Part1 SDN概述和基本原理

1.1 SDN的产生与发展

什么是SDN？

SDN(Software Defined Network)即软件定义网路，是一种网络设计理念，通过将网络设备控制面与数据面分离开来，从而实现了网络流量的灵活控制，让网络成为了一种可灵活调配的资源。

广义SDN：泛指面向上层应用开放资源接口，可实现软件编程控制的各类新型网络。

SDN的特点（数控分离）；

控制平面与数据平面分离；集中式控制模型；可编程；

SDN与传统网络的对比？

传统网络只可配置，不可编程；

数据平面和控制平面紧耦合。

1.3 SDN标准化

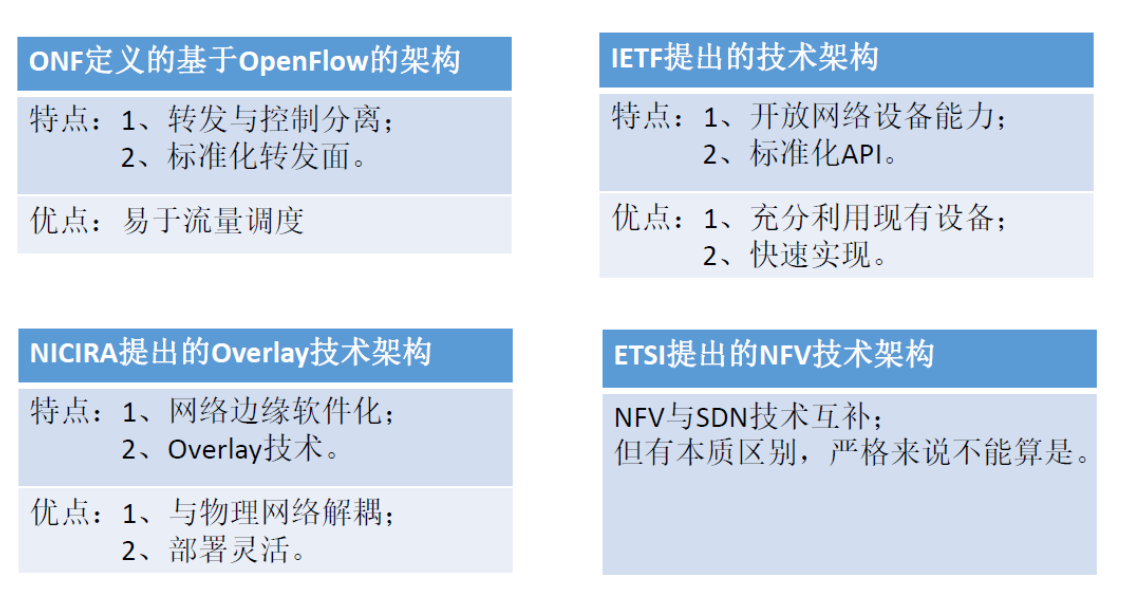
标准化组织：ONF、ITEF；

ONF：最为活跃、最具影响力的SDN标准化组织，SDN标准化和产业化最主要的推动力量。

IETF：互联网领域最具权威的技术标准化组织。由网络设备厂商主导，更关注SDN相关功能和技术如何在网络中实现的细节上。

1.4 SDN的基本架构

SDN主流架构；



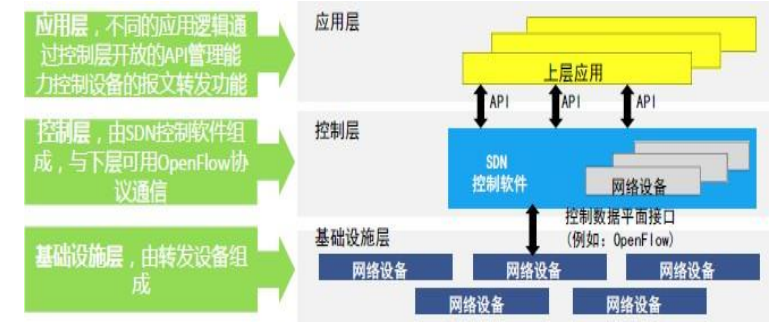
ONF定义的SDN基本架构（分层，架构图、内容）；

技术特点：  
1. 转发与控制的分离

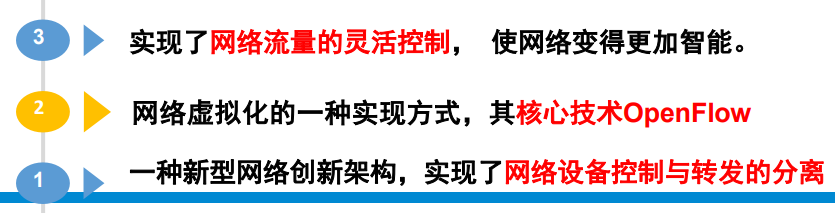
2. 转发面进行了标准化

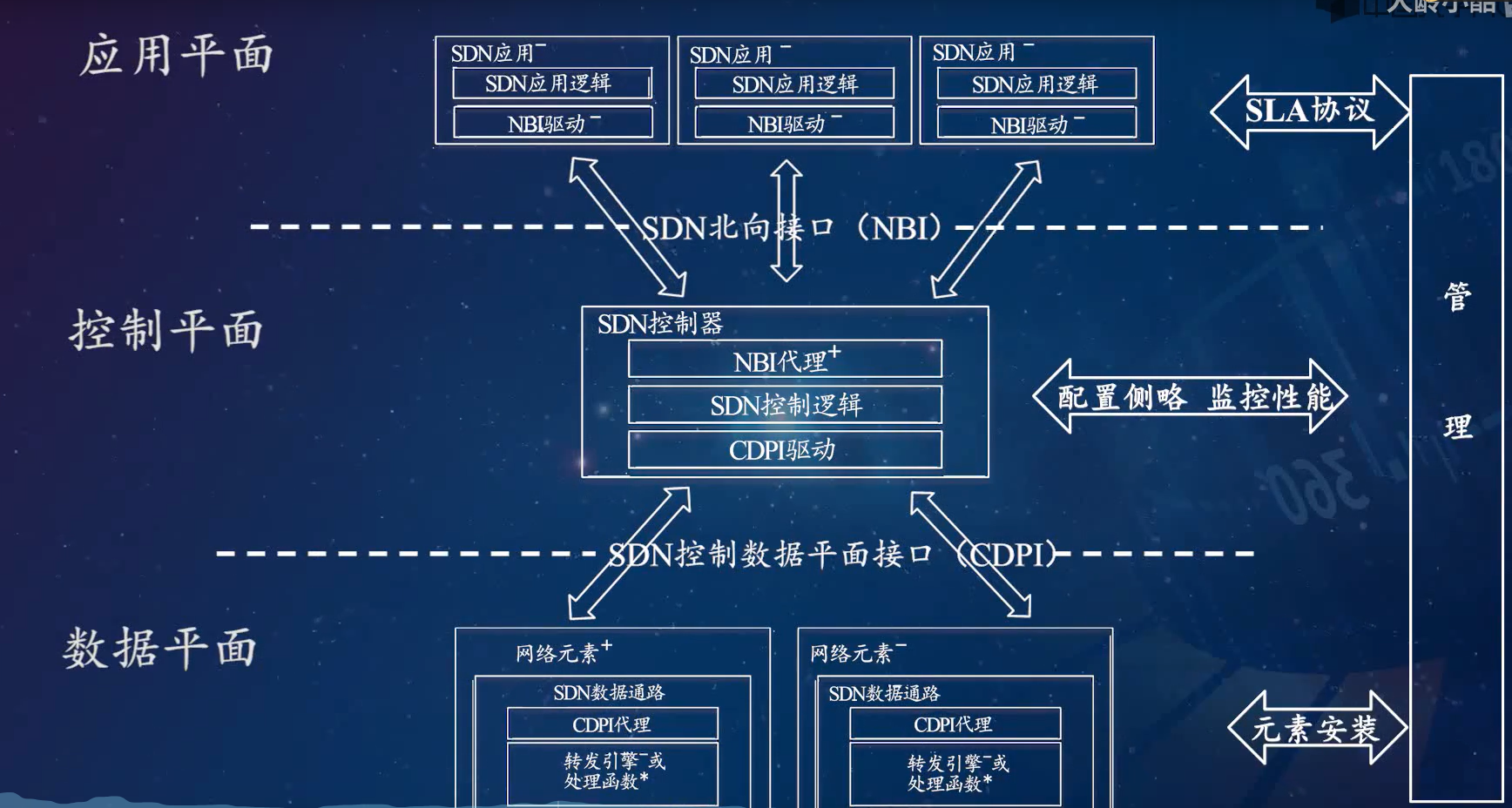
优点：

1. 流量调度
2. 开放生态链



基础设施层也叫转发层





其中，

数据平面的SDN Datapath：逻辑上的网络设备，负责转发和处理数据。

控制平面的主要两个任务：1. 将SDN应用层请求转换到SDN Datapath；2. 为SDN应用提供底层网络的抽象模型（状态或事件）

管理平面负责静态的工作：网元初始化配置，指定控制器，定义控制器及应用的控制范围。

南向接口：控制平面和数据平面之间的接口（CDPI）

功能：转发行为控制、设备性能查询、统计报告、事件通知等。

标准化的南向接口协议（Openflow），不依赖于底层具体厂商的交换设备

关键技术：转发面开放协议（南向接口协议）：允许控制器控制交换机的配置以及相关转发行为。

北向接口：应用平面与控制平面之间的接口（NBI）

功能：向应用层提供抽象的网络视图，使应用能够直接控制网络的行为。

是开放的、与厂商无关的接口

关键技术：SDN北向接口设计：控制器将网络能力封装后开放接口，供上层业务调用。REST API成为SDN北向接口的主流设计。

1.5 SDN的核心思想

解耦、抽象和可编程及其涵义；

解耦：数据平面与控制平面的解耦。

1. 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。
2. 是实现网络逻辑集中控制的前提。
3. 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。

解耦带来的问题与挑战：

