# Abstract

摘要——由于其简单、透明和性能，

FlowVisor 在当今用于 OpenFlow 网络的虚拟化工具中囊括了一切。我们认为，这种有效性是以对同时使用的不同 OpenFlow 版本的不容忍以及有限的交换机功能为代价的。更重要的是，基于 FlowVisor 的管理框架无法运行和配置网络控制器。在本文中，我们提出了一个集成的 OpenFlow 虚拟化框架，它能够 (i) 运行和管理具有不同转发能力和 OpenFlow 版本的 OpenFlow 交换机的多个实例，(ii) 运行和配置专为控制虚拟机而设计的完整控制器或网络应用程序。建议框架管理下的网络，以及 (iii) 在网络中配置 QoS。除了介绍架构外，我们还在原型系统中共享实现和部署细节。

索引词——OpenFlow、虚拟化、FlowVisor、QoS、配置、网络管理

# Introduction

当“OpenFlow”[1] 和“virtualization”[2] 这两个词在一些文本中出现在彼此的附近时，“FlowVisor”这个词也可以被预测到，并且有很小的错觉。事实上 FlowVisor——用作 OpenFlow 交换机和控制器之间的中间通信层——提供快速、简单和透明的给定网络分解为虚拟切片，并将切片特定消息委托给指定的 OpenFlow 控制器。 FlowVisor 概念的成功体现在它在全球 OpenFlow 测试平台中的广泛使用，包括 GENI [3] 和 Ofelia [4]。然而，FlowVisor 的有效性是有代价的

网络中交换机功能的不灵活性和局限性。这些限制是三重的。首先，为交换机添加功能扩展——例如OAM 功能、非正统的转发机制、伪线或新的匹配结构——是非常有问题的，因为交换机和 OpenFlow 协议都必须更新。由于 FlowVisor 受限于 OpenFlow 协议版本，对于每个协议更新，FlowVisor 也必须更新。其次，在一个网络中，可以有多种OpenFlow交换机和控制器同时使用不同版本的OpenFlow协议1。 FlowVisor 无法处理此类情况，因为它本质上绑定到单个版本。最后，现有的基于 FlowVisor 的虚拟化框架仅提供对数据平面和 FlowVisor 的访问。例如，权宜之计类型 [5] 的管理工具，就像 Ofelia [4] 中使用的那样，控制平面和网络控制器必须由基础设施的用户管理。这种方法极大地阻碍了集成管理框架的实施。为了消除基于 FlowVisor 的方法和其他现有管理框架的缺点，我们最近提出了一种新颖的 OpenFlow 虚拟化架构，并准备了一个实现其关键元素的概念验证原型 [6]。我们提出的架构为在同一网络和同一交换机上运行不同版本的协议提供了解决方案。我们还提出了额外的功能来运行、停止和配置适当的控制器，以进一步简化网络的整体管理。由于篇幅限制 [6] 仅包含我们架构的高级描述，在这里我们分享有关实现和原型的详细信息。此外，该架构通过 OpenFlow 中当前可用的 QoS 配置功能进行了扩展，我们简要讨论了该框架未来应用所需的 QoS 功能。必须解决 QoS 供应问题才能获得

适合运营商级网络和服务提供商的架构，因此我们在这里的 QoS 相关贡献特别重要。分组交换网络中的 QoS 供应历史悠久。自 90 年代初定义基本 QoS 要求以来，大量研究考虑了分组网络中的 QoS 问题，并提出了大量机制和架构来阐述令人满意的解决方案。鉴于过去二十年产生的 QoS 相关文献的数量，人们可能会惊讶于 QoS 配置仍然是当今分组网络中一个未解决的问题。 QoS 成为这样一个长期问题的现象背后的原因肯定是复杂的。然而，我们认为网络设备的封闭接口对将 QoS 提供变成一个永久性的经典问题做出了重大贡献。由于 OpenFlow 可以支持 QoS，我们相信其开放式架构与集成管理框架相结合，可以激活和激励研究社区从实际角度重新审视 QoS 并制定功能系统，类似于 BitTorrent 设法在应用层实现多播。本文的其余部分安排如下。在第 II 节中，总结了 OpenFlow 相关的网络管理方法，并给出了有关 OpenFlow 中 QoS 支持的最新技术。在第三节中介绍了所提出的架构，然后在第四节中介绍了我们的数据模型的详细信息。最后，我们的原型系统在第五部分介绍了许多部署细节，第六部分总结了论文。

