Computer Science and Application 计算机科学与应用, 2019, 9(2), 359-374
Published Online February 2019 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2019.92041

Hans汉斯

Design and Implementation of Virtual Machine Network Access Control System Based on Software-Defined Networking

Xiaojun Zhou, Xiangquan Shi

Computer College, National University of Defense Technology, Changsha Hunan
Email: 559582266@qq.com

Received: Jan. 31st, 2019; accepted: Feb. 12th, 2019; published: Feb. 20th, 2019

Abstract

Through analyzing the traditional network and the SDN network access control technology based on OpenFlow protocol, a custom virtual machine users' access control protocol based on the SDN network is designed, to realize the control of virtual machine user's operation of accessing a network. In this paper, the working principle and implementation method of the custom access control protocol are designed and analyzed in detail. A custom access control protocol's authentication server module and client module are designed and implemented, and the authentication process of virtual machine user access control in network is described in detail. Message listener module, message processing module and information certification module of the authentication server module, as well as user authentication information acquiring module in the client authentication module, special message sending and receiving process module, authentication information sending and acknowledging module are designed and implemented.

Keywords

SDN, Controller, Access Control, Network Authentication

基于SDN的虚拟机网络访问控制系统设计与实现

周小俊，时向泉

国防科技大学计算机学院，湖南 长沙
Email: 559582266@qq.com

文章引用: 周小俊, 时向泉. 基于 SDN 的虚拟机网络访问控制系统设计与实现[J]. 计算机科学与应用, 2019, 9(2): 359-374. DOI: 10.12677/csa.2019.92041

收稿日期：2019年1月31日；录用日期：2019年2月12日；发布日期：2019年2月20日

摘　要

本文在分析了传统网络和基于OpenFlow协议的SDN网络中的访问控制技术的基础上，设计出了一种基于SDN网络的自定义虚拟机用户入网控制协议，来实现对虚拟机用户的接入网络的操作进行访问控制。本文对自定义访问控制协议的工作原理和实现方法进行了详细的设计分析。针对自定义访问控制协议中的认证服务器模块和客户端模块进行了设计与实现，并详细描述了网络中虚拟机用户入网访问控制的认证过程。对认证服务器模块中的消息监听模块，消息处理模块和信息认证模块进行了详细的设计与实现，对认证客户端模块中的获取用户认证信息模块，特殊报文的发送和接收处理模块，认证信息的发送与确认模块进行了设计和实现。

关键词

软件定义网络，控制器，访问控制，入网认证

1. 绪论

1.1. 引言

计算机网络的普及程度已经成为衡量社会发展及进步的重要指标。随着人们对于网络应用要求的逐步提高，网络规模的不断扩大，各种新的业务需求对于网络的性能要求越来越高，网络技术在发展和实践中不断创新。虚拟化技术的出现可以说是网络发展的历史性突破，也是当今信息时代的一次重大变革，为企业的发展带来了新的转机。由于网络设备的固定化，传统网络设备内部复杂，协议过于复杂繁多，开拓和创新的网络设备需要花费很大的人力和精力。网络研究者在现有网络架构部署上很难再有新的突破，网络的进一步发展遇到了阻碍。对于一种新的网络架构的需求已经迫在眉睫，控制与转发分离的网络架构被适时的提了出来。

同时，新型网络架构下的网络安全技术也应运而生，其中网络访问控制技术是网络安全领域的重要技术手段，在新型网络架构下，新型的网络访问技术也需要新的改变和新的技术。

1.2. 研究背景和意义

网络安全是自网络出现以来一直探讨和研究的问题，网络安全是为了保障数据的安全和核心数据的合法使用，尤其是随着网络的普及使用，网络安全对于用户个人以及企业来说都非常重要。网络安全只是一个相对的概念，并非绝对的。可以说是没有绝对的网络安全，因为网络受到种种因素的限制，网络安全不可能做到任何形式下的绝对安全。非虚拟环境下，操作系统都是通过硬件来操作运行的，操作系统都是运行在硬件基础之上，对于硬件资源的分配和管理都是由操作系统来管理和实现的，一个操作系统占据了对整个机器的资源。而虚拟环境中则不同，操作系统并非直接运行在硬件上，且对于资源的使用和管理都是由虚拟机管理器来统一管理和调度的，同时在虚拟环境中，一个虚拟机管理器中可以同时

运行多个虚拟机，对于底层硬件的资源使用各个虚拟机之间是相互共享的。随着虚拟化技术的不断应用，虚拟化技术所带来的安全问题已经不容忽视，安全认证和访问控制技术，是保障网络安全的一项重要技术，在传统网络中已经广泛应用并取得了一定的成效[1]。传统的访问控制机制都是建立在对传统网络的构建基础之上，大都是基于对物理端口的访问控制，在虚拟化环境下一个物理端口与多个虚拟化端口相接，传统的访问控制技术无法实现在虚拟化技术对每个端口的精细控制，给虚拟化环境下的网络安全带来很大隐患。在 SDN 网络中数据转发是基于流表匹配规则而进行的，不受实际物理端口的影响，即使在虚拟环境下也能够对虚拟网络中的用户进行入网控制。

目前，传统的认证系统主要有以下三种：即 PPPoE 认证[2]、802.1X 认证[3]和 Web 认证[4]。PPPoE 认证是基于 PPP 协议的认证，在 PPP 协议中有一项重要功能就是提供了对身份的验证。PPPoE 认证是当前较为常用的一种认证方式，如现在比较常用的 ADSL 宽带接入就使用了 PPPoE 认证，PPPoE 在认证过程中需要封装一层 PPP 协议，不仅增加了网络的费用而且减缓了传输速率；802.1x 认证在网络认证体系中应用比较热门，在局域网中应用较多，它是基于客户端和服务器的一种认证方式，它可以限制用户和设备对网络的接入，所有用户和设备想要获得和使用交换机以及局域网中的资源必须要通过验证，802.1x 认证的缺陷是无法跨越 3 层网络，从而达不到统一的认证管理；Web 认证的方法与其他认证有所不同，首先需要给用户分配一个地址，在用户访问网络时使用，用户在访问相关门户的网站时需要输入验证信息，如用户名和密码等，验证信息通过 Radius 客户端发送给认证服务器进行验证，在认证通过后，客户端将被触发新的地址分配请求，用户将得到一个可以访问外网的地址。Web 认证在认证过程虽然不需要设立特定的客户端，但在认证过程中需要跨越多个交换机。由于认证方式特点，PPPoE 认证在本论文中不适用，Web 认证同样如此。802.1x 认证在 SDN 虽然有一定的应用，但 802.1x 只能对物理端口的认证，而在虚拟网络中，由于物理端口可以连接多个虚拟交换机，802.1x 无法做到对每个虚拟交换机端口进行精细化控制。所以在本文根据现实需求，自定义了入网认证协议，期望能解决虚拟化云计算环境下传统的网络访问控制技术不能充分适用，需要开发适应虚拟化计算环境下的虚拟机访问控制技术的问题。