1. 网络规模的扩大，单一控制器成为网络性能的瓶颈 –>>多个分布式的控制器
2. 多控制器如何交换路由信息，保持分布式网络节点状态的一致性，是一个重要的挑战。
3. 控制平面的响应延迟，导致数据平面的可用性问题。

抽象：ONF网络架构实现转发抽象、分布状态抽象和配置抽象

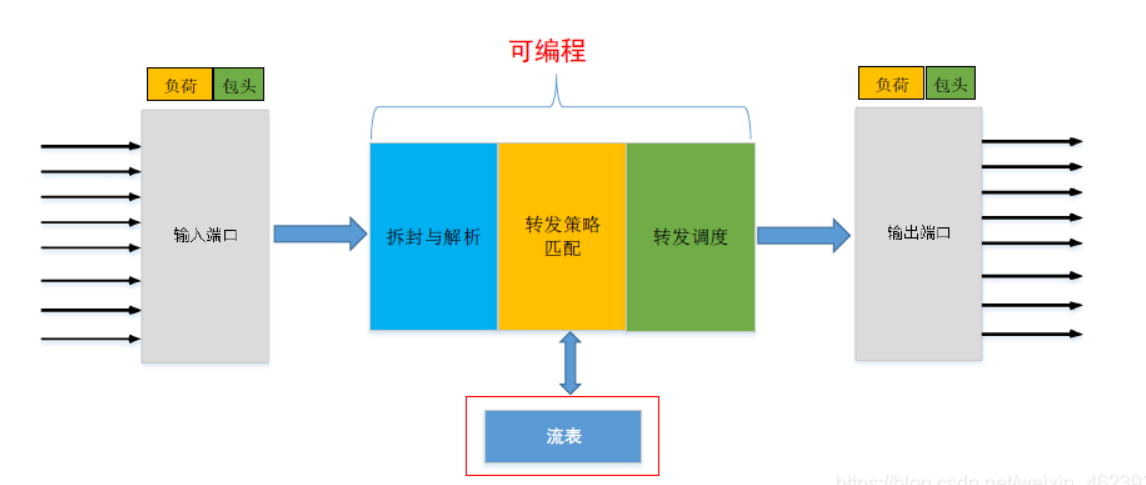
1. 转发抽象（forwarding abstraction）: 隐藏了底层的硬件实现，转发行为与硬件无关；把各种转发表项如mac表，路由表，MPLS标签表，ACL访问控制表统一抽象成流表。
2. 分布状态抽象（distribution abstraction）: 屏蔽分布式控制的实现细节，为上层应用提供全局网络视图；控制层将设备的分布式状态抽象成全局的网络视图，从而达到逻辑上的集中控制。
3. 配置抽象（specification abstraction）: 网络行为的表达通过网络编程语言实现，将抽象配置映射为物理配置。

可编程：北向接口，南向接口。

Part2 数据平面

2.1 SDN数据平面

SDN数据平面的定义和特点；



处理流程中的：解析（Parser）、转发（Forwarding）和调度（Scheduling）都是可编程、协议无关的；

传统网络设备的二层或三层转发表抽象成流表。

2.2 OpenFlow

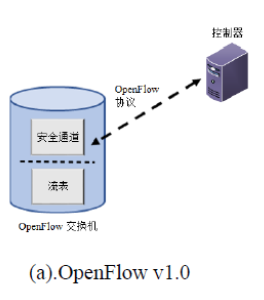
OpenFlow架构；

1. 流表（Flow Table），每个动作（Action）关联一个流表项（Flow Entry），指示交换机如何进行流（Flow）的处理；
2. 安全通道（Secure Channel），OpenFlow交换机通过安全通道与远端控制器链接，负责控制器与交换机之间的交互；
3. OpenFlow协议（Protocol），定义了一种南向接口标准，为控制器与交换机之间的通信提供了一种开放标准的方式。

OpenFlow主要版本和特点（重点是OpenFlow v1.0）；



OpenFlow v1.0的组成结构；



2.3 OpenFlow流表

流表的定义；

流的概念：

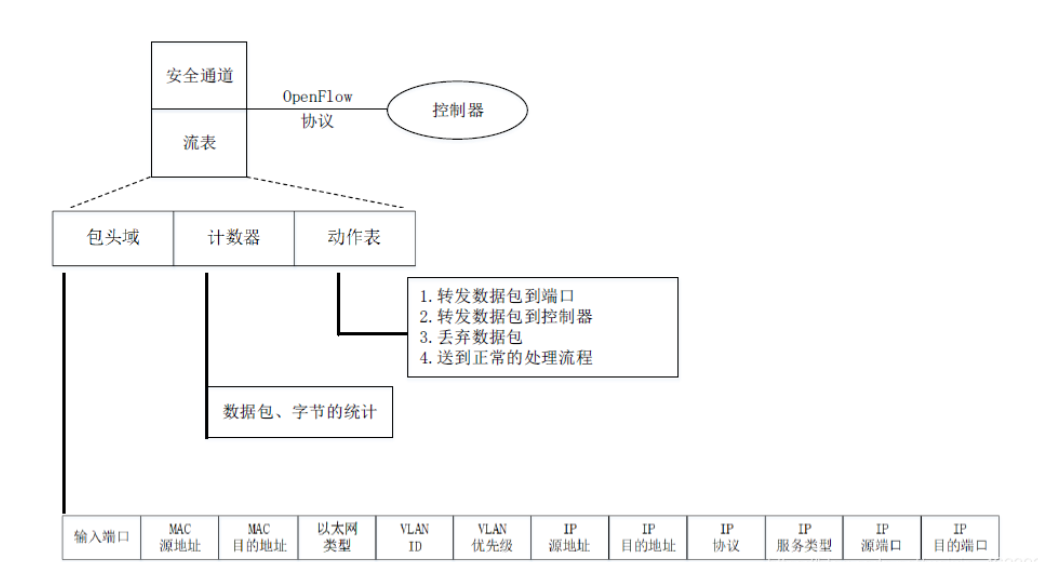
* 同一时间，经过同一网络中具有某种共同特征（属性）的数据，抽象为一个流。比如，可以将访问同一目的地址的数据视为一个流；
* 流一般由网络管理员定义，根据不同的流执行不同的策略；
* OpenFlow体系中，数据以“流”为单位进行处理。

流表：针对特定流的策略表项的集合，负责数据包的查找与转发。一张流表包含了一系列的流表项。（Flow Entries）

V1.0的流表结构及其内容（包头域、计数器和动作）；

流表项包括三个域：

* 包头域（Head Fields）：涵盖了链路层、网络层和传输层大部分标识；
* 计数器（Counters）：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；
* 动作表（Actions）：指示与该流表项匹配的数据包应该执行的下一步操作。



