



基于软件定义价格的 SDN 应用体系结构

诸葛斌¹, 王保霞¹, 王怡宁², 吴春明², 姚敏辉¹

(1. 浙江工商大学信息与电子工程学院 杭州 310018;

2. 浙江大学计算机科学与技术学院 杭州 310027)

摘要: 首先, 从用户付费调用网络资源的角度提出了 SDN 应用概念; 接着, 通过对比传统的桌面应用编程、传统的网络编程和 SDN 应用编程, 描述了 SDN 应用编程的优越性; 然后, 提出了软件定义价格的概念, 综合说明了 SDN 应用的特点, 设计了基于软件定义价格的 SDN 应用框架, 并将其应用于 OpenDaylight 开源项目; 最后, 给出了基于软件定义价格的虚拟租户网络测试案例。实验表明, 用户可根据软件定义价格自主选择所需的虚拟租户网络。

关键词: 软件定义网络; SDN 应用; 软件定义价格; 虚拟租户网络

doi: 10.11959/j.issn.1000-0801.2015207

Architecture of SDN Applications Based on Software Defined Price

Zhuge Bin¹, Wang Baoxia¹, Wang Yining², Wu Chunming², Yao Minhui¹

(1. School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China;

2. College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China)

Abstract: Firstly, the concept of SDN applications was proposed from the perspective of users payment for allocating network resources. Secondly, with the comparison between traditional desktop application programming and network programming, the advantage of SDN programming was described. Thirdly, the concept of software defined price (SDP) was introduced. Based on the characteristics of SDN applications, the framework of SDP-based SDN application so as to apply to the OpenDaylight open source project was designed. Finally, an implementation of creating a virtual tenant network based on SDP was showed. The results of experiment present that the users can choose their desired virtual tenant according to the mechanism of SDP.

Key word: SDN, SDN application, software defined price, virtual tenant network

收稿日期: 2015-03-25; 修回日期: 2015-08-03

基金项目: 国家重点基础研究发展计划 (“973” 计划) 基金资助项目 (No.2012CB315902), 国家自然科学基金资助项目 (No.61379120, No.61402408, No.61170215), 浙江省重点科技创新团队基金资助项目 (No.2011R50010-15, No.2011R50010-16, No.2011R50010-17), 浙江省新型网络标准与应用技术重点实验室基金资助项目 (No.2013E10012)

Foundation Items: National Basic Research Program of China (973 Program) (No.2012CB315902), The National Natural Science Foundation of China (No.61379120, No.61402408, No.61170215), Zhejiang Leading Team of Science and Technology Innovation Program (No.2011R50010-15, No.2011R50010-16, No.2011R50010-17), Zhejiang Provincial Key Laboratory of New Network Standards and Technologies Program (No.2013E10012)

论文引用格式: 诸葛斌, 王保霞, 王怡宁等. 基于软件定义价格的 SDN 应用体系结构. 电信科学, 2015207

Zhuge B, Wang B X, Wang Y N, et al. Architecture of SDN applications based on software defined price. Telecommunications Science, 2015207



1 引言

SDN (software defined networking, 软件定义网络) 是一种新型的网络架构及技术, 起源于 2006 年美国斯坦福大学的 Clean Slate 研究课题, 由 McKeown 教授^[1]于 2009 年正式提出^[2,3]。其主要特征为控制平面和转发平面分离、开放接口和集中控制^[4]。将转发平面和控制平面分离能够把网络看成一个逻辑或虚拟的实体, 将转发行为抽象; 提供标准化的开放接口, 能够实现应用和网络无缝集成; 集中控制则可以获取网络资源(如流、功能、链路、虚拟租户网络)的全局信息并根据业务需求对资源进行全局调度和优化^[5]。当前主流的 SDN 主体框架有从用户角度提出的 ONF (Open Networking Foundation, 开放网络基金会) SDN 架构、从网络运营商角度提出的 ETSI (European Telecommunications Standards Institute, 欧洲电信标准化协会) NFV (network functions virtualization, 网络功能虚拟化) 架构^[6]和为具体实现 SDN 架构由业界巨头 Cisco (思科)、IBM、Microsoft (微软) 等联合提出的 OpenDaylight^[7] 开源 SDN 项目。OpenDaylight 开源项目于 2013 年 4 月提出, 旨在打破大厂商对网络硬件的垄断, 以驱动网络技术创新力。