1.3. 本文的主要工作

本文的主要研究内容是：在虚拟化环境中采用 SDN 技术实现虚拟机用户入网的访问控制。

本文尝试通过自定义的网络访问控制协议，对虚拟机用户入网进行访问控制，解决虚拟化环境下缺乏有效技术手段对虚拟机用户网络访问进行细粒度控制的问题。首先，运用当前网络安全中的访问控制技术和安全认证技术，结合基于 OpenFlow 协议[5]的 SDN 网络架构的工作特点，提出一种自定义的访问控制协议的网络认证构架，对自定义协议中的使用到的相关报文进行特殊标识，成为虚拟机客户端到交换机专属的识别报文，从而限定虚拟机用户的网络访问限制。通过实验编码完成相关的报文的制定和认证过程的操作，并搭建相对应的实验平台对自定义协议实施的可行性进行测试和验证。

其次，对 Ryu 控制器进行访问控制应用模块开发，在控制器端开发具有认证服务功能的模块，实现认证服务与访问控制服务聚合，认证服务模块内置于 SDN 控制器之中，提高认证服务模块的运行效率。同时在控制器实现访问控制模块，在虚拟机网络初始化时，通过控制器下发相应的流表至交换机，设定交换机只能识别含有特殊匹配字段格式的报文，实现对虚拟机用户网络访问控制的初始化。

2. 相关技术研究

2.1. 网络访问控制技术

访问控制技术的提出最早源于 70 年代，起初的目的是为了解决对大型主机系统上共享数据的访问进行授权和有效管理[5]。随着计算机技术的不断发展，网络的应用在各个领域都占据了重要地位，信息系

统的安全性让访问控制技术的思想和方法得到了迅速的推广和应用。在这些年的发展过程中，访问控制技术得到了不断的完善，也先后出现了多种形式的访问控制技术，但它们的目的是一致的，都是为了防止未授权用户对网络的非法访问，同时也合法用户对系统资源的非法使用。访问控制通常以对用户的身份认证为基础，在身份认证的基础上实施各种访问控制策略从而达到对合法用户的使用规范和控制。在 ISO [6] (International Organization for Standardization)国际标准化组织的网络安全体系设计标准 ISO-7498-2 中定义了五大安全服务功能。即身份认证服务、访问控制服务、数据保密服务、数据完整性服务、不可否认服务。访问控制服务作为其中的五大服务之一，在整个网络安全体系中具有非常重要的作用，它不仅可以限制用户对关键资源的访问，而且通过对访问的限制防止非法用户的入侵，避免因合法用户在使用中使用不当和非法操作对系统造成破坏。

2.1.1. 相关概念

网络访问控制作为网络安全体系中必不可少的一个环节，是网络安全和防范的重要策略。有效的网络访问控制能对网络资源起到很好的保护作用，一方面使网络资源只对已授权的合法用户进行开放，合法用户在使用中之能按照授予的权限进行资源的访问和获取；另一方面通过访问控制权限来阻止非法用户的入侵，给网络资源和网络系统造成破坏。首先介绍一下网络访问控制的相关概念。

● 网络访问控制的主要目标和任务：顾名思义，网络访问控制的目标是为了保护网络资源合法访问和使用，使网络资源保持完整和安全；其主要任务是防止未经授权的访问窃取和破坏网络资源。

● 网络访问控制的客体：即需要访问的对象，通常也称为目标客体，主要是指网络中需保护的网络资源。如网络中的数据信息、重要的程序进程和各种网络服务等，这些都可以称之为客体。

● 网络访问控制的主体：通常指一个网络实体，可以主动对网络客体进行访问，网络主体可以是一个用户，也可以是一段程序。

● 网络访问控制的授权：是指网络主体访问网络客体之前，授予网络主体访问的权限，从而网络主体可以访问相应的网络资源。

2.1.2. 入网访问控制技术

入网访问控制是网络访问控制的重要手段和技术，也是网络访问控制的第一步，其主要目的是对用户的登录进行控制，有效的控制用户的入网方式和时间等。入网访问控制通常包括三个方面：即对用户的用户名进行识别和验证、用户的口令进行识别和验证、以及用户账户的缺省限制检查。所有用户在入网前必须要通过以上三个步骤，用户方可成功接入网络。对用户的用户名和口令验证是入网的第一道关卡，用户在入网时需要进行用户信息的注册，用户在登录时输入相应的用户名和密码，服务器会对用户所输入的验证信息进行验证。首先将对用户的用户名进行验证，只有用户名验证通过后，服务器才会进一步操作，对用户的口令进行验证；验证不通过服务器将拒绝用户的入网访问请求。在整个入网过程中，网络管理员能够对一般用户的账号使用情况进行管控，如用户访问网络的时间、方式等，而用户名以及用户的个人账号构成了计算机系统最基本的安全形式。系统管理员建立相应的用户账号，而用户的口令是用户访问网络的钥匙，是用户访问网络的关键，如同人的身份证件一样。当用户的用户名和口令全部验证通过后，服务器将会进行用户账号的缺省限制检查，网络不仅需要控制用户入网的站点，而且能够对用户的入网时间以及用户访问入网的工作站数量进行限制。此外，网络应该对所有用户的访问进行审查，如果用户在验证时口令多次输入错误，则判定该用户为非法入侵，应该给予用户相应的报警信息。

2.2. SDN 技术

2.2.1. SDN 基本架构

SDN 是针对传统网络的弊端设计时所提出的一个概念，同时也是一种全新的网络架构[7]。在传统网

络中，网络的传输都是基于网络设备，网络设备在设计时通常依赖市场所需，为了提高生产效率，通常采用集成的设计方式。交换机和控制器是传统网络所必备的设备，他们在设计时都采用了传统的大规模集成设计而成，在设备的性能上存在很大的局限，用户对于网络设备只限于操限性，无法对网络设备进行控制，这对于用户的使用和网络的发展带来很多弊端。与传统网络不同，SDN 网络实现了控制与转发的分离。在这种架构下，交换机只负责对数据进行转发，而对数据的控制主要交于控制层面负责，因此网络的整体效率不受网络设备性能的影响。且 SDN 最突出的优点是控制层面具有可编程性，使得整个的网络结构中，上层应用和底层的网络设备之间的通信，可以通过软件编程的形式来实现，网络将变得更加灵活和便捷。ONF [8]是一家致力于 SDN 研究的组织机构，长期以来一直从事对 SDN 的研究工作，ONF 可以说是当前所有从事 SDN 研究机构中最大的一个，ONF 所提出的 SDN 架构非常具有代表性，是目前所提出的 SDN 架构中比较经典的代表。在此阐述一下 ONF 所提出的 SDN 架构，具体如图 1 所示：