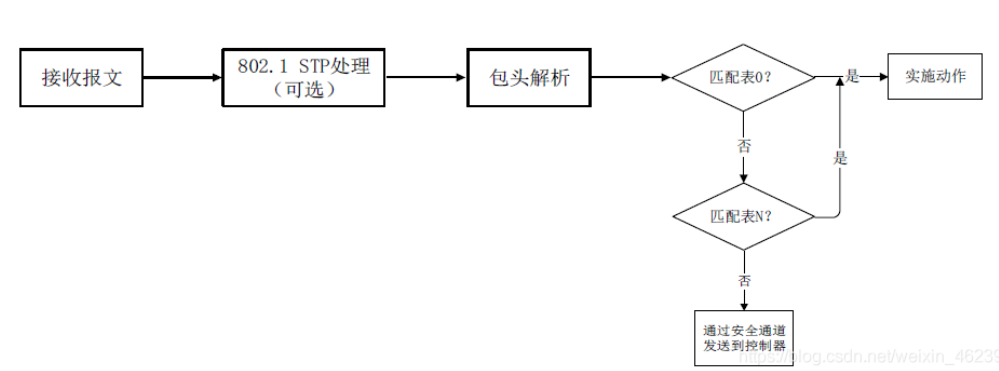
流表动作列表（常用必备动作、可选动作）；

* 必备动作：需要所有OpenFlow交换机默认支持
* 可选动作：需要交换机告知控制器它所能支持的动作种类



OpenFlow数据包处理流程；

每个包按照优先级依次去匹配流表中表项，匹配包的优先级最高的表项即为匹配结果。匹配成功，对应的计数器将更新，同时实施转发动作；如果没能找到匹配的表项，则转发给控制器。



OpenFlow的保留端口及内容；

· 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

· TABLE：对packet\_out数据包执行流表操作

· LOCAL：转发给本地的网络栈

· IN\_PORT：从入端口发出

· NORMAL：利用交换机的传统转发机制处理数据包

· FLOOD：按照最小生成树从设备出端口洪泛发出

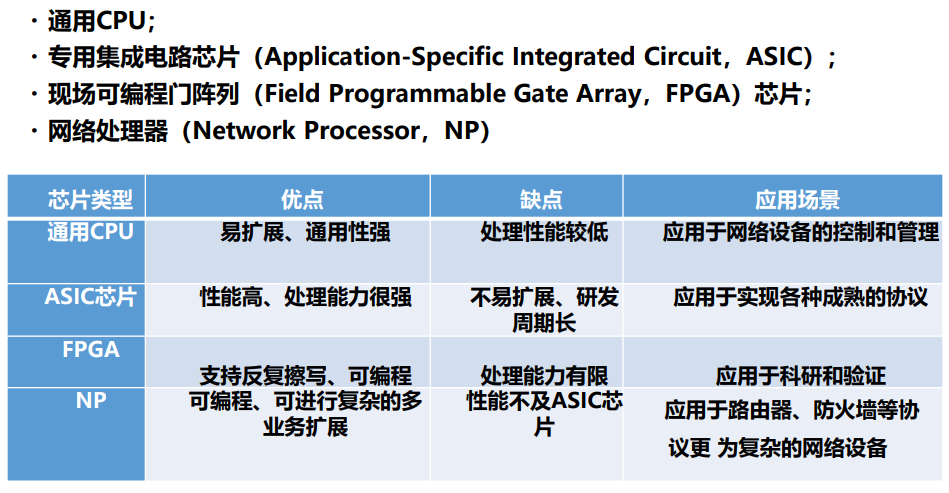
2.4 SDN交换机

交换机类型；

根据交换机的应用场景及其所能够支持的流表动作类型，OpenFlow交换机可以被分为：

* OpenFlow专用交换机（OpenFlow-only）：只支持OpenFlow协议
* OpenFlow使能交换机（OpenFlow-enabled）：
  + OpenFlow1.1及后续版本将其更名为“OpenFlow-hybrid”
  + 考虑了OpenFlow交换机与传统交换机混合组网时可能遇到的协议栈不兼容问题，能同时运行OpenFlow协议和传统的二层/三层协议栈
  + 支持OpenFlow可选转发动作中的NORMAL动作

交换芯片类型；



SDN交换机；

SDN物理（硬件）交换机：多数为支持OpenFlow的混合模式交换机，支持传统二、三层处理模式和SDN处理模式

SDN虚拟（软件）交换机：成本低、配置灵活、性能满足中小规模网络要求

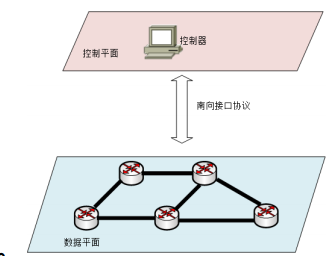
SDN交换机选型的参数考虑(例如背板带宽、频率等)；

* 背板带宽：
  + 交换机接口处理器和数据总线间的最大吞吐数据量，背板带宽越高，所能处理数据的能力就越强
  + 背板带宽从几个Gbps到几百Gbps不等，不是越高越好，只要满足端口转速转发就可以
  + 对于转速转发的交换机，背板的带宽计算方法：端口数 \* 相应端口 \* 2（全双工模式）
* 端口密度：代表交换机转发能力。密度越大，设备的转发能力越强。
* 端口速率：速率越高，设备的处理性能越强；
* 支持模块类型：类型越多，实用性越强，可应用于不同的网络环境，比如：LAN接口、WAN接口、ATM接口
* 端口带宽类型：越丰富越好，既支持40G、100G高速端口，又支持百兆、千兆低速端口，还支持XFP又支持SFP、SFP+、CFP等等多种光接口类型。