# Background

## A. OpenFlow 中的网络管理

网络管理是大规模网络不可或缺的组成部分，但是，OpenFlow 缺乏定义明确的管理框架。最近，提出了一种新协议，即 OF-Config 协议 [7]，以便向标准管理框架迈出第一步。 OF-Config 是支持 OpenFlow 的设备的配置和管理协议。目前它基于Netconf [8]、[9] 网络管理协议，因为它比传统的SNMP 更适合配置管理。该提案还使用 Yang [10] 数据定义语言给出了 OpenFlow 组件的数据模型。尽管 OpenFlow 中缺乏标准的管理工具，但近年来在美国和欧洲也建立了几个支持 OpenFlow 的测试平台。这些设施为专注于数据平面和 FlowVisor [2] 配置的实验者提供了特定的控制和管理接口。例如，Expedient [11] 是 Ofelia 当前使用的配置工具，之前由 GENI 使用。现在GENI 中的网络配置基于FOAM [12]、[13]。 OpenFlow 的另一个重要特性是提供网络虚拟化的能力。目前，FlowVisor 是提供轻量级网络虚拟化或网络切片的广泛使用的组件。当然，管理工具也必须能够配置和管理该组件的操作。 Expedient 的一个广泛使用的配套工具是 Opt-In Manager [14]，用于管理有关用户拥有的流空间和运行实验的信息。通过将流空间信息下推到 FlowVisor 来请求对网络资源进行切片。

## B. OpenFlow 中的服务质量

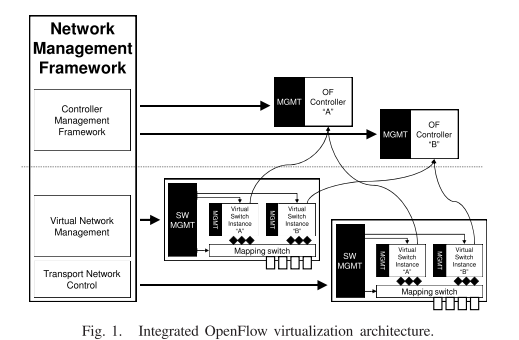
OpenFlow 社区通过用一些 QoS 功能丰富 1.0 版本的协议来意识到服务质量支持的重要性 [15]。然而，即使是最新的 1.3 版本，通过简单的排队机制也只有有限的 QoS 支持 [16]。在 1.0 版本中，可以通过可选的 enqueue 动作将数据包转发到输出端口的队列，而在 1.2 版本中，该动作被转换为可选的 set-queue 动作。控制器可以通过 OpenFlow 协议查询各种队列统计信息和队列配置参数，但该协议不支持创建队列或改变现有队列的行为。因此，队列配置需要在 OpenFlow 领域之外进行。虽然 OpenFlow 协议不支持创建

或修改队列，可以查询支持 OpenFlow 的交换机以报告附加到特定端口的队列，以及与队列关联的保证最小速率。此外，在 1.2 版本之后，它还可以向控制器报告最大速率。最小和最大速率当然不足以构建一个中等复杂的 QoS 场景，但流量队列可以具有其他无法查询的特性。例如，OpenFlow 控制器无法区分最小阈值为零的小型 RED 队列和导致缓冲区膨胀的巨大 FIFO 队列。而不是将交换机配置指令添加到