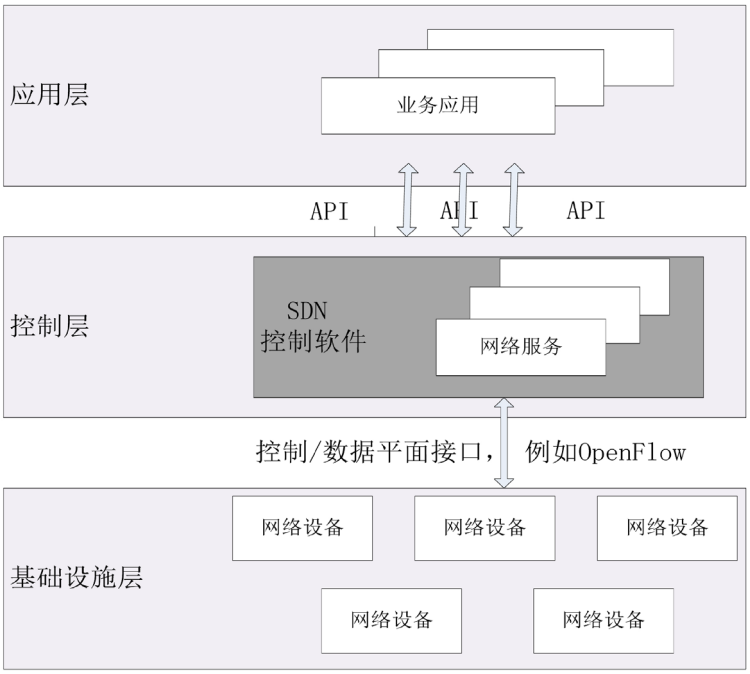


Figure 1. The basic architecture of SDN
图 1. SDN 基本架构

ONF 所提出的 SDN 经典架构如上图所示，它将 SDN 分成了三个层级结构，分别为应用层、控制层和基础设施层。

● 应用层：应用层位于最上一层，应用层存在不同的应用和业务，分布着大量的应用软件，可以通过对应用程序接口的调用，并结合现实业务的需求来实现相应的程序。如网络的运行情况、网络安全、负载均衡以及对网络性能的管理和监测等很多功能和服务都是通过软件应用程序的方式来体现的。

● 控制层：控制层位于中间层，控制器是控制层的主要构成部件，主要功能是实现对数据平面的资源进行合理的编排，查看并维护网络的拓扑信息等。在 SDN 网络中可以存有多个控制器，一个控制器可以连接多个设备并对设备进行控制。控制器在 SDN 网络中相当于一个平台，对下可以与 SDN 交换机进行会话，对上可以提供应用程序的编程接口。

● 基础设施层：最下层为基础设施层，基层设施层通常由 SDN 交换机和 OpenFlow 协议所组成，主要功能是负责数据的处理、转发和状态收集。

2.2.2. OpenFlow 交换机

作为 OpenFlow 网络的核心部件，OpenFlow 交换机是构成 OpenFlow 网络的重要组件之一[9]。OpenFlow 交换机通常由三个部分组成：即流表、安全通道和 OpenFlow 协议。OpenFlow 交换机与传统的交换机在功能有所区别，即 OpenFlow 交换机只负责对数据的转发。在本文中采用的是 OpenvSwitch 软件交换机，在此对该交换机进行介绍。

OpenvSwitch (OVS) [10]是目前 SDN 网络中应用较多的一款虚拟机，遵守 Apache2.0 许可证，它是基于软件来实现其功能的。OVS 的设计是根据 OpenFlow 而制定的，OVS 是一款开源的多平台化的虚拟交换机，支持 OpenFlow 协议，与控制器密切相连。如果控制器同样支持 OpenFlow 协议，用户可以在操作时使用该类型的控制器对 OVS 进行远程管控。顾名思义，虚拟交换机在进行数据交换时使用的是虚拟平台，其主要部件是通过软件的形式来构成的。OVS 虽然是虚拟交换机，与传统的物理交换机有所不同，但在工作原理上相类似。在具体实现中，虚拟机分别与计算机中的物理网卡和对应的虚拟网卡相连接，在进行数据转发时，主要通过虚拟机链路来实现，而虚拟机链路与数据所携带的 MAC 地址相对应。其具体工作原理如图 2 所示：

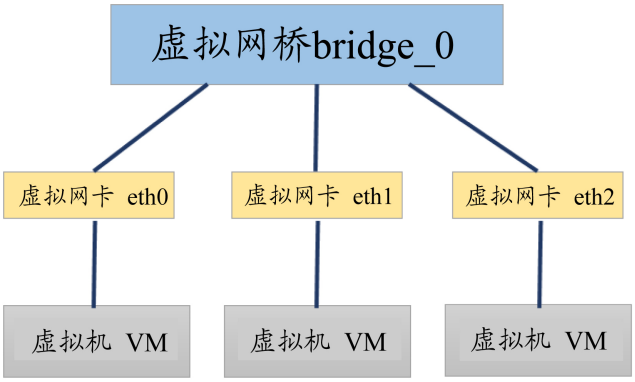


Figure 2. Virtual switch working schematics
图 2. 虚拟交换机工作原理图

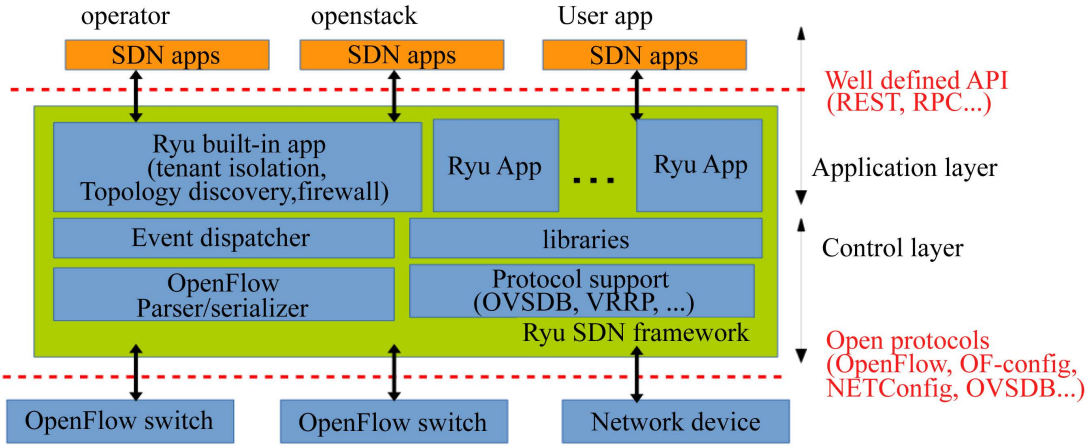
当数据包发出后，数据包首先要通过虚拟网卡，虚拟网卡是根据虚拟机而配置的，数据包通过虚拟网卡后，数据包将被转发至虚拟交换机。OVS 交换机是根据 OpenFlow 而设定的，OpenFlow 对其提供了支持能力，当数据包到达 OVS 交换机后，因为 OVS 交换机自身保存了相应的流表，所以首先会对数据包进行本地流表的查找匹配。如果发现相应的流表匹配，对数据包的处理将会依据流表所对应的指令来进行操作；如果没有相应的匹配项，OVS 交换机将会把数据包发送至控制器，由控制器进行处理，控制器将根据数据包的有效数据来制定相应的流表，并将流表下发至交换机。同样，数据包的转发可以通过物理网卡，因为 OVS 交换机的一端与物理网卡相连，如需通过物理网卡进行转发，则只需将数据包发送至物理网卡上，最终转发给外部与计算机物理网卡相连的网络设备上。