SDN 的主旨是希望通过对网络进行抽象来推动网络业务的创新, 并从用户的角度调度网络资源以改进网络。通过北向接口, SDN 的开发者可以运用软件编程的方式实现对不同网络资源和能力的调用^[8]。鉴于 SDN 本身的特点, 相比其他应用, SDN 应用具有更强的智能性、动态控制性和可编程性, 整体网络可根据业务的网络需求及时动态地调整网络资源能力的交付^[9-11]。当前 SDN 已经被广泛应用, 例如, 可编程的网络实验床、数据中心网络、企业级网络、无线网络、网络测量等^[12]。数据中心网络由于具有流量大、周期性强和突发性强等特点, 要求网络能够多路径转发负载均衡、带宽按需提供和集中管理控制, 而 SDN 具有集中控制特性, 可通过统一部署的控制器获取数据中心的流量需求为其调度, 按需分配; 在数据中心采用 SDN 架构不仅能优化路由路径、降低网络维护代价, 还能提高网络设备的利用率并增加网络设备的可管理性和灵活性^[13]。企业级网络具有业务类型多的特点, 要求网络灵活、自定义, 并具有很高的安全性, SDN 因其控制和转发分离的特点, 网络设备简单化、通用化, 简化了网络管理, 并且应用层可通过北向接口和控制层进行交流^[14]。同时, 高校也进行了 SDN 应用创新开发大赛^[15]。但 SDN 的研究不应只局限在

SDN 上, 而是应推进创新网络业务的发展, 即本文提出的 SDN 应用。从当前工业界和学术界来看, SDN 应用这个概念还只是局限于数据中心、流量优化和 QoS 保障等方面, 并没有被很好地阐述。因此, 本文首先通过比较传统的桌面编程、传统网络编程和新型 SDN 应用编程之间的不同, 明确了 SDN 应用的概念, 然后提出了软件定义价格 (software defined price, SDP) 的概念, 设计了基于 SDP 的 SDN 应用框架, 并将其应用于 OpenDaylight 开源项目, 最后给出了基于 SDP 的虚拟租户网络测试案例。

相比其他文献, 本文最显著的特点是提出了价格模块, 从用户的角度根据价格调用网络资源, 并对网络资源进行了定价; 其次, 引入了多虚拟平台概念, 在用户调用底层网络资源时, 通过与多个虚拟平台进行协商, 在满足用户提供的 <网络资源、资源定价> 条件下, 为用户创建虚拟租户网络, 虚拟租户网络通过定价机制和下层进行协商, 获取底层网络资源; 最后, 打破了底层物理网络的限制, SDN 应用是建立在虚拟租户网络之上的, 用户对于底层网络环境来说是透明的, 用户所关心的是创建虚拟租户网络所需的配置和价格。

2 问题分析与动机

2006 年 Google (谷歌) 首席执行官 Eric Schmidt 在搜索引擎大会上首次提出了“云计算”^[16] (cloud computing) 的概念。云计算描述了一种基于互联网的新的 IT 服务增加、使用和交付模式, 通常涉及通过互联网来提供动态易扩展且通常是虚拟化状态的资源。美国国家标准与技术研究院 (National Institute of Standards and Technology, NIST) 定义云计算是一种按使用量付费的模式, 这种模式提供可用的、便捷的、按需的网络访问, 进入可配置的计算资源共享池, 资源能够被快速提供, 只需投入较少的管理工作。

云计算的目的是整合共享云计算资源, 利用虚拟化技术将其形成具有服务能力的云计算服务产品^[17,18], 通过互联网将数据和计算资源以服务的形式租用给所需的用户。云计算着重为用户提供服务, 服务的使用和支付都依赖于商业模式的处理, 因此, 经济学成为了云计算发展的驱动力^[19]。在云计算市场中, 多个云计算资源提供商和多个用户共同参与交易形成竞争市场, 双方针对多类云计算服务展开交易并追求效益最大化。同时, 由于云计算服务的高性价比、高可靠性、应用分布性、可扩展性、高度灵活性等特点, 已经使得 IT 界大型企业进军云计算服务领域, 云计算市场规模急剧增长。近些年, 国内外已有相当多的文献对云计算中的