Part3 南向接口协议

3.1 南向接口协议概述

南向接口的定义（SDN架构中的位置）；



什么是南向接口协议？

为控制平面的控制器和数据平面的交换机之间的信息交互而设计的协议。

常见南向接口协议及其设计目标（OpenFlow、OF-config、NETCONF、OVSDB）；

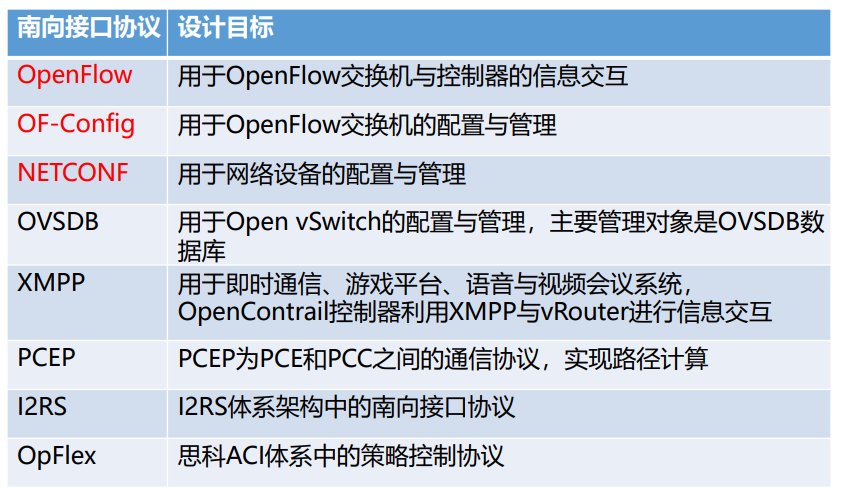
· 实现数据平面与控制平面的信息交互

· 向上收集数据平面信息；

· 向下下发控制策略，指导转发行为

· 实现网络的配置与管理

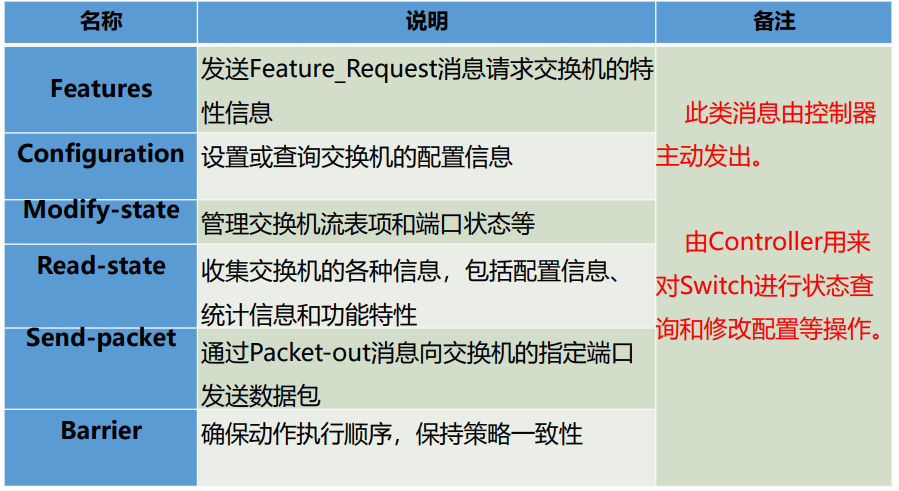
· 实现路径计算，包括链路属性（带宽与开销）、链路状态、和拓扑信息等



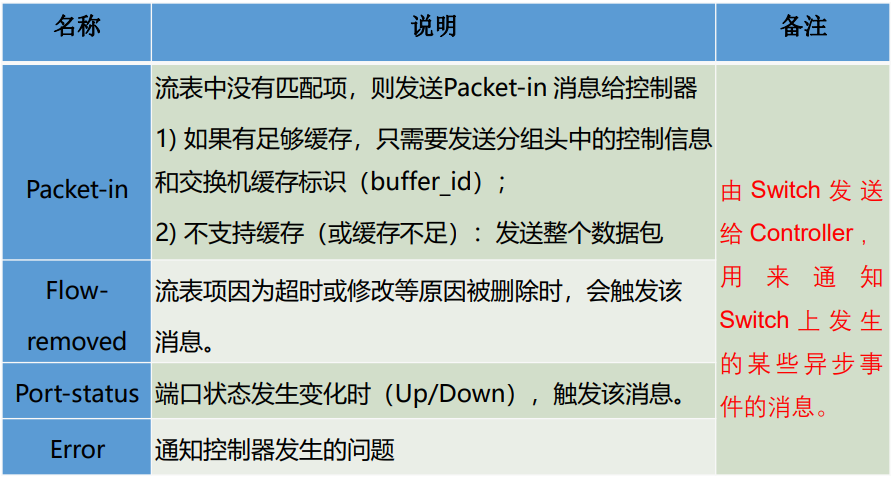
3.2 OpenFlow协议

OpenFlow协议的消息类型；

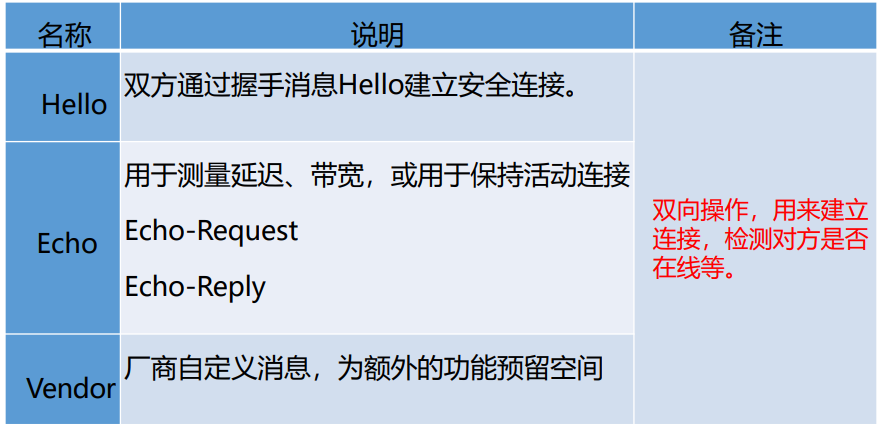
* Controller-to-Switch消息：由控制器发起，用来管理或获取OpenFlow交换机的状态



* Asynchronous（异步）消息: 由OpenFlow交换机发起，用来将网络事件或交换机状态变化更新到控制器



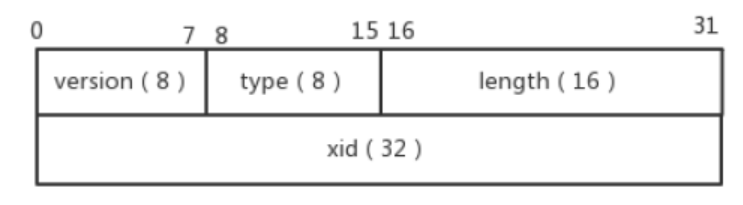
* Symmetric(对称)消息：由交换机或控制器发起。

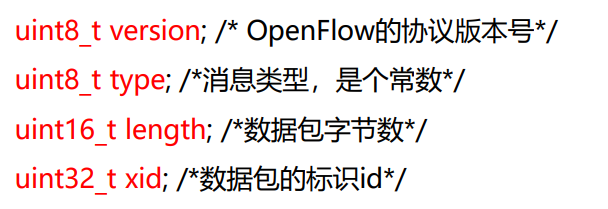


消息格式；

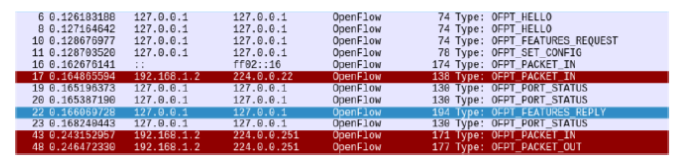
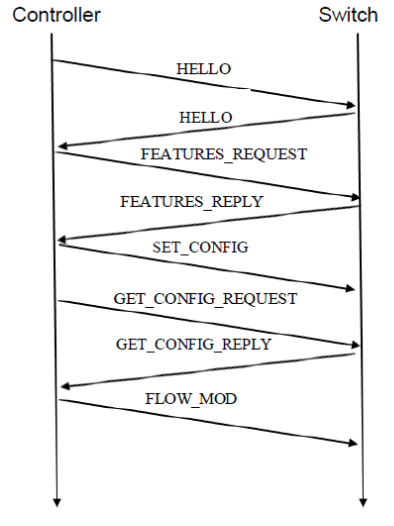
OpenFlow协议数据包：OpenFlow Header和OpenFlow Message

OpenFlow Header格式：





OpenFlow的SDN通信流程；



左边的图通过hello信息连接建立的过程：

◆控制器开启TCP的6633端口等待

交换机的连接

◆交换机启动时,尝试连接到指

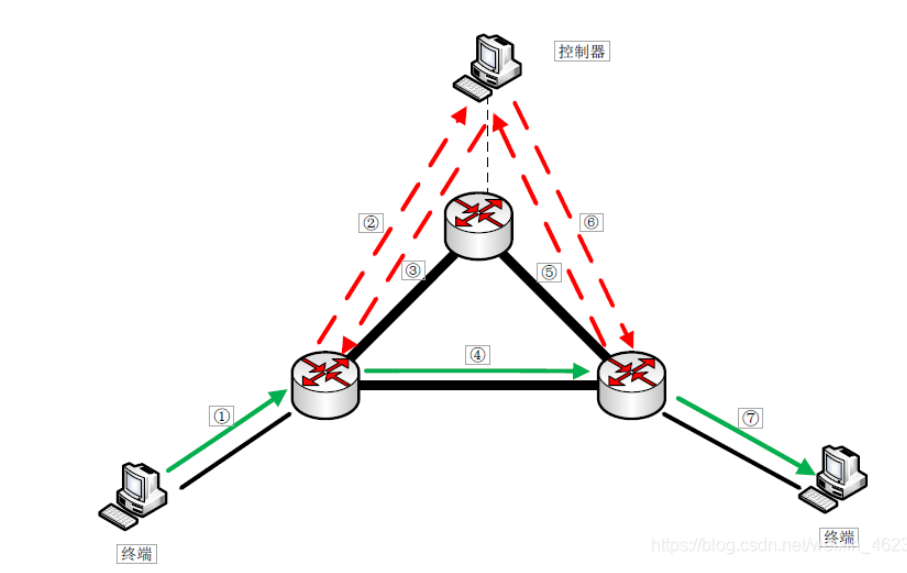
定控制器端口;

◆安全通道采用TLS( Transport

Layer Security)连接加密;