2.2.3. SDN 控制器

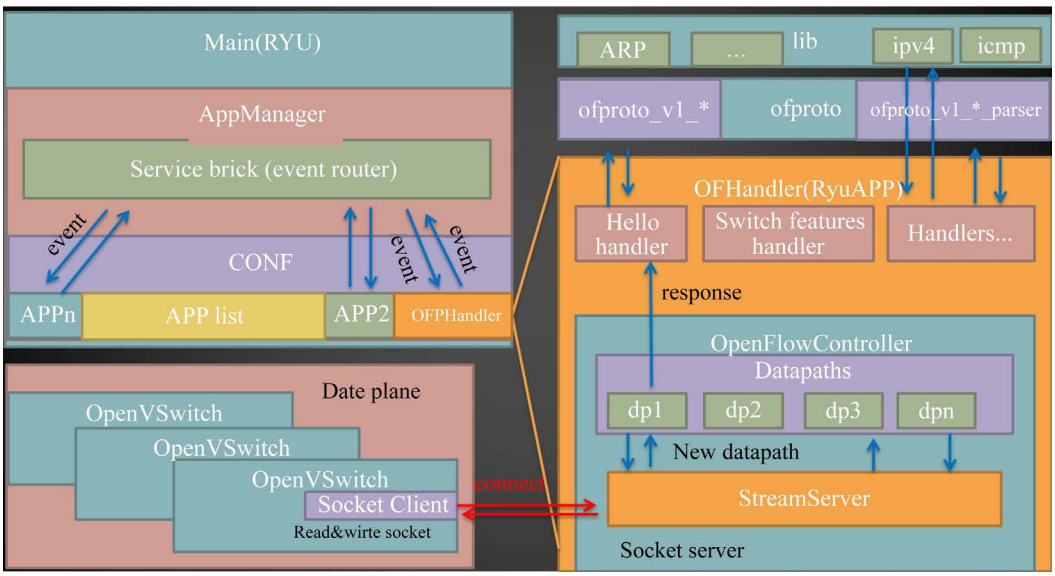
控制器是整个 SDN 网络架构中的核心组件，在 SDN 网络中，控制器也被称为一个小型的网络操作系统(Network Operations System NOS)，网络管理者可以通过控制器提供的编程接口来实现对网络的监控和管控。随着 OpenFlow 协议在 SDN 网络中日益成熟的应用，OpenFlow 控制器作的 SDN 网络的核心作用越来越明显，近两年来，OpenFlow 控制器的发展也是非常迅速，市场上出现的控制器的种类也是十分的多样，如大家最早开始使用的 NOX/POX 控制器，都很好的支持了 OpenFlow 协议，也成为开发其他控

制器的参考依据。在专业领域中广泛使用的控制器软件有 OpenDaylight 控制器，Floodlight 控制器，ONOS 控制器以及 Ryu 控制器，前三款都是基于 Java 语言进行开发设计并实现的，主要应用于商用领域；而 Ryu 控制器软件是基于 Python 语言开发的，主要应用在学术领域的研究工作中。本文采用了 Ryu 控制器，在此对 Ryu 控制器进行一些相关的介绍。

Ryu 是日本 NTT 公司主导开发的开源项目，其字面意思是日语中“Flow”的意思。Ryu 调度目标是提供一个拥有逻辑上集中控制能力的 SDN 操作系统，该系统具有设计完备的 API 接口，使网络应用者能够便捷的创建新的管理和控制应用。Ryu 用 python 语言编写，完全遵守 Apache 许可证，能够支持 v1.0、v1.2、v1.3 等多个版本的 openflow 协议。Ryu 架构和 SDN 架构非常契合。控制层主要提供控制能力，通过北向接口的 Rest API 为 SDN apps 提供服务，供 SDN apps 调度和控制流量和网络。通过南向接口的 Openflow 等协议控制 openflow 交换机，完成流量交互。其中 Ryu 控制层起到了承上启下的作用，是北向接口的控制和交换中枢。Ryu 的核心架构如下：



Ryu 内部核心模块如下图所示：



目前 Ryu 控制器主要实现的功能主要有：

1) 在网络启动之初，对网络中的一些基本构成进行发现和管理，如拓扑结构，设备，交换机和主机等；

2) 通过 OpenFlow 协议并利用安全通道机制实现控制器和交换机之间的消息通信；

3) 对交换机发送过来的 Packet-In 消息进行监听和处理，并向交换机下发流表，控制交换机中网络数据的流向；

4) 管理 App 模块，共享存储、线程、测试等资源；

5) 提供一个 web 管理界面和调试服务器，用来对整个网络中的信息进行监控和管理。

2.2.4. OpenFlow 协议

在网络中通信协议是网络之间传输的标准和规范，OpenFlow 协议同样也是一种标准，是基于 OpenFlow 控制器和 OpenFlow 交换机之间，二者在进行交互作用时所使用的信息的接口标准。其协议信息的集合是 OpenFlow 协议的核心。OpenFlow 协议主要支持以下三种类型的消息：即 controller-to-switch 消息、asynchronous 消息以及 symmetric 消息。其中每个消息由多个子消息所构成。具体对消息的描述如表 1 所示：

Table 1. Message list of OpenFlow protocol
表 1. OpenFlow 协议的消息列表

类型	名称	说明
Controller-to-switch	Features	在进行 TLS 会话时，控制器发送 Feature 请求消息至交换机，交换机应答相应的 Feature 信息，该信息必须包含交换机所支持的功能
	Configuration	控制器设置或查询交换机上配置的相关信息。交换机只需要应答查询消息
	Modify-state	控制器对流表项和交换机端口状态等进行管理
	Read-state	控制器对交换机的流表、端口以及流表项等统计信息进行收集
	Send-packet	控制器通过交换机向指定端口发数据包
	Barrier	控制器确保消息依赖已经被满足或收到完成操作的通知
asynchronous	Packet-in	交换机收到的数据包在流表中没有匹配项，或者所匹配的流表项中含有“转发到控制器”动作，则向控制器发送 Packet-in 消息。如果交换机缓存空间足够，数据包将被临时放在缓存中，数据包中的部分内容(默认 128 字节)和缓存中的序号一起封装在 Packet-in 消息中，并将消息发给控制器；如果交换机缓存没有多余空间，则将所有数据包封装在 Packet-in 消息中发给控制器
	Flow-removed	流表中的流表项应超时或修改等原因被删除时，向控制器发送 Flow-removed 消息
	Port-status	当交换机端口状态改变时，发送 Port-status 消息给控制器
	Error	交换机通过发送 Error 消息来告知控制器所发生的问题
symmetric	Hello	用于检测交换机和控制器之间连接
	Echo	交换机和控制器均可发送该消息，主要用来测量时延和 TCP 链接保持等

2.3. 本章小结

本章主要介绍了 OpenFlow 协议和 SDN 网络架构的特点，以及虚拟化云计算环境下传统的网络访问控制技术不能充分适用，对适应虚拟化计算环境下的虚拟机访问控制技术的现实需要，同时对虚拟化云计算等相关背景做了介绍。

3. 虚拟机网络访问控制系统设计

3.1. 系统总体架构设计

3.1.1. 基本架构

网络的安全访问是长期以来一直在探索和有待解决的问题，尤其随着 SDN 网络的不断发展，网络规模的不断扩大，对于用户入网访问的控制越来越难，随着网络应用的业务需求不断扩大和网络的结构和管理越来越复杂，对 SDN 网络中虚拟机入网访问控制进行细粒度的认证是我们所面临的一个重大难题。认证和访问控制技术，作为网络安全的重要技术手段，已经在传统网络中广泛运用。本文采用了使用自定义虚拟机入网认证的方法来实现对虚拟机用户的认证和访问控制。系统由三大部分组成：SDN 控制器(认证服务模块和访问控制模块)、虚拟交换机、运行于虚拟机中的认证客户端，如下图 3 所示：