OpenFlow 协议，开放网络基金会创建了一个名为 OF-Config [7] 的辅助协议。它的第一个版本支持配置支持 OpenFlow v1.2 的交换机。2 从 QoS 的角度来看，它只能用于远程设置队列的最小保证和最大速率。有趣的是，OF-Config 应该用于配置队列，而 OpenFlow 应该用于配置仪表表。

# ARCHITECTURE

为了处理第一部分中列出的所有缺失部分，我们定义了一个集成的 OpenFlow 虚拟化框架。该架构如图 1 所示。它基于数据平面元素的灵活虚拟化能力。这些元素能够运行 OpenFlow 交换机的多个实例和版本，并为这些交换机实例提供专用上下文。它们还包括实现交换机物理接口和属于不同上下文的虚拟接口之间的映射的交换机逻辑。这一较低层，数据平面元素的虚拟化实体可能属于基础设施提供商，并且能够将虚拟网络映射到物理网络。基础设施用户（例如服务提供商）获得虚拟网络，该网络是物理网络的可配置视图。与 IEEE 802.1Q 虚拟 LAN 类似，我们的虚拟化层支持两种类型的映射：独占映射实现物理和虚拟接口之间的一对一关系，同时它还允许将多个虚拟端口映射到公共物理中继.在后一种情况下，必须向所有数据包添加虚拟网络的附加链路特定标识符，例如 VLAN 标记或 MPLS 标签，以区分属于不同虚拟实例的流量。



提议的集成架构实现了用于管理这些具有增强虚拟化能力的交换机的基本管理和控制功能。一方面，网络管理系统配置交换机：实例化虚拟交换机实例，启动/暂停/停止它们，将物理端口分配给虚拟端口，并且可以选择下载新版本的OpenFlow虚拟交换机。另一方面，它还维护受管网络的物理拓扑。例如，它可以重新配置物理到虚拟端口分配作为物理故障的响应。需要注意的是，这种重新配置不会对虚拟交换机的内部配置产生任何影响。该框架还提供了运行完整控制器或网络应用程序的方法，这些控制器或网络应用程序旨在在提议的框架的管理下控制虚拟网络。此类方法包括（1）加载和启动先前下载的第三方控制器，（2）配置先前实例化的虚拟交换机以连接到刚刚启动的控制器，（3）监控正在运行的控制器的不同特性，最后但并非最不重要的（4） ) 根据需要维护/停止/重启控制器。一个自然的扩展是不仅为运行控制器提供了虚拟节点操作系统，而且整个网络操作系统都由框架实现。这允许仅实现网络应用程序而不是完整的控制器；这反过来又是软件定义网络的推动者。与控制器配置框架类似，

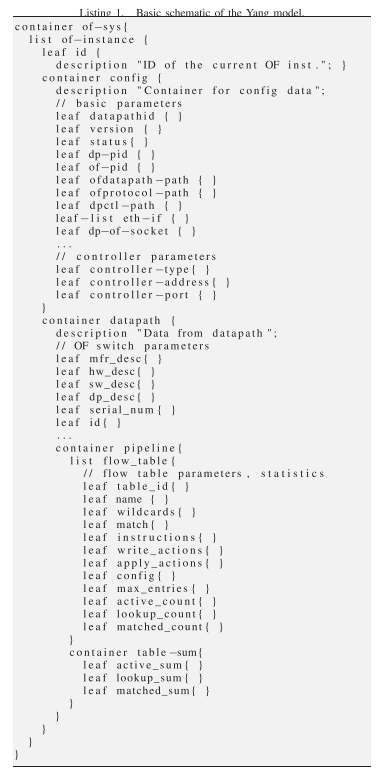
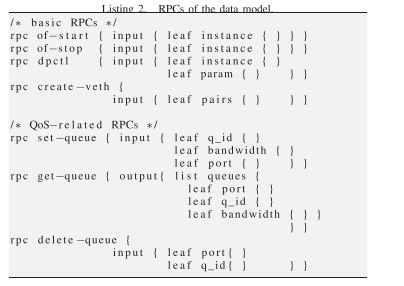
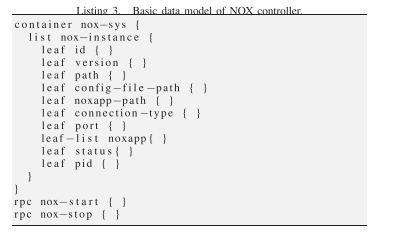
像 FlowVisors 这样的轻量级网络虚拟化方法也可以用这个框架来管理。 QoS 供应是虚拟化网络环境中的一个多阶段过程。基础设施用户首先需要拥有具有 QoS 保证的资源，以便能够将它们进一步共享给其用户。从建筑的角度来看，有两个极端。在集中式情况下，基础设施用户将每个 QoS 请求转发给基础设施提供商，并将收到的资源专用于最终用户。在分布式情况下，基础设施用户使用长期的 QoS 保证资源进行初始化，他们负责处理最终用户的本地 QoS 请求并与基础设施提供商重新协商保证资源。集中式案例可以轻松适应分层数据包调度器（例如，[17]、[18]）的丰富文献的结果，而在分布式案例中，现有的 OpenFlow 系统可以更容易地移植到虚拟化环境中。

