- 传统网络的管理接口： CLI、SNMP等，是初级的网络编程方式；
- 网络管理者需要基于整个网络的，而不是基于某一设备的可编程，
- 网络可编程相关研究： 主动网络（Active Networking）、4D架构



# SDN开放可编程接口



# SDN可编程接口

- 网络可编程接口：
  - 北向接口： REST（Representational State Transfer） API， RESTCONF协议
  - 南向接口： OpenFlow、 OF-Config、 NETCONF、 OVSDB、 XMPP、 PCEP、 I2RS、 OPFlex等协议；
  - 东西向接口： 负责控制器之间的通信， 未形成统一标准。
- 数据平面可编程技术： Intel主导的DPDK、 斯坦福大学主导的P4(Programming Protocol-independent Packet Processors)等

谢谢！