

Software Defined Networks

软件定义网络

SDN历史发展

- 2006年，美国GENI项目资助了斯坦福大学Clean Slate课题。Clean Slate项目的最终目的是要重新发明英特网，旨在改变设计已略显不合时宜，且难以进化发展的现有网络基础架构。
- 2007年，斯坦福大学的学生Martin Casado 领导了一个关于网络安全与管理的项目Ethane，该项目试图通过一个集中式的控制器，让网络管理员可以方便地定义基于网络流的安全控制策略，并将这些安全策略应用到各种网络设备中，从而实现对整个网络通讯的安全控制。
- 2008年，基于Ethane 及其前续项目Sane的启发， Nick McKeown 教授等人提出了OpenFlow 的概念，并于当年在ACM SIGCOMM 发表了题为《OpenFlow: Enabling Innovation in Campus Networks》的论文，首次详细地介绍了OpenFlow 的概念。该篇论文除了阐述OpenFlow 的工作原理外，还列举了OpenFlow 几大应用场景。
- 基于OpenFlow 为网络带来的可编程的特性， Nick McKeown教授和他的团队进一步提出了SDN（Software Defined Network，软件定义网络）的概念。2009年，SDN 概念入围Technology Review年度十大前沿技术，自此获得了学术界和工业界的广泛认可和大力支持。
- 2009年12月，OpenFlow规范发布了具有里程碑意义的可用于商业化产品的1.0版本。

SDN的概念

软件定义网络是一种动态的、可管理的、经济有效的和适应性强的新型体系结构，非常适合现在具有高带宽动态特性的应用。

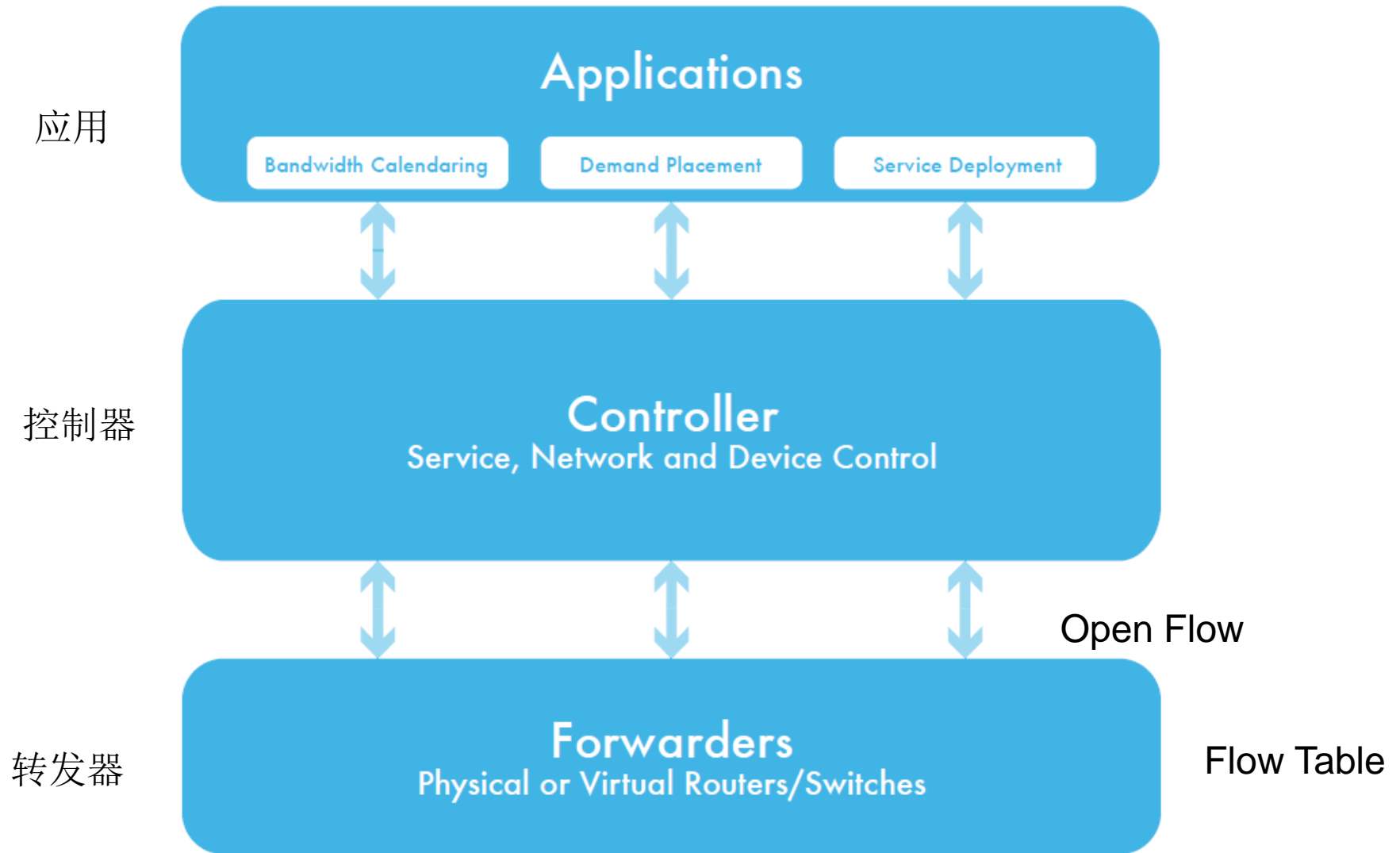
SDN体系结构解耦了网络控制功能和转发功能，使直接编程网络来控制网络应用服务的底层基础架构成为可能。