# DATA MODELS

我们框架的管理接口基于 Netconf [8]、[9] 协议，我们使用 Yang [10] 作为数据建模语言。 本节专门介绍我们为相关网络元素设计的数据模型的一些细节。

A. OpenFlow 交换机的 Yang 模型

数据模型的结构可以分解为三个主要部分：配置和当前状态数据，来自 OF 交换机实例的信息（例如描述，如制造商，或流表统计），远程过程调用（例如启动/停止实例，创建虚拟接口）。清单 1 显示了数据模型的基本示意图，而清单 2 显示了 RPC。顶级容器处理各种 OpenFlow 参考交换机实例。每个实例都可以通过它们的 ID 进行区分，并且它有两个容器。一个包含成功执行所需的关键配置参数，而另一个只有在交换机能够与 Netconf 守护程序通信时才重要。 config 容器保存着交换机实例的基本参数，如二进制文件的路径（ofdatapath、ofprotocol、dpctl）、对应控制器的参数和接口信息（eth-if Leaf-list）。数据模型也能够处理虚拟以太网接口。此外，容器保存有关交换机实例状态的信息（例如，PID 值、状态）。数据路径容器能够通过直接查询 OpenFlow 交换机来收集统计数据。此功能需要支持与 Netconf 服务器通信的交换机扩展（例如，通过 Unix 域套接字）。可查询参数包括流表的几个描述和统计。统计信息保存在一个列表中，该列表旨在具有动态大小，以便只能查询活动表。数据模型有四个基本的 RPC，即 start、stop、dpctl 和 create-veth，如清单 2 所示。前两个处理 OpenFlow 实例的启动和停止，而 dpctl 是一个通用的手动控制器RPC 直接调用给定 dpctl-path 上的 dpctl。最后，create-veth RPC 创建可用于交换机实例虚拟化的虚拟接口对。 QoS 队列的管理是通过使用三个 RPC 来实现的，更具体地说，获取、设置和删除。队列由 ID 和端口号的组合标识。唯一可用的 QoS 参数是带宽限制。 Config true list 不适合处理队列参数，因为它将值存储在 Netconf 守护进程中，而 Netconf 服务器无法检测队列重新配置。但是，由于 Yuma toolkit3 中的错误，config false list 也不适合。作为解决方法，我们创建了一系列 RPC。 get-queue RPC 列出了所有配置的队列，这些队列直接从交换机查询。 RPC 本身首先确定队列的数量，然后创建所需数量的元素并从交换机查询它们的值。