◆双方通过交换证书进行认证

后续的过程可以参考上方的信息



两PC终端的信息交互过程：

①主机向网络发送数据包

②OF交换机流表无匹配项，通过

Packetln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

3.3 OF-Config

协议概述，设计目标和设计思想；

OF-Config协议是一种OpenFlow交换机管理配置协议，是OpenFlow的伴侣协议

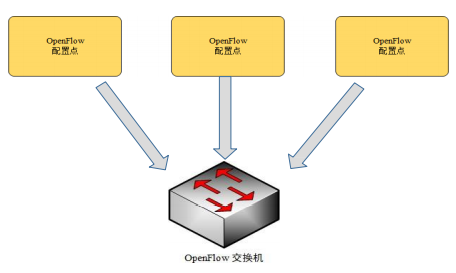
协议设计需求（Requirements）

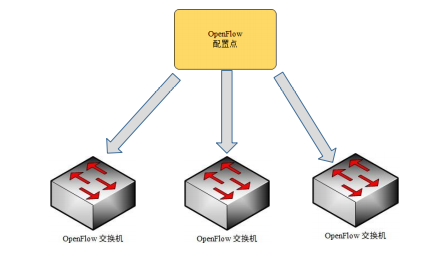
* 配置需求（Specification Requirements）
* 操作运维需求（Operational Requirements）
* 管理协议需求（Management Protocol Requirements）

配置需求：

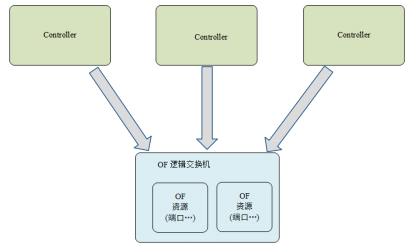
* 控制器连接设置：控制器IP地址、控制器端口号和传输协议（TLS或TCP）
* 多控制器（Multiple Controllers）：多控制器连接的参数配置
* OpenFlow逻辑交换机：
* 连接中断（Connection Interruption）：配置连接失效后进入的模式
* 加密（Encryption）：身份认证方式
* 队列（Queues）：最小速率（min-rate）、最大速率（max-rate）、自定义速率（experimenter）
* 端口（Ports）：禁止接收（no-receive）、禁止转发（no-forward）、禁止Packet-in消息（no-packetin）、管理状态（admin-state）
* 能力发现（Capability Discovery）：发现逻辑交换机能力特性
* Datapath ID

操作运维需求：  
· 支持OF交换机被多个OpenFlow配置点配置；

  
· 支持一个OpenFlow配置点管理多个OF交换机



* 支持一个OpenFlow逻辑交换机被多个控制器控制



* 支持配置OpenFlow交换机的端口和队列；
* 支持OpenFlow逻辑交换机的能力发现；
* 支持配置隧道，如IPinGRE、VXLAN及NVGRE

管理协议需求：

* 保障安全性，支持对交换机和配置点双向认证；
* 支持配置请求和应答的可靠传输
* 支持由配置点或交换机进行连接设置
* 能够承载局部交换机配置以及大范围交换机配置
* 支持配置点再交互及配置参数接收来自交换机的配置参数
* 支持在交换机创建、更改及删除配置信息，并支持报告配置结果
* 支持独立发送配置请求，并支持交换机到配置点的异步通知
* 支持记忆能力、可伸展性以及报告自身属性和能力

OF-Config数据模型（XML）和传输协议（NETCONF）



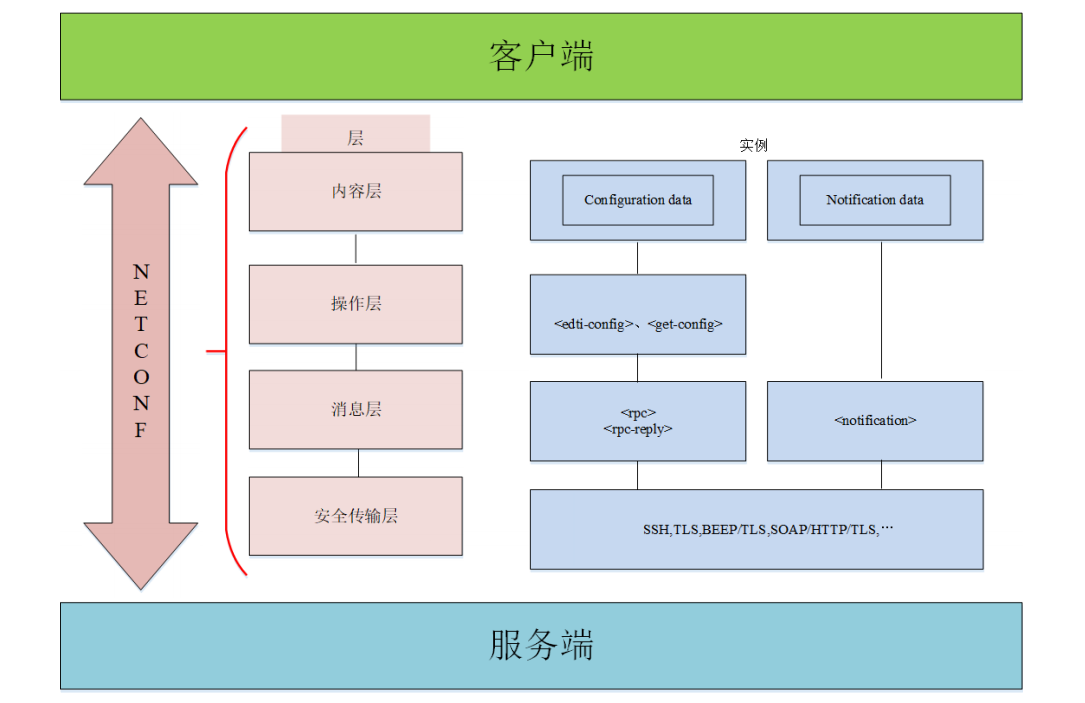
通过NETCONF协议来传输其内容（RFC6241）

3.4 NETCONF

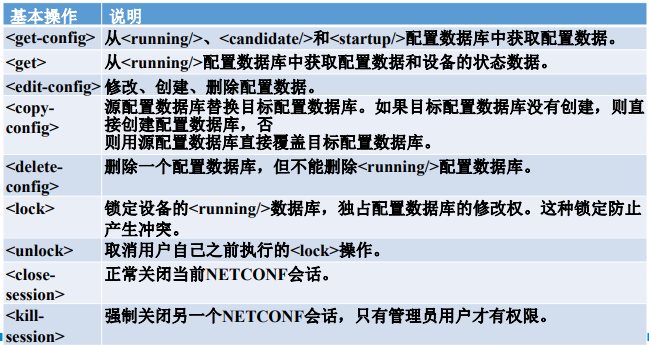
什么是NETCONF协议？设计目标；

新一代网管协议；网络配置协议NETCONF（Network Configuration Protocol）提供一套管理网络设备的机制。

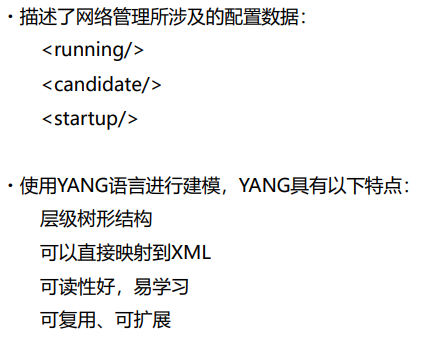
NETCONF协议框架（操作层和内容层的内容）



操作层：对数据库信息的获取、配置、复制和删除等功能



内容层：



Part4 SDN控制平面与北向接口协议

4.1 SDN控制平面概述

什么是SDN控制平面（SDN架构中的位置）？

SDN控制平面：一个或多个SDN控制器组成，是网络的大脑。

* 对底层网络交换设备进行集中管理，状态检测，转发决策以及处理和调度数据平面的流量；
* 向上层应用开放多个层次的可编程能力

南向控制协议的任务和功能（链路发现、拓扑管理、策略制定、表项下发及其涵义）；

通过南向接口协议进行链路发现、拓扑管理、策略制定、表项下发等；

* 链路发现和拓扑管理：利用上行通道对底层交换设备上报信息进行统一监控和统计
* 策略制定和表项下发：利用下行通道对网络设备实施统一控制

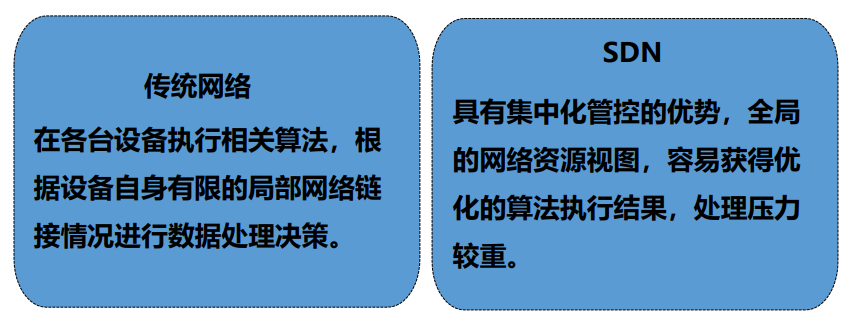
链路发现：获得SDN全网信息，实现网络地址学习、VLAN、路由转发

* OF交换机直连链路的发现：LLDP协议
* OF交换机非直连链路的发现：广播

拓扑管理：监控和采集SDN交换机的信息，反馈工作状态和链路连接状态

* 定时发送带LLDP数据包的Packet\_out消息，根据Packet\_in消息获知交换机信息，检测交换机工作状态，完成网络拓扑视图更新
* 导致较慢的收敛过程