SDN优势

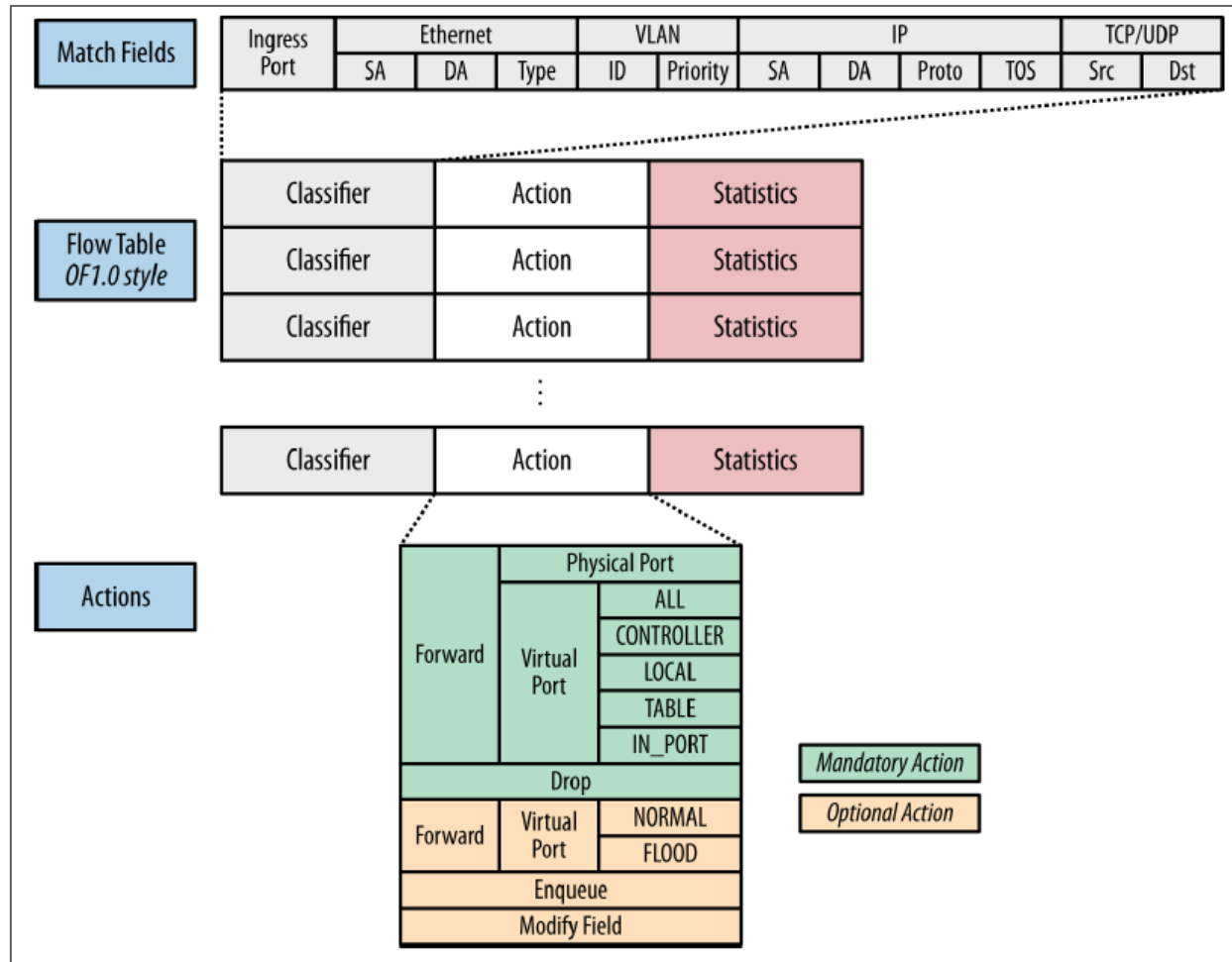
- 传统网络是根据业务需求配置路由器和交换机等网络设备。对于瞬息万变的因特网，传统网络的高稳定与高性能还不足以满足业务需求，灵活性和敏捷性反而更为关键。
- **SDN**所做的是将网络设备上的控制权分离出来，由集中的控制器管理，无须依赖底层网络设备，屏蔽了来自底层网络设备的差异。而控制权是完全开放的，用户可以自定义任何想实现的网络路由和传输规则策略，从而更加灵活和智能。
- 进行**SDN**改造后，无需对网络中每个节点的路由器反复进行配置，网络中的设备本身就是自动化连通的。只需要在使用时定义好简单的网络规则即可。如果你不喜欢路由器自身内置的协议，可以通过编程的方式对其进行修改，以实现更好的数据交换性能。