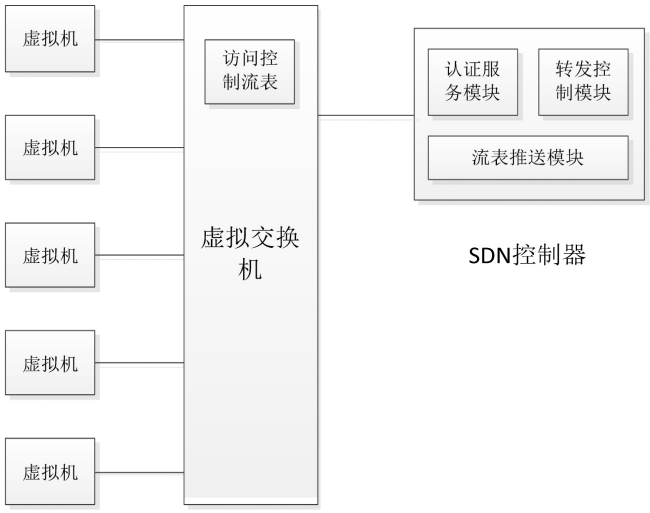


Figure 3. The overall architecture diagram
图 3. 整体架构图

SDN 控制器：1) 与交换机建立链接，控制器内的程序通过 OpenFlow 协议指导交换机进行转发；2) 通过 OpenFlow 协议指导交换机进行认证报文上报、认证结果返回；3) 接收到客户端认证报文，根据认证口令决定是否认证成功；4) 返回认证结果给客户端；5) 根据认证结果 SDN 控制器决定是否下发转发面报文。

系统中的交换机基于 OpenFlow 协议，具有很强的可编程性，负责整个网络的转发行为：1) 与控制器进行建链、握手，建立基于 TCP 协议(或 UDP 协议)的 OpenFlow 链路；2) 接收 OpenFlow 报文，建立转发面行为。

此外本系统中的交换机还负责安全认证的职责：1) 过滤认证报文，在认证的初始阶段滤除非认证报文，保证系统的安全；2) 上报认证报文给控制器，让控制器进行鉴权认证；3) 回送认证结果，控制器认证鉴权完毕后会发送认证结果给客户端，指导客户端的下一步工作。

客户端软件是在虚拟机中运行，是整个认证过程的发起者。虚拟机在云计算管理软件(如 OpenStack、AWS)启动后，如果需要接入网络，必须首先接入网络会向其接入交换机发送认证报文，接入交换机可能是服务器上的虚拟机交换机，如 OpenvSwitch，也可能是硬件交换机。

客户端发送完认证报文后会等待认证结果，如果认证成功，客户端可以进一步接入网络、如果认证失败，虚拟机发出的报文匹配不到接入交换机的流表，任何报文都无法进入网络。

认证鉴权控制面和转发行为控制面依赖关系，逻辑上，不言而喻，认证鉴权控制面是转发行为控制面的前提和基石，转发控制面的实现依赖于认证鉴权的结果；实现上，转发流表的生成依赖于认证鉴权的结果，通过认证鉴权才会下发转发流表，虚拟机才能接入网络。

解耦特性体现在三方面，控制器内认证鉴权功能模块和转发控制功能模块解耦；OpenFlow 交换机内，转发行为的流表和认证鉴权相关的流表分别位于不同的表空间；虚拟机内客户端认证软件与其他功能互相解耦，独立于其他功能。

通过对接入交换机的控制，从根本上切断了接入网络的可能性，换言之，从虚拟机内部发起的，或者针对虚拟机的网络攻击难度增大，成功率降低。

整个架构如下图 4 所示可分为三层:最下一层为虚拟机客户端，虚拟机用户通过客户端向与之相连的交换机提出认证请求。第二层为数据平面层，主要安装有虚拟交换机，负责网络流的转发工作，通过交换机中安装的匹配规则来实现虚拟机用户发送的特殊报文的解析和转发。第三层为控制面，主要对数据面的发的用户数据对用户信息的认证，并通过流表的转发来实现对用户入网的控制。其中控制器和认证服务通过 RPC 交互，达到模块间解耦。

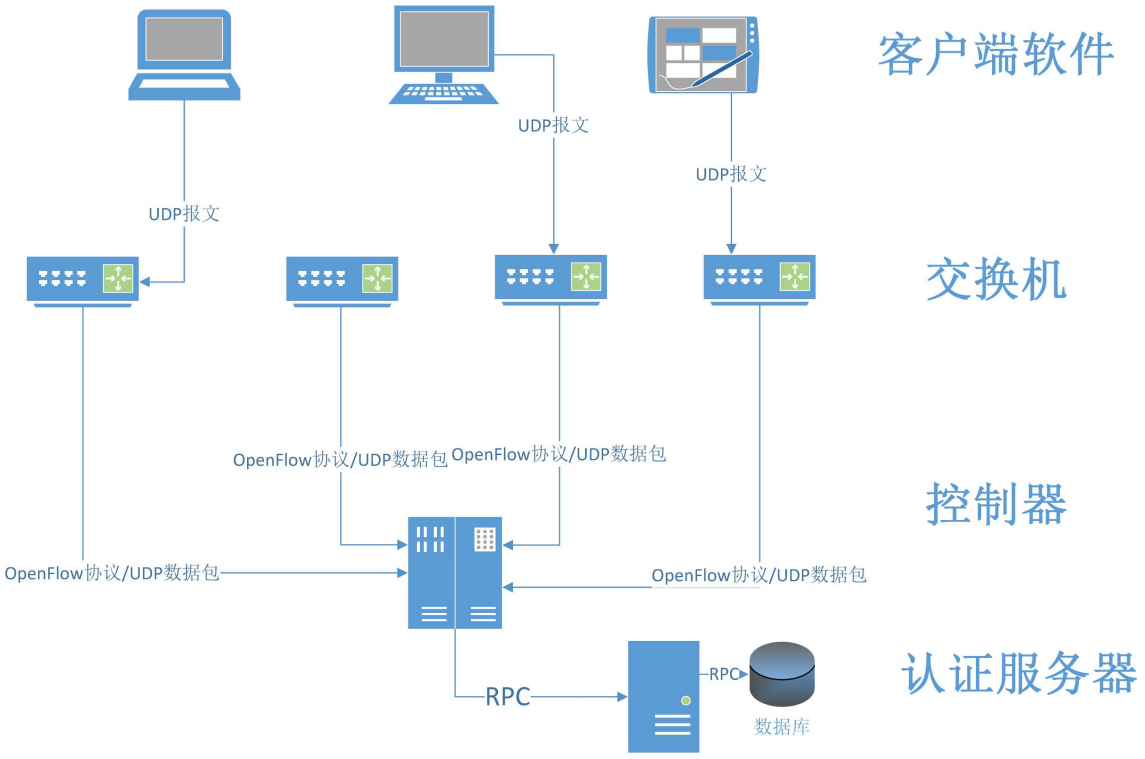


Figure 4. Schematic architecture diagram of custom access control protocol
图 4. 自定义访问控制协议原理架构图