策略制定：

* 流表生成算法是影响控制器智能化水平的关键因素
* 针对不同层次的传输需求，制定相应的转发策略并生成对应的流表项  
  

表项下发：通过流表下发机制控制交换机的数据包转发

* 主动（proactive）下发：数据包到达交换机之前进行流表设置
* 被动（reactive）下发：交换机接收到一个数据包并且没有发现匹配的流表项，将其送给控制器处理。

北向业务支撑方式；

· 通过北向接口为上层业务应用及资源管理系统提供灵活的网络资源抽象

· 北向接口定义是SDN领域关注和争论的焦点

· REST API 是用户比较容易接受的方式

4.2 主流开源控制器

NOX（C++）、POX（Python）、Ryu（Python）、Floodlight（java）、OpenDayLight（Java）

POX:

* 采用Python语言开发的基于OpenFlow的控制器；
* 是NOX的兄弟版本，实现了简单的控制平面的功能。

包括：

* 内核（core）：OpenFlow和of\_01;
* 组件（component）

OpenDayLight：

* 提供了一个模型驱动服务抽象层( MD-SAL),允许用户采用不同的南向协议在不同厂商的底层转发设备上部署网络应用
* 与其他控制器架构的明显差别是Open Daylight架构中有服务抽象后SAL( Service Abstraction Layer)。SAL主要完成插件的管理,包括注册、注销和能力的抽象等功能。

Part5 SDN北向接口协议

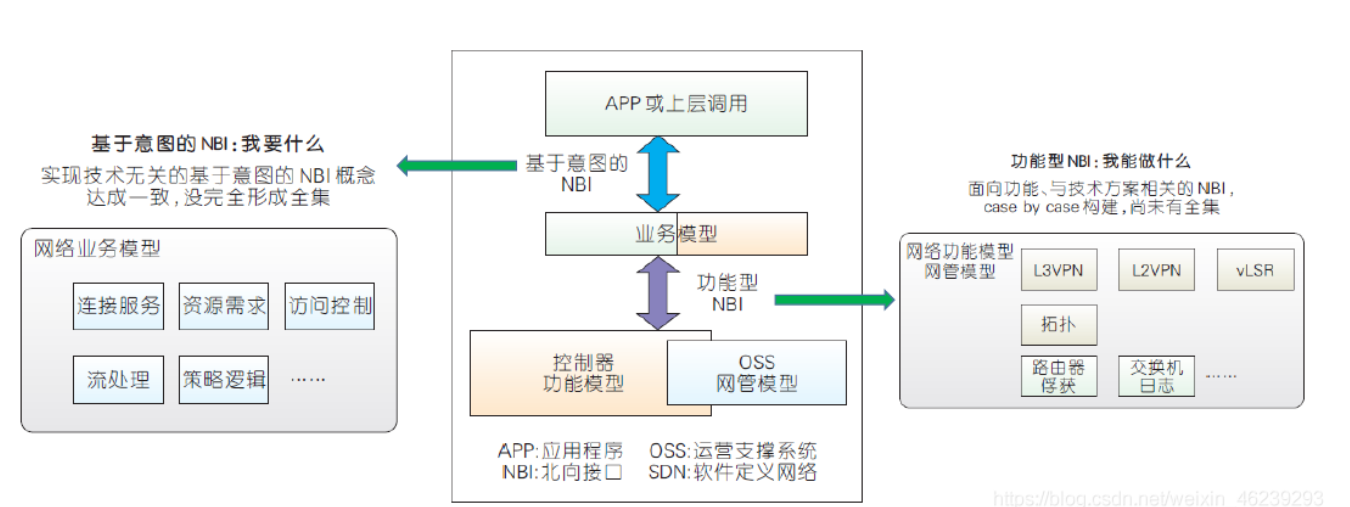
5.1 SDN北向接口概述

什么是北向接口？

应用平面与控制平面之间的接口（NBI），通过控制器向上层业务应用开放的接口，为上层业务应用和资源管理系统提供灵活的网络资源抽象。

北向接口的设计（功能型、基于意图）与网络模型；

* 功能型北向接口（Functional NBI）：自下而上看网络，重点在网络资源抽象及控制能力的开发，包括Topology、L2VPN、L3VPN、Tunnel等接口。
* 基于意图的北向接口（Intent-based NBI）：自上而下看网络，关注应用或者服务的需求，同具体的网络技术无关。