B. NOX 控制器的 Yang 模型 NOX 控制器的 Yang 数据模型在

清单 3. 它与 OpenFlow 参考交换机的数据模型有相似之处，例如处理多个实例、启动和停止每个实例。

低参数是必不可少的：路径、连接类型（例如，tcp、ssl）、侦听端口和至少一个 NOX 应用程序。 由于 NOX 版本和 OpenFlow 协议版本之间的兼容性，路径被分为三个不同的路径变量（二进制路径、配置文件路径和 NOX 应用程序路径）。 该模型不是一个简单的叶子，而是为 NOX 应用程序定义一个叶子列表，以便同时运行它们。 还添加了状态和 PID 叶以进行监控。

C. 与OF-Config 的比较 我们的管理框架的数据模型和官方的OF-Config 协议有几个相似之处。两种方案都支持逻辑交换机的虚拟化，并提供统计数据的查询4。主要区别在于OF-Config 1.1支持更广泛的可查询统计数据，并且还支持配置数据（例如流条目）的设置和查询。另一个区别是虚拟化的实现方式。尽管这两种解决方案都允许创建逻辑交换机，但我们的实现在处理不同版本的 OpenFlow 交换机方面更加灵活。 OF-Config 以统一的方式定义每个资源，然后在逻辑交换机之间分配它。这会导致交换机之间的版本依赖性，因为每个交换机处理其资源的方式略有不同。我们的解决方案完全独立于版本，不同交换机之间共享的唯一资源是物理和虚拟接口。该框架不仅允许使用不同版本的 OpenFlow 协议，还允许使用定制的交换机代码。

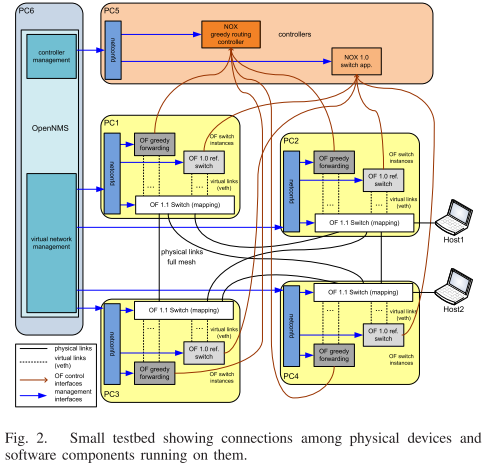
# PROTOTYPE IMPLEMENTATION

我们准备了概念原型制作的证明

使用专注于所呈现架构的关键元素的开源组件。源代码作为开源发布，可以公开获得[19]。我们选择 OpenNMS [20] 作为管理框架，因为它是开源的、高度灵活并且可以轻松地以模块化方式扩展。添加了两个新颖的组件，即虚拟网络管理模块和控制器管理模块。一方面，虚拟网管组件负责通过Netconf协议（以Yang为数据建模语言）配置虚拟化层映射交换机，建立虚拟拓扑。目前，不同虚拟实例的报文之间的区别是基于相应流表项管理的 VLAN 标签。此外，可以在基础设施用户层实例化具有不同转发能力的虚拟交换机并配置给定参数（例如控制器地址），例如 OpenFlow v1.0 和 v1.1 参考交换机、Open vSwitch [21]、OpenFlow switch with greedy转发 [22]，以及使用基于布隆过滤器的转发打开 vSwitch。该组件还负责创建虚拟以太网对并将虚拟交换机与映射交换机连接起来。另一方面，在控制器管理模块中实现了基本配置管理功能，用于（重新）启动和停止 NOX 控制器实例作为演示。值得注意的是，该框架不仅限于NOX，其他控制器（如Beacon、Maestro、Trema等）也可以通过这种方式进行管理和配置。 通过这种方式，基础设施用户可以上传和运行自己的控制器管理框架控制下的应用程序或服务。此外，基础设施提供商可以提供可供不同基础设施用户使用的通用控制器应用程序。我们原型的 Netconf 管理接口基于开源 Yuma 工具 [23]，分别提供 Netconf 服务器（netconfd）和客户端（yangcli）。我们将客户端组件与 OpenNMS 集成，并向交换机和控制器添加了代理。原型系统目前在每台 PC 上运行一个 Netconf 代理，运行在 PC 上的逻辑网络组件通过一个通用代理进行配置。我们通过在图 2 所示的拓扑结构上运行实验来验证我们的原型框架的功能。数据平面由四台 PC (PC1-4) 组成，运行映射器交换机、贪婪转发交换机和 OpenFlow 1.0 参考交换机， 分别。这些 PC 以全网状物理连接。贪婪域的 NOX 控制器和标准的 MAC 学习 NOX 应用程序运行在 PC5 上，而 OpenNMS 托管在 PC6 上。我们使用单独的带外控制和管理网络。测试平台系统使用单个 Netconf 代理运行