3.1.2. 系统工作过程

系统工作流程分为三大部分，第一部分是初始配置，包括底层 Underlay 链路、OpenFlow 链路建立、虚拟机客户端启动；第二部分是认证鉴权过程，包括认证鉴权类的流表下发、客户端发起认证鉴权、控制器处理认证鉴权并返回结果；第三部分是根据认证结果，选择进行控制行为转发面，或者拒绝访问。

第一部分流程具体如下：

首先，网络中的交换机底层配置完毕，如下链路建立成功：交换机之间转发链路、交换机和控制器

的 OpenFlow 通道链路、虚拟机和交换机之间的连接链路。上述链路不限于物理连接，SDN 应用的网络虚拟化场景中，很多链路已经不再局限于物理链路，如本文所述的虚拟机场景，虚拟机和虚拟交换机都位于主机服务器上，在 Linux 操作系统中可以通过 tap 端口相连。

其次，底层链路建立完成后，交换机和控制器完成 OpenFlow 协议握手、建立 OpenFlow 协议的逻辑链路；

最后，虚拟机启动后，在操作系统完全启动后，启动认证鉴权软件，发起认证鉴权过程。

第二部分流程细分为 8 个步骤，具体如下：

1) 控制器首先向交换机下发认证鉴权过滤流表和认证鉴权上报流表；

2) 虚拟机中的认证鉴权客户端启动后，发送鉴权认证探测报文；

3) 交换机过滤掉所有非探测报文，如果收到探测报文，返回探测成功消息给虚拟机；

4) 客户端收到探测成功报文后，发送认证鉴权报文给接入交换机；

5) 接入交换机收到认证鉴权报文后，根据之前下发的认证鉴权上报流表，转发给控制器；

6) 控制器收到认证鉴权报文，根据报文内的口令鉴权认证，

a) 发送结果给交换机，同时还有指导交换机发送报文的转发行为；

b) 如果认证成功，给交换机下发虚拟机接入网络所需的流表；如果失败则不下发。

7) 交换机根据收到的含有认证结果的报文内的转发指令，将认证结果转发虚拟机内的客户端；

8) 虚拟机内的客户端收到认证结果，如果成功可以进一步访问网络。

上述 8 个步骤基本描述了本文所述的认证鉴权控制面的工作流程，是一个完整的自定义的网络认证协议，协议包括 8 个步骤和四种报文(图 5)。

第三部分是网络控制面流表下发，这一阶段主要下发正常网络转发流表，如 ARP、DHCP、ICMP、IP、TCP、UDP 等匹配条件的报文，这部分内容不在本文论述，不再赘述。

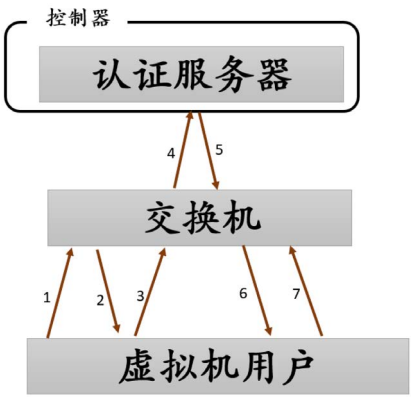


Figure 5. Custom access control protocol working process diagram
图 5. 自定义访问控制协议工作过程示意图

3.2. 自定义网络认证协议

在传统的网络当中对用户接入网络的控制技术逐渐成熟，但是这些方法在应用到 SDN 网络当中，会出现多种问题。比如 802.1x 协议在对交换机端口入网过程中，是对真实的交换机端口入网进行访问控制，但是在 SDN 网络架构中使用了虚拟化的技术，在一个交换机物理端口上可能存在多个虚拟机用户共用入网端口的情况，所以这种方式在实现虚拟机用户精确的入网控制方面存在不足。

上一节描述了系统认证的工作过程，本节将会从认证协议的角度描述协议涉及的细节，自定义网络

认证协议是本文的创新点，认证过程简单、协议流程健壮，对实际系统有着很强的实用性。本文将从报文、流程、转发行为三个方面具体描述本协议。

自定义网络认证协议的报文有：鉴权认证探测报文(Auth_Sniff)、鉴权认证探测确认报文(Auth_Sniff_ACK)、鉴权认证请求报文(Auth_Req)、鉴权认证请求确认报文(Auth_Req_ACK)。这四种报文都是基于现有 TCP/IP 报文，没有创建新的报文类型，理论上可以在现有所有支持 TCP/IP 协议的网络系统中应用，所有自定义网络认证协议有很强的向后兼容性，体现了本协议的实用性。

通过在现有协议里设置特殊字段的方式，利用 SDN 网络的灵活性，可以灵活高效的读取和修改报文内容，可以高效的实现本协议整个过程，同时使本协议具备了良好的可扩展性，如进一步扩展特殊字段的含义，可以增加认证过程，进一步提升认证安全性。

协议流程描述如下：网络进行初始化，控制器开始进行链路发现，掌握网络的拓扑信息。同时启动认证服务器，触发控制器向交换机下发用来识别链路层认证报文的流表项。虚拟机用户将携带链路认证报文的数据包发送给交换机。

交换机接收到虚拟机用户发送的报文后，进行本地流表查找匹配(此时交换机不能识别其他格式的报文，只能识别链路层认证报文，因为在交换机上预先下发了只匹配该报文的流表项)，如果匹配成功，就将此消息打包成 Packet-In 消息发送给认证服务器。

认证服务器会返回一个相应的报文给虚拟机用户，提示虚拟机用户输入用户名和密码。虚拟机用户收到消息后，输入用户名和密码并将认证消息发送给交换机，交换机将虚拟机用户的认证信息转发给认证服务器，认证服务器对虚拟机用户的有效数据进行认证，如果认证成功，通知控制器该虚拟机用户可以接入网络，并返回一个入网成功的报文给虚拟机用户，同时控制器会下发一个相应的流表项到交换机，打开该虚拟机用户的入网端口。如果匹配不成功，提示虚拟机用户认证失败。

虚拟机入网认证协议总体可以分为图中 7 个步骤，用户在进行入网申请时，需要在客户端输入相关认证消息，比如，用户名和密码。然后整个认证过程会完成以下步骤：

虚拟机会发送一个鉴权认证探测报文(Auth_Sniff)到 OpenFlow 交换机，交换机只识别此种报文，对其它形式的报文全部选择丢弃处理，交换机识别此类报文的依据是根据控制器预先下发的匹配规则，只能匹配到此类型的报文。(在实际的工作中，我们定义了一个特殊的 MAC 地址来进行匹配)。

当 OpenFlow 交换机匹配成功特殊报文之后会改动报文中的相关的内容，然后把改动后的鉴权认证探测确认报文(Auth_Sniff_ACK)报文返回给虚拟机，虚拟机在接到来交换机返回回来的报文之后，查看报文的改动情况，如果发再报文中相关字段有所改动，就会进行虚拟机接入网络的下一步认证操作。

虚拟机确认交换机返回的报文之后，会通过鉴权认证请求报文(Auth_Req)把用户名和密码发送到交换机，因为在交换机中没有处理此数据报文的流表规则，所以交换机会触发 Packet-In 消息，把此报文封装成 Pacekt-In 消息经过步骤 4 发到认证服务器模块。