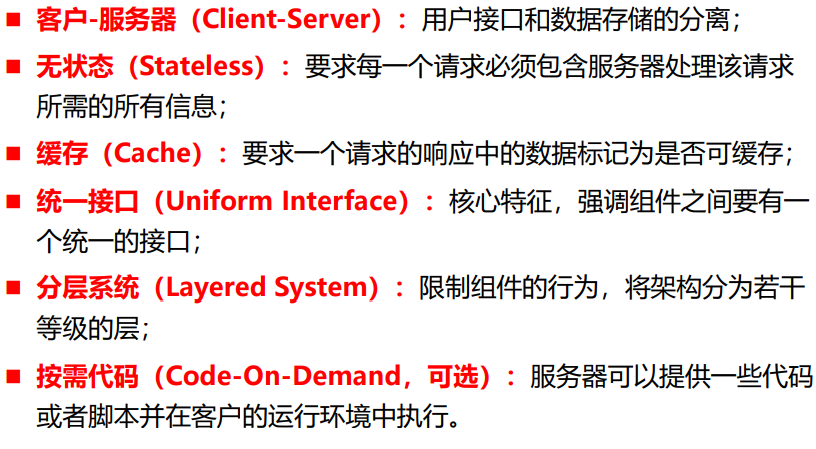
北向接口的实现（主流实现——RESTAPI）；

REST API是北向接口的主流实现方式

REST的几个重要概念：



REST的约束条件与原则



5.2 RESTAPI及其设计

RESTAPI的设计；

HTTP动词+URI

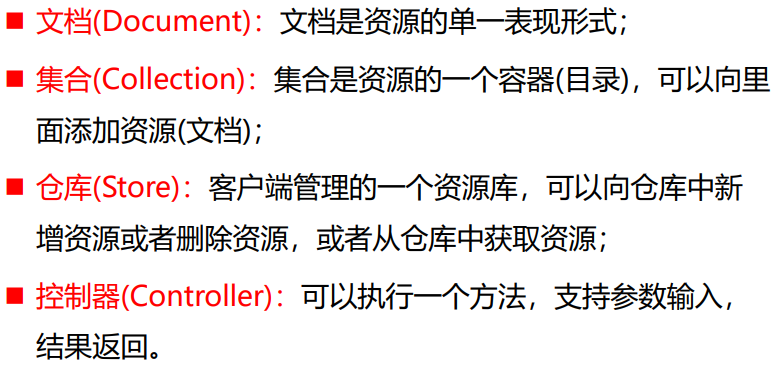
* HTTP动词：描述操作
* URI：标识资源

HTTP动词；

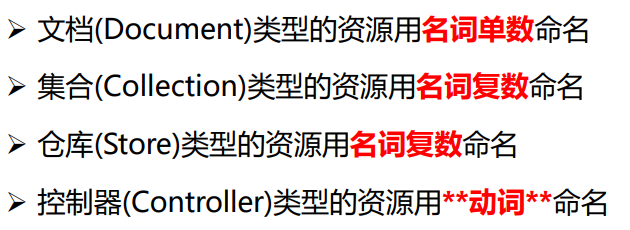


URI规范；

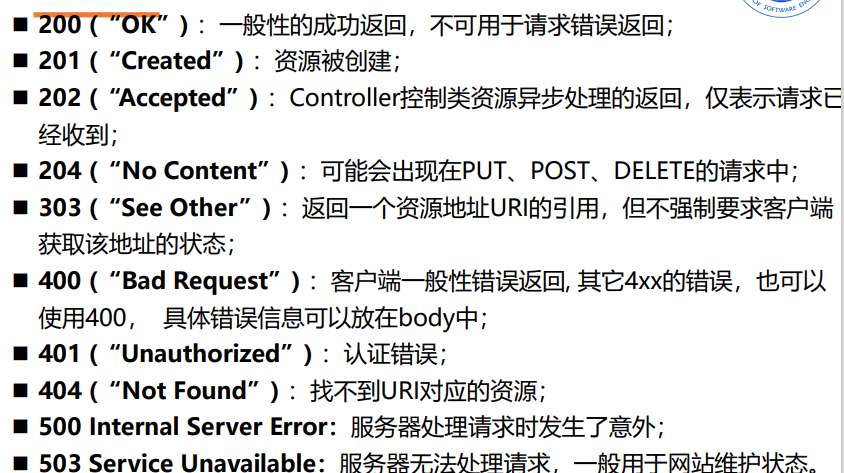
资源的类型：



命名规范：



HTTP响应状态码；



Part6 网络虚拟化&NFV

6.1 虚拟化技术

什么是虚拟化？

虚拟化是资源的逻辑表示，它不受物理限制的约束。

“虚拟化”的定义包含了三层含义：

* 虚拟化的对象是各种各样的资源
* 经过虚拟化后的逻辑资源对用户隐藏了不必要的细节
* 用户可以在虚拟环境中实现其真实环境的部分或全部功能

虚拟化的特点与优点；

· 集中化管理

· 提高硬件利用率

· 动态调整机器/资源配置

· 高可靠性

· 减低总体成本

· 降低终端设备数量

6.2 网络虚拟化

网络虚拟化，vNetwork及其组件；

网络虚拟化就是在一个物理网络上模拟出多个逻辑网络来。 网络虚拟化的内容一般指虚拟专用网络(VPN)。 VPN对网络连接的概念进行了抽象，允许远程用户访问组织的内部网络，就像物理上连接到该网络一样。 网络虚拟化可以帮助保护 IT 环境，防止来自 Internet 的威胁，同时使用户能够快速安全的访问应用程序和数据。

vNetwork的组件主要包括虚拟网络接口卡vNIC、vNetwork标准交换机vSwitch和vNetwork分布式交换机dvSwitch。

虚拟网络接口卡：每个虚拟机都可以配置一个或者多个虚拟网络接口卡vNIC，vNIC拥有独立的MAC地址以及一个或者多个IP地址，且遵守标准的以太网协议。

虚拟交换机vSwitch：虚拟交换机用来满足不同的虚拟机和管理界面进行互联。虚拟交换机的工作原理与以太网中的第2层物理交换机一样。

分布式交换机：dvSwitch将原来分布在一台ESX主机上的交换机进行集成，成为一个单一的管理界面，在所有关联主机之间作为单个虚拟交换机使用。这使得虚拟机可在跨多个主机进行迁移时确保其网络配置保持一致。

虚拟设备；

6.3 网络功能虚拟化NFV

NFV的产生背景；

NFV让电信、移动提供商和运营商有能力提供更好的数字业务，加快新服务投放市场的速度，可以借此与大的软件公司竞争。NFV提供一定程度的架构、资本、vendor-sourcing等方面的敏捷性，与传统的基于专用运营商级网络设备的实现方式不同。

传统的通过物理设备提供服务的方式有如下特点，大大降低了敏捷性：  
· 部署周期和成本高，物理设备必须购买且部署这些物理设备需要大量的人力，可能需要数月的时间才能完成部署。

* 高运营成本，物理设备需要特殊地理环境，能源和冷却消费很高。
* 效率低，专用设备必须部署在能够提供最小服务覆盖的地方，不管 实际的使用和需求是多少，这导致了不能充分利用和成本浪费。
* 厂商锁定，单一厂商的电气设备和采购其他产品的负担非常重。

通信4.0时代，以IT化为核心，具有敏捷化、开放化和软件化这三大显著特征。归根到

底，根本特征是虚拟化。

NFV的定义；

NFV（Network Function Virtualization），即网络功能虚拟化，背景是电信网络，旨在采用虚拟化的方法，将原本运行在专用中间设备（middle box）的网络功能（如网关、防火墙）用软件的方式实现，通过在标准的通用设备（服务器、存储器、交换机）中运行的虚拟网络功能（VNF）得以实现。

NFV框架：

纵向分层：基础设施层、虚拟网络层、运营支撑层

基础设施层：NFVI是NFV Infrastructure的简称，从云计算的角度看，就是一个资

源池。 NFVI映射到物理基础设施就是多个地理上分散的数据中心，通过高速通信网

连接起来。 NFVI需要将物理计算/存储/交换资源通过虚拟化转换为虚拟的计算/存

储/交换资源池。 虚拟网络层：虚拟网络层对应的就是目前各个电信业务网络，每个物理网元映射为

一个虚拟网元VNF，VNF所需资源需要分解为虚拟的计算/存储/交换资源，由NFVI

来承载，VNF之间的接口依然采用传统网络定义的信令接口（3GPP+ITU-T)，

VNF的业务网管依然采用NE-EMS-NMS体制。

运营支撑层：运营支撑层就是目前的OSS/BSS系统，需要为虚拟化进行必要的修改

和调整。

横向分域：业务网络域，管理编排域（MANO，包括哪些内部实体）

业务网络域：就是目前的各电信业务网络

管理编排域：同传统网络最大区别就是，NFV增加了一个管理编排域，简称MANO

，MANO负责对整个NFVI资源的管理和编排，负责业务网络和NFVI资源的映射和

关联，负责OSS业务资源流程的实施等，MANO内部包括VIM，VNFM和Orchestrator

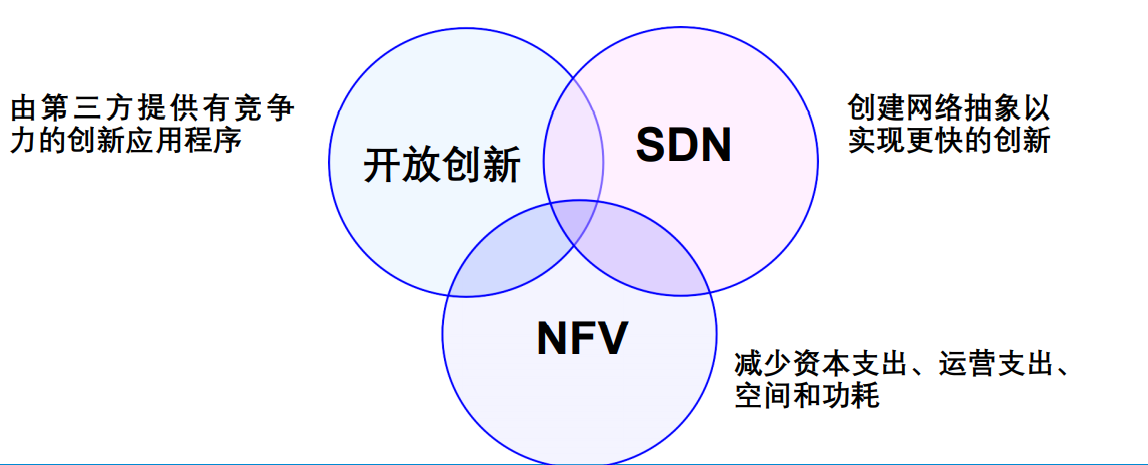
三个实体，分别完成对NFVI，VNF和NS（Network Service：即业务网络提供的网络服