经济模型^[20]进行了研究,典型的有基于经济模型的网格资源管理分配^[22]和任务调度优化等相关问题的研究。

其次,目前已有很多技术将云计算和SDN结合起来,以满足用户的需求定制化和多业务的需求,例如OpenStack中Neutron和SDN的结合,弥补了Neutron组件中网络节点的公网瓶颈^[23-25]。随着网络的不断发展,软件定义手机、软件定义服务器、软件定义网络也逐渐为人们所知,也必将是未来网络发展的方向之一。但是,鉴于当前SDN发展不健全,较少有或者还未有基于经济模型的网格资源分配的相关研究文献。由于SDN能够将网络资源虚拟化,通过集中控制的方式进行集中管理,所以SDN资源能够进行交易;另外,在实际的SDN应用中,资源并不是无偿使用的,必须引入经济模型中的定价机制,鉴于云计算中经济模型的研究已非常成熟,本文借鉴云计算定价机制在SDN中提出了资源定价机制。

本文提出的SDN应用指以SDN为基础,通过软件编程的方式调用不同的网络资源和服务,充分利用SDN的特点,根据用户需求特点提供个性化的便捷服务。在本文提出的SDN应用中,云计算和SDN应用的相似之处在于具备资源能够池化、按需服务、支持异构资源和支持应用层异构多业务等功能;另外还是一种生产者—消费者模型,强调资源专有,用户通过付费的方式获取自己的专有资源。

另外,在实际的云计算服务产品中,基于价格的服务层出不穷,如国外微软的Windows Azure和国内的新浪云平台对地理信息服务的收费模式。新浪云平台提供3种级别的地理信息服务,分别为:一级地理信息服务为开发者提供自驾路线信息、公交线路信息以及公交站信息;二级地理信息服务为开发者提供根据地理信息坐标返回实际地址服务;三级地理信息服务为开发者提供根据移动基站Wi-Fi等数据获取当前位置信息服务以及生成一张静态地图图片等服务。根据这3种级别的地理信息服务模式,新浪云平台相应的计费方式分别是:调用一级服务,免费且次数不限;调用二级服务,成功调用则支付0.1付费云豆(新浪云的计价方式为1元=100付费云豆);调用三级服务,成功调用则支付0.5付费云豆。

不仅在新浪云,很多其他云平台都采用相似的收费方式,例如,传统IT厂商的IBM“蓝云计划”与“智慧城市”、惠普(HP)云计算、浪潮和宝德;还有互联网提供商转型为云计算提供商的Amazon EC2(Elastic Compute Cloud)、GCE(Google Compute Engine)、阿里云、百度云、盛大云和华为

云;最后是软件厂商Microsoft Windows Azure、VMware、SalesForce和Citrix^[26]。从上述平台的计费模式可以看出,目前基于价格的服务中,资源的调用是通过调用函数的成功次数来计算其价格的,划分的颗粒度和资源价格的单位足够细小,但其价格都是人为规定的,缺乏一定的灵活性。所以,本文提出了软件定义价格(SDP)这个概念,并同时提出了基于SDP的SDN应用,旨在解决当前服务缺乏价格因子或价格死板的问题。在基于SDP的服务中,其价格随时间、负载等动态地变化。

3 SDN应用概念

编程是指使用某种程序设计语言编写程序代码,并最终得到结果的过程^[27]。计算机编程语言从最早的机器语言、汇编语言发展到了后来的面向过程的程序设计语言以及面向对象的程序设计语言。与此同时,随着网络技术的不断发展,编程模式也变得更加多样化。最初的编程模式是简单的桌面应用编程模式,其后是逐渐壮大的基于网络的编程模式。现如今随着SDN的提出,编程模式又发生了变化,也就是即将提出的SDN应用编程。

首先从编程模式进行探讨,分析其不同,并得出SDN应用编程的概念以及应用。

3.1 传统的桌面应用编程

传统的桌面应用编程是指通过计算机程序设计语言,调用计算机内核提供的API(application programming interface,应用程序编程接口),并得到最终结果的过程。传统的桌面应用编程如图1所示,在控制台里输入time,该命令调用系统自带的时间函数,最终显示系统当前的时间为10:19:52。这种形式的编程一般比较简单,只是简单地调用计算机内核提供的接口,一般只涉及一台电脑,且不存在网络通信。