认证服务器模块在收到交换机发送的 Packet-In 消息之后，会对 Packet-In 消息进行检查，如果是虚拟机入网请求消息就自己处理，如果是其它类型的 Packet-In 消息，认证服务器就会把此报文转给其它模块进行处理，自己不做干预。

认证服务器通过对 Packet-In 消息的解析，验证用户身份的准确性，发送鉴权认证请求确认报文(Auth_Req_ACK)，如果身份正确，通知控制器下发用于此用户入网的数据流转发规则，并通过过程 5 返回给用户一个确定消息，提示入网成功，用户便可以进行上网操作。如果认证没有通过，认证服务器只会通过步骤 5、6 返回给用户一个错误消息提示，认证没有通过，可能出现的错误提醒，如：用户名和密码错误等。不会通知控制器有针对该用户的进一步操作。

用户在收到认证服务器返回的消息之后，如果认证成功，虚拟机用户就可以通过步骤 7 进行入网操

作，如果认证失败则不能进行入网操作。

以上内容就是本论文所制定的虚拟机入网安全访问控制协议，通过此协议，SDN 网络可以对虚拟机入网请求达到精确的访问控制，虚拟机用户与交换机之间的链接接口可以是物理接口，也可以是逻辑接口，所以此协议完全适用于 OpenFlow 交换机的端口类型，可以较好的应用于基于 OpenFlow 协议的 SDN 网络当中。

3.3. 认证客户端模块

虚拟机客户端模块主要基于 TCP 协议和 socket 编程来实现。认证客户端安装在每一个虚拟主机中，虚拟机用户通过认证客户端向交换机申请操作请求，所有虚拟机用户必须要通过验证才能接入网络，虚拟机客户端验证的方法采用了传统的用户名和口令的方式进行验证。因为虚拟机用户要进行接入网络的操作，所以首先会对虚拟机客户端接入网络进行验证，如果在服务器端验证通过，则可以根据用户报文携带的相关信息通过控制器直接下发流表到虚拟交换机实现通信。

1) 虚拟机用户登录界面设计：采用传统的用户名和口令的方式,利用口令来确认用户的身份，是当前比较常用的一种认证技术，口令一般由字母、数字或者字母和数字的混合所构成的，口令的生成可以由系统直接生成，也可以由用户自己设定，通常一般为用户自己设定以便于记忆。本文采用了基于口令的验证技术，虚拟机用户在登录时提示输入用户名和密码，虚拟机用户根据自己设定的用户名和口令进行输入，为了增加安全，对用户的口令做了加密处理，本文的加密方式主要采用 MD5 加密算法，用户的口令和密码经过加密之后再进行传输，提高用户的使用安全。

2) 后台数据处理过程：在虚拟机用户输入信息并确定后，程序会把用户输入的认证信息保存在缓存当中，然后启动特殊报文封装和发送模块，向与虚拟机相连的交换机发送含有特殊目的 MAC 地址和标记字段的特殊报文到达交换机，因为在交换机中控制器事先预下发了关于匹配 PFA 报文的流表项，交换机只能够识别 PFA 格式的报文，在虚拟机没有完成认证之前，交换机对虚拟机发送的其他格式的报文实行丢弃处理，交换机在收到报文后，会对报文进行本地流表项的匹配与查找，如果匹配成功，下一步的操作过程就会转到流表项的动作集对报文进行处理，交换机处理此报文的过程把改变 PFA 特殊报文中的标记字段，然后把转发的输出端口更改为接收报文的端口，操作目的就是让更改后的报文原路返回到虚拟机上，虚拟机在收到交换机更改的报文之后，首先对报文进行验证，如果确定修改，就会把本地缓存中的用户认证信息，通过交换机发送到认证服务器上，进行用户身份的认证工作。

以上就是认证客户端工作过程的设计过程，客户端的主要工作就是获取用户的认证信息，在确定特殊报文通过交换机的认证之后把用户的认证信息发送到认证服务器上进行用户身份的认证。

3.4. 认证服务器模块

在 Ryu 控制器中开发一个具有认证服务器功能的模块。认证服务器模块作为 Ryu 控制器中的一个模块，随着 Ryu 控制器启动并加载运行，在控制器启动之后会向所有的边界交换机发送一个匹配规则，这个规则的作用就是用来匹配虚拟机用户在认证过程当中，第一步要发送的特殊报文。认证服务器提供整个网络的认证信息，所有未经认证的虚拟机用户访问网络都必须要主动经过该服务器认证。

当虚拟机用户的认证报文经交换机转发到认证服务器后，会经过认证服务器中的消息监听模块对消息类型进行监听。判定是否为携带用户认证信息的数据包，如果结果为真，则会把此数据包传递给消息处理模块对数据包进行进一步的分析处理，并通过解析数据包的方式提取出数据包中包含的用户认证信息。然后把用户认证信息传递到信息认证模块进行用户身份的认证操作。

信息认证模块会根据用户输入的认证信息与在认证服务器上保存的用户信息进行比较，实现对用户

信息合法性的认证。如果认证成功，认证服务器器会通过交换机返回给虚拟机一个认证通过的提示信息，并通知控制器该虚拟机用户身份合法，允许该虚拟机用户接入网络；如果认证失败，则将返回登录界面，提示用户名、密码错误等。

3.5. 基于特殊报文的认证和访问控制原理

3.5.1. 特殊报文(Packet for Authentication) PFA 的设计

数据报文是网络中信息交换与传输的载体，报文中通常包括了即将要发送的一些数据信息。同时无论在传统的以太网中，还是在 SDN 网络中，报文的长短并不是固定的，报文的长度可以根据携带信息量的大小，具有可变性。报文也是网络中网络传输的单位，在网络传输过程中，报文会不断的被封装成分组、包、帧来进行传输，而封装的方法通常是对报文添加一些信息段，这些信息段通常有报文头(由报文的类型、报文的版本、报文长度、报文实体等信息)所组成，再经过一定的格式所组织起来的。报文的认证方式主要有传统的加密认证方式、使用密钥报文认证码方式、数字签名的认证方式以及单向散列函数的认证方式。

本文基于以太网中数据链路层的报文基本形式，设计了自定义虚拟机用户入网访问控制协议中使用到的特殊报文，PFA (packet for authentication)报文，实现同数据链路层中其他报文的区别，以便于认证过程的特殊处理，主要是在报文的信息上添加了一个特殊的 MAC 地址用来与控制器规则进行匹配，同时增加了一个标记字段，用来进行虚拟机与交换机之间特殊报文的确认过程，报文中各字段组成如表 2 所示：

Table 2. Field composition of PFA messages
表 2. PFA 报文的字段组成

目的 MAC 地址	源 MAC 地址	标记字段	数据单元	校验位

PFA 报文主要包括目的 MAC 地址、源 MAC 地址、标记字段，有效数据单元和帧校验位，为了便于区别于其他格式的报文，PFA 增加了特殊的 MAC 地址，该 MAC 地址是一个固定的地址，不会因为虚拟机用户的改变而改变，就像是一个人和身份证号码一样，固定的目的 MAC 地址是 PFA 报文所特有的标志，设置数据帧校验位的目的是为了防止报文在在传输过程中数据被篡改，以太网中数据链路层的报文都会有一个校验位。PFA 报文在传输使用过程对用户是透明的，即用户感觉不到他在入网认证过程中，会有一个特殊报文的存在。