然而，PC 对虚拟域的相应代理或 NMS 实例的扩展很简单。此外，我们已经将我们的原型移植到 OpenWRT 固件中，也可以使用物理设备代替运行交换机实例的 PC。值得注意的是，我们在用户空间运行的 TP-Link TL-WR1043ND 设备（或 PC）上使用软件开关，这些硬件设备具有低性能的 CPU。当然，运行多个交换机实例会导致性能下降，但是，我们发现即使是这些低性能设备也能够在不显着下降的情况下运行6。由于我们正在开发概念验证原型，因此在此阶段我们不关注性能。我们进行实验是为了验证网络和组件的运行。这个小型测试平台能够创建和管理

由不同类型的 OpenFlow 交换机组成的多个虚拟网络。更具体地说，我们已经对 OF v1.0 和 v1.1 交换机以及我们实现贪婪转发机制的自定义交换机进行了多次实验。可以从我们基于 OpenNMS 的管理系统轻松配置网络。在我们的场景中，我们通过相同的物理网络发送了不同的视频流，但使用了不同的虚拟拓扑。实验验证了框架在所有情况下的行为。作为配置机制的说明，创建作为配置机制的说明，创建具有 QoS 保证的队列会生成以下事件链。 首先，网络运营商在 OpenNMS 的 QoS 模块中设置具有最小保证速率的队列的参数。 然后 OpenNMS 向目标节点的 Netconf 服务器发送一条 set-queue Netconf 消息。 Netconf 服务器将此消息转换为 dpctl 命令。 收到此 dpctl 命令后，OpenFlow 交换机进一步将配置请求映射到流量控制 (tc) 命令，最终在内核中创建正确的队列。 （虽然 Netconf 服务器可以直接发出 tc 命令，但是需要通知软件交换机有关新/更改队列的信息。）



# CONCLUSION AND FUTURE WORK

在本文中，我们介绍了一个集成的 OpenFlow 网络虚拟化和管理框架，它可以适应 OpenFlow 版本的多样性，可以配置除数据路径之外的 OpenFlow 控制器并提供对 QoS 的支持。我们还分享了我们在系统实施和部署方面的经验。到这个时候，原型系统已经能够执行基本的管理任务，未来有很多改进的可能性。未来的一个方向是使用额外的 QoS 功能扩展框架。虽然，似乎开放网络基金会在每个新版本中不断增强 OpenFlow 协议及其辅助 OF-Config 协议中的 QoS 支持。然而，到目前为止，它只专注于提供流量速度保证。目前不可能提供到例如具有优先排队的 VoIP 流量的低延迟连接。同样，协议不允许配置或查询活动队列管理属性，例如，避免由于 TCP 同步导致的大量数据包丢失。如果我们支持这些 QoS 技术，那么即使没有直接协议支持，基础设施用户及其 OpenFlow 应用程序也将从新的 QoS 保证中受益。总的来说，具有复杂 QoS 功能的 OpenFlow QoS 架构是一个具有挑战性的研究方向。我们在该领域的未来工作计划在一篇配套论文中进行了总结[24]。我们还解决了具有拓扑可视化功能的管理系统的扩展问题。这个新模块将分别包括物理和虚拟拓扑、链路状态和交通信息的可视化。