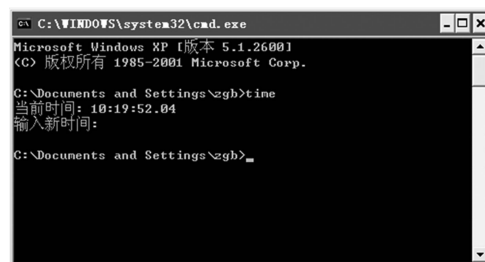


图1 桌面应用编程示意

3.2 传统的网络编程

传统的网络编程指在发送端把信息按照规定好的协



议组成分组,接收端根据规定好的协议解析分组,从而提取出相应的信息,达到通信的目的。其中,最主要的工作包括对数据分组的组装、过滤、捕获、分析以及处理^[28]。

狭义的网络编程指在操作系统上的套接字(socket)编程;广义的网络编程泛指网络环境下的一切程序设计;socket 程序编程和动态网页 Web 程序设计。网络编程的主要结构是客户端/服务器(client/server,C/S)工作模式,客户端一般为网络用户的计算机,其处于主动的地位,可以向服务器发出各种请求;服务器指在网络上能够提供特定服务的主机,处于被动的地位,根据用户的请求做出相应的回答,提供相应的服务。用户机进程为用户提供图形用户界面(graphical user interface, GUI),根据用户输入的数据和命令向服务器发出请求,将服务器做出的回答进行分析处理,通过 GUI 提交给用户。服务器端进程为多个用户机提供服务,以实现资源共享。

传统的网络编程一般是 C/S 工作模式。图 2 和图 3 表示通过 Java 编程来获取服务器上的时间。在此过程中,Java 程序调用 Java API 中的 currenttimemillis (true, url)函数,分别向 URL 为“ntp-sop.inria.frserver”、“time-nw.nis. gov”以及“time.asia.apple.com”的服务器发送获取当前时间的请求,服务器在收到客户端的请求后,根据客户端的请求 URI(uniform resource identifier,统一资源标识符)和参数,解析客户端的请求,计算客户端需要的时间参数,将所得结果返回给客户端,最后,客户端获取到 3 个服务器传过来的数据,对其进行解析,并将最新的时间显示在终端,分别为 2014-06-16 12:32:51、2014-03-16 12:33:54 以及 2014-06-16 12:34:05。一般情况下,传统的网络编程通过一个客户端访问一台服务器,且只涉及一条网络链路。

```
public static void main(String[] args) {
    String url = "ntp-sop.inria.frserver";
    long start = System.currentTimeMillis();
    long time = currentTimeMillis(true, url);
    long end = System.currentTimeMillis();
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    String date = sdf.format(new Date(time));
    System.out.println("服务器地址: " + url);
    System.out.println("当前时间: " + date);
    long spendTime = end - start;
    System.out.println("消耗时间: " + spendTime + "毫秒");
    System.out.println();

    url = "time-nw.nist.gov";
    start = System.currentTimeMillis();
    time = currentTimeMillis(true, url);
    end = System.currentTimeMillis();
    date = sdf.format(new Date(time));
    System.out.println("服务器地址: " + url);
    System.out.println("当前时间: " + date);
    spendTime = end - start;
    System.out.println("消耗时间: " + spendTime + "毫秒");
    System.out.println();

    url = "time.asia.apple.com";
    start = System.currentTimeMillis();
    time = currentTimeMillis(true, url);
    end = System.currentTimeMillis();
    date = sdf.format(new Date(time));
    System.out.println("服务器地址: " + url);
    System.out.println("当前时间: " + date);
    spendTime = end - start;
    System.out.println("消耗时间: " + spendTime + "毫秒");
    System.out.println();
}
```

图 2 获取时间的 Java 程序

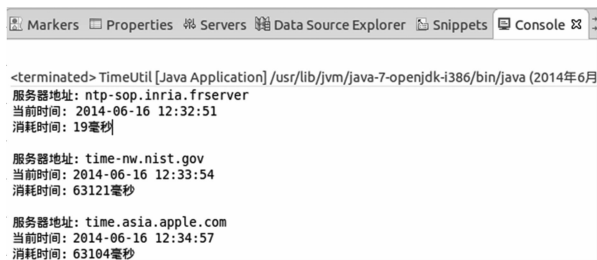


图 3 获取时间的结果显示

3.3 新型的 SDN 应用编程

本文所提出的新型 SDN 应用编程能够根据用户需求特点提供个性化的便捷服务,以 SDN 为基础,通过软件编程的方式调用不同的网络资源和服务,并且充分利用 SDN 的特点。本文所论述的 SDN 应用均特指在 SDN 中添加 SDP 模块并且能够根据价格选择服务或资源的应用。