3.5.2. 基于 PFA 报文的认证原理

如图 6 所示：虚拟机用户在接入网络之前，要经过交换机对于报文的判别和认证服务器的认证，虚拟机用户只有认证通过后才能接入网络进行下一步的操作，这样增加了网络使用的安全性，下面分别从交换机对报文的判别和服务认证两个过程来详细介绍 PFA 报文的认证过程。

1) 交换机对 PFA 对报文的识别：

一个交换机可以同时连接多个虚拟机用户，每个虚拟机都对应一个端口(可以是物理端口，也可以是逻辑端口)，所有虚拟机接入的端口都是通过用户主机的物理网卡与交换机进行连接。网络初始化后，虚拟机与交换机相互连接，但是虚拟用户与交换机之间没有任何的数据传递，即虚拟机用户是不能通过交换机连接到互联网的。在当前状态下，虚拟机用户可以通过虚拟机上的客户端向交换机发送任何形式的报文，交换机的功能是负责数据的转发，当交换机接收到虚拟机用户发来的报文后，首先会对报文的格式进行判别，看是否为 PFA 报文，判别的依据是根据控制器事先预下发了对于 PFA 报文匹配的流表项，

所以交换机本地存有关于 PFA 报文的匹配规则，不用把这种数据报文再封装成 Packet-In 消息发送到控制器上，这一步可以大大的提高虚拟机用户入网访问控制的效率。但是虚拟机用户发送过来的普通报文，比如没有经过验证，就开始访问互联网操作，因为交换机不能识别从而不能对报文进行转发操作，所以不会对这样的报文进行处理，直接丢弃。但是如果是 PFA 格式的报文，交换机对本地规则进行查找后，如发现 PFA 格式的报文，交换机就会改变此报文中标识字段的值，然后再把这个报文返回到虚拟机用户。

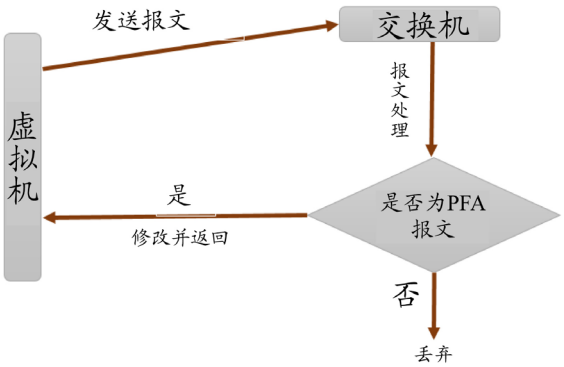


Figure 6. PFA message authentication process
图 6. PFA 报文认证过程

2) 认证服务器对报文的认证：

虚拟机在收到由交换机返回回来的报文之后，检查报文标识字段值的变化情况，如果发现标识字段的值有变化，虚拟机就会把认证信息，即用户输入的用户名和密码通过网络传输传送到认证服务器上。接着认证服务器模块实现对虚拟机用户身份进行验证，判断其合法性。本文中的认证服务器是在控制器中开发的认证模块，内置于控制器之中。交换机在确定虚拟机用户发送过来的认证报文以后，因为本地没有进行此报文的匹配规则，无法实现报文的转发，根据 OpenFlow 协议，交换机会将报文打包成 packet-in 消息并将这个消息通过交换机与控制器之间的安全通道转发给认证服务器，认证服务器在收到交换机发来的 Packet-In 消息后，提取报文所携带虚拟机用户的有效数据，并对虚拟机用户的有效数据进行认证，如果匹配成功，则将消息发送给控制器，告知控制器该用户身法已通过验证，可以接入网络，控制器将根据 报文的有效数据，形成相应的流表规则，并将流表下发至交换机，告知交换机在以后的处理过程中，对该用户的数据进行处理，实现数据流的转发，实现虚拟机用户入网的操作。如果用户的身份认证失败，则将结果反馈给虚拟机用户，提示虚拟机用户用户名密码错误，虚拟机用户可根据提示进行修改再次进行操作请求。图 7 为虚机用户入网访问控制整个认证过程。

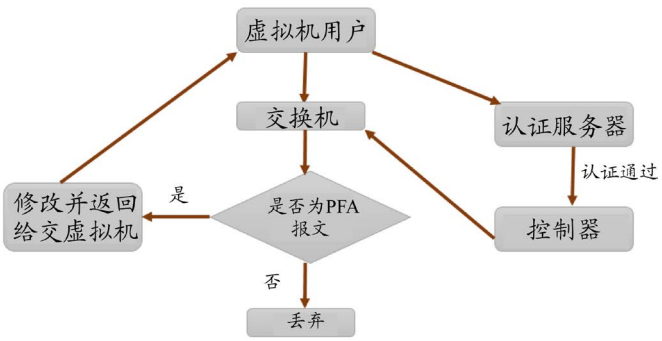


Figure 7. Virtual switch user network authentication process
图 7. 虚拟机用户入网的认证过程

3.6. 本章小结

本章的主要工作是介绍了本论文的自定义虚拟机入网认证的协议和工作流程，详细描述了基于OpenFlow 协议的 SDN 网络实现虚拟机用户入网访问控制的自定义协议，并介绍了此协议的工作流程，以及在实现此协议过程中相关模块的介绍，通过本章的介绍我们可以知道，自定义虚拟机入网访问控制协议可以实现对虚拟机用户接入网络的更加细粒度的控制，能够对每个过程进行详细的描述，认证过程也十分的具体。

参考文献

[1] 李艳, 郝志安, 李宁, 卢冀. 虚拟机安全监控与防护系统的设计与研究[J]. 计算机与网络, 2014(5): 1008-1739.

[2] 张诚. 基于 SDN 的宽带接入网用户认证方式研究[J]. 通讯世界, 2015(23): 8-9.

[3] 王正昶, 杨波. 802.1x 认证协议在局域网管理中的应用[J]. 科技致富向导, 2013(3): 7-9.

[4] 高亚军. 基于接入交换机的 Web 认证研究与实现[D]: [硕士学位论文]. 广州: 华南理工大学, 2013: 2-9.

[5] 徐张廷. 基于 Flex 的 RIA 系统中访问控制的设计和实现[D]: [硕士学位论文]. 杭州: 浙江大学, 2011: 1-7.

[6] 房秉毅, 张歌, 张云勇, 黄韬, 谢俊峰. 开源 SDN 控制器发展现状研究[J]. 邮电设计技术, 2014(7): 29-36.

[7] 陶松. 基于 SDN 的网络虚拟化安全研究[J]. 电脑知识与技术, 2015, 11(15): 23-25.

[8] Open Network Foundation (2012) Software-Defined Networking: The New Norm for Networks. https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

[9] 胡佳. SDN 路由交换技术研究与实现[D]: [硕士学位论文]. 北京: 北京信息科技大学, 2013: 1-7.

[10] 李彬先. 基于 Openflow 的虚拟交换技术的研究[D]: [硕士学位论文]. 济南: 山东大学, 2015: 1-7.