- 假如网络中有SIP、FTP、流媒体几种业务，网络的总带宽是一定的，那么如果某个时刻流媒体业务需要更多的带宽和流量，在传统网络中很难处理，在SDN改造后的网络中这很容易实现，SDN可以将流量整形、规整，临时让流媒体的“管道”更粗一些，让流媒体的带宽更大些，甚至关闭SIP和FTP的“管道”，待流媒体需求减少时再恢复原先的带宽占比。
- 正是因为这种业务逻辑的开放性，使得网络作为“管道”的发展空间变为无限可能。如果未来云计算的业务应用模型可以简化为“云—管—端”，那么SDN就是“管”这一环的重要技术支撑。

SDN体系结构



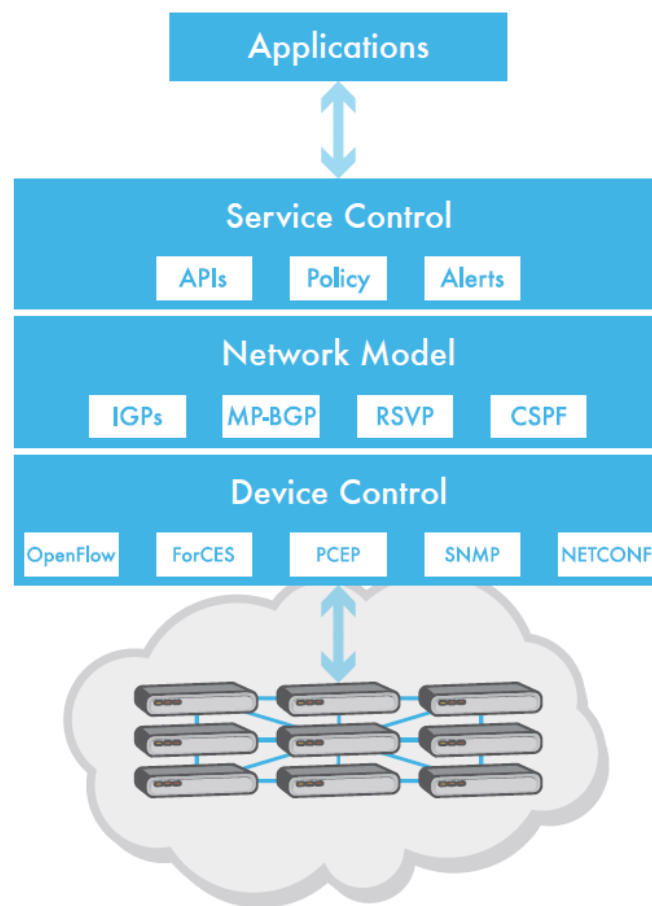
转发器 (Forwarder)



Flow Table

控制器 (Controller)

- SDN架构的主要组件是控制器。控制器把它的北向API提供给应用程序，使编程、重组和重构应用程序网络非常方便
- 控制器可以使用 OpenFlow、ForCES、PCEP5、NETCONF6、SNMP7或者专有的机制为转发器编程。
- 该控制器跟踪所有应用程序请求，维护网络拓扑和流量负载模型，使用这个模型为应用程序计算路径，编程转发器。
- 右图是一个典型的控制器结构。



Controller Architecture

应用

(Application)

- 现在最突出的**SDN**应用是网络虚拟化。许多现代应用正在构建为云应用，使用户可以不管它们的位置或它们正在使用的设备去访问它们。云是一个运行了很多虚拟机(**Virtual Machine**)的一组服务器。
- 随着应用的用户群的扩大（或缩小），需要添加（或删除）虚拟机。当用户群的地理位置发生变化时，虚拟机会自动迁移以便更好地服务于用户群。
- 虚拟机迁移还可以帮助在停电或自然灾害，如飓风和海啸提供业务连续性。增加、删除和迁移虚拟机都可以编程完成。然而，网络需要重构来反映这些变化。通过人工操作实现这些重构通常是一个冗长的和容易出错的任务。网络虚拟化可以使这些重构由**VM**管理系统进行编程实现。