务）三个层次的管理。

6.4 NFV与SDN

有哪些区别与联系？

NFV和SDN是高度互补的。这两个主题是互利的，但不是相互依赖的。



NFV：网络设备体系结构的再定义

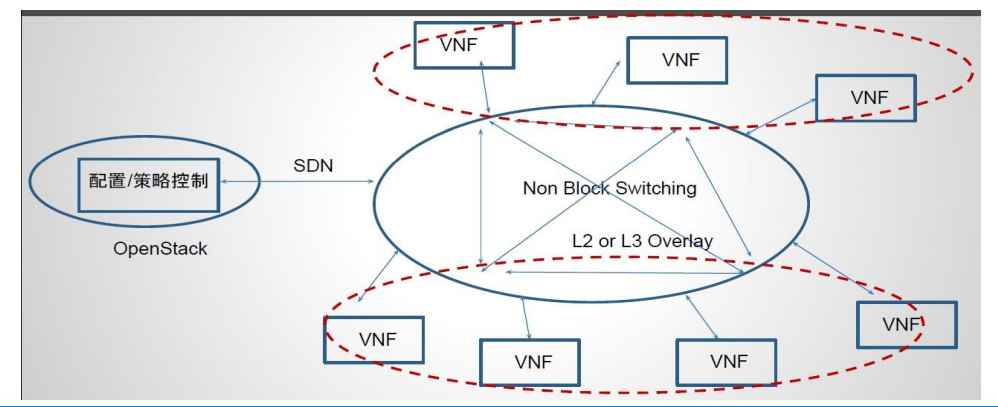
NFV的诞生是为了满足服务提供商（SP）的需求：

* 通过减少/消除专有硬件降低资本支出
* 将多种网络功能整合到行业标准平台上

SDN：网络体系结构的重新定义

SDN来自IT世界

* 分离数据层和控制层，同时集中控制
* 提供使用定义良好的接口编程网络行为的能力



NFV可以是SDN运行的“基质”

Part7 云计算网络与Overlay

7.1 云计算网络

虚拟化对传统数据中心提出的挑战

传统的三层数据中心架构结构的设计是为了应付服务客户端-服务器应用程序的纵贯式大流量，同时使网络管理员能够对流量流进行管理。工程师在这些架构中采用生成树协议(STP)来优化客户端到服务器的路径和支持连接冗余。

虚拟化从根本上改变了数据中心网络架构的需求。 最重要的一点就是，虚拟化引入了虚拟机动态迁移技术。从而要求网络支持大范围的二层域。从根本上改变了传统三层网络统治数据中心网络的局面。

7.2 Overlay网络

Overlay技术的由来

1. 虚拟机迁移范围收到网络架构限制：由于虚拟机迁移的网络属性要求，其从一个物理机上迁移到另一个物理机上，要求虚拟机不间断业务，则需要其IP地址，MAC地址等参数维持不变，如此则要求业务网络是一个二层网络，且要求网络本身具备多路径多链路的冗余和可靠性。
2. 虚拟机规模受网络规格限制：在大二层网络环境下，数据流均需要通过明确的网络寻址以保证准确到达目的地，因此网络设备的二层地址表项大小（即MAC地址表），成为决定云计算环境下虚拟机规模的上限。
3. 网络隔离/分离能力限制：当前的主流网络隔离技术为VLAN（或VPN），在大规模虚拟化环境下部署会有问题，VLAN数量在标准定义中只有12个比特单位，即可用的数量为4000个左右，这样的数量级对于公有云或大型虚拟云计算应用而言微不足道。在此驱动力基础上，逐步演化出Overlay的虚拟化网络技术。

Overlay技术的定义及其特征，组成部分

Overlay指一种网络架构上叠加的虚拟化技术模式，对基础网络不进行大规模修改下，以基于IP的网络技术为主承载应用，并于其它网络业务分离。

针对以上提出的三大技术挑战，Overlay在很大程度上提供了全新的解决方式

1. 针对虚拟机迁移范围受到网络架构限制的解决方式：Overlay是一种封装在IP报文之上的新的数据格式，因此，这种数据可以通过路由的方式在网络中分发，而路由网络本身并无特殊网络结构限制，具备良性大规模扩展能力，并且对设备本身无特殊要求，以高性能路由转发为佳，且路由网络本身具备很强的故障自愈能力、负载均衡能力。
2. 针对虚机规模受网络规格限制的解决方式：虚拟机数据封装在IP数据包中后，对网络只表现为封装后的网络参数，即隧道端点的地址，因此，对于承载网络（特别是接入交换机），MAC地址规格需求极大降低，最低规格也就是几十个（每个端口一台物理服务器的隧道端点MAC）。
3. 针对网络隔离/分离能力限制的解决方式：针对VLAN数量4000以内的限制，在Overlay技术中引入了类似12比特VLAN ID的用户标识，支持千万级以上的用户标识，并且在Overlay中沿袭了云计算“租户”的概念，称之为Tenant ID(租户标识)，用24或64比特表示。针对VLAN技术下网络的TRUANK ALL(VLAN穿透所有设备)的问题，Overlay对网络的VLAN配置无要求，可以避免网络本身的无效流量带宽浪费，同时Overlay的二层连通基于虚机业务需求创建，在云的环境中全局可控。

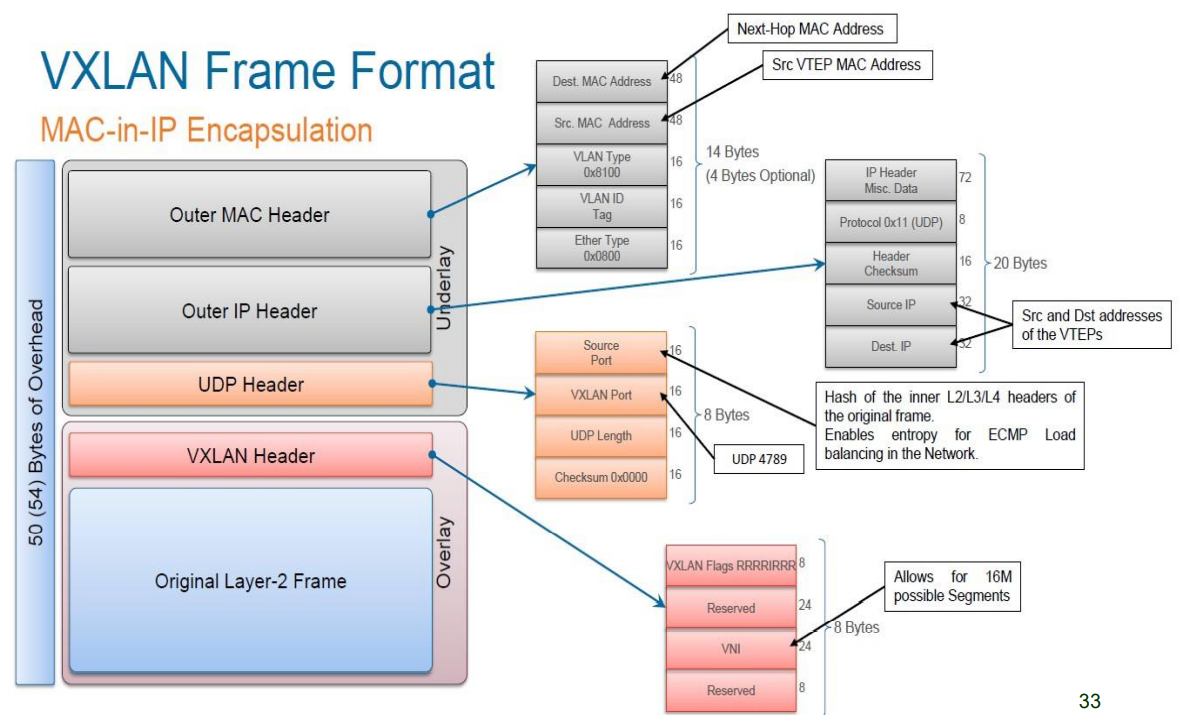
组成部分：

1. 边缘设备：与虚拟机直接相连的设备
2. 控制平面：主要负责虚拟隧道的建立维护以及主机可达性信息的通告
3. 转发平面，承载Overlay报文的物理网络

什么是VXLAN？

VXLAN报文是在原始二层报文前面再封装一个新的报文，新的报文中和传统的以太网报文类似，拥有源目MAC，源目IP等元组。

VXLAN的报文格式（封装与解封装，UDP header、VXLAN header）；



Part8 SDN开源项目

8.1 OVS

OVS的组成结构

8.2 OpenStack

Neutron概述