为了更好地理解 SDN 应用,本文以微信公众平台为基础,以时间服务为例,通过微信公众平台开发了一个获取当前时间的 SDN 应用,微信请求时间结果如图 4 所示。

该 SDN 应用隐藏地获取当前时间的流程如图 5 所示。

微信客户端请求时间的具体过程为:

- (1)发送问号(?)给微信服务器;
- (2)微信服务器根据公众账号将对客户端的请求发给虚拟服务器;
- (3)虚拟服务器根据客户端获取时间的请求将请求发送给不同的时间服务器;
- (4)SDN 控制器结合时间服务选择算法,比较不同时



图 4 微信请求时间结果

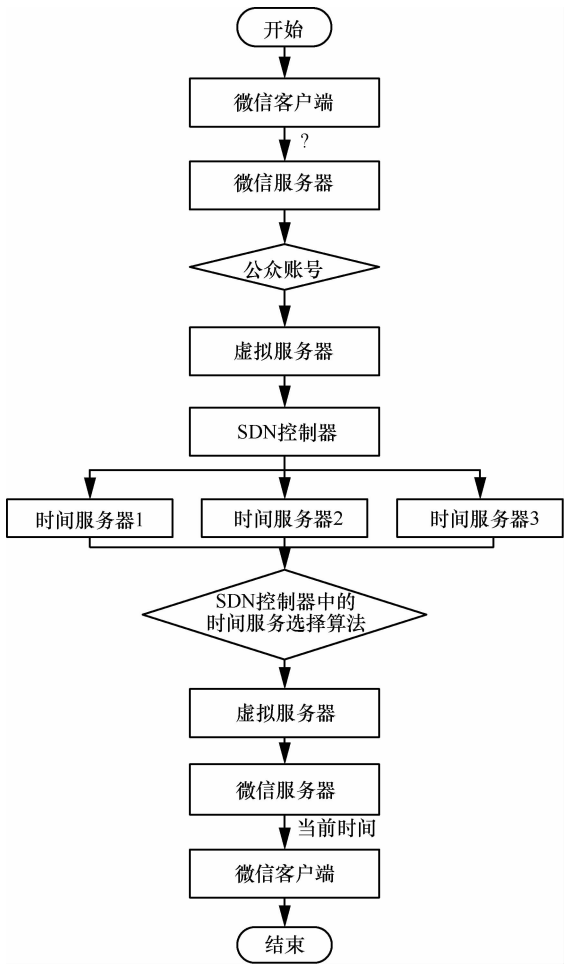


图 5 微信请求时间流程

间服务器返回的时间服务,选择最优的时间服务返回到虚拟服务器;

(5)虚拟服务器将对应的时间返回到微信服务器;

(6)微信服务器将当前时间返回到微信客户端。

在所提出的 SDN 应用编程中,微信客户端访问虚拟服务器,虚拟服务器通过特定协议访问多组时间服务器以获取当

前的时间资源,这些服务器可以由多个虚拟网络、数据中心、虚拟服务器协同工作组成,为用户提供个性化的便捷服务。

最后,在操作便捷性、服务质量、是否涉及网络、机器数量、地域范围/规模、软件收费方式、软件存放位置、软件维护升级、知识产权保护能力和用户群可扩展性等方面对传统的桌面编程、传统的网络编程以及新型的 SDN 应用编程进行了综合的比较,具体见表 1。由表 1 可知,传统的桌面编程由于受到网络、硬件等各方面因素的限制,相比另外两种编程模式存在一定的劣势;而传统的网络编程由于受到网络灵活性的限制,与新型的 SDN 应用也有一定的差距;同时,新型的 SDN 应用编程,以 SDN 为基础,不管是在操作便捷性、服务质量、是否涉及网络、机器数量和地域范围/规模方面,还是在软件收费方式、软件存放位置、软件维护升级、知识产权保护能力和用户群可扩展性等方面,都具有明显的优越性,可以为用户提供个性化的服务。

4 基于 SDP 的 SDN 应用体系结构

基于 SDP 的 SDN 应用体系结构是一种面向 SDN 应用编程、基于资源定价和交易的体系结构。与现有的编程模型不同,该结构指导编程人员从一个新的角度从事编程工作,给用户提供一个良好的视图界面。鉴于 SDN 应用具有价格模块,首先提出 SDP 的概念;然后,结合 SDP 与 SDN 应用编程体系,提出基于 SDP 的 SDN 应用体系框架。

4.1 SDP 概念

随着 SDN 的发展,网络中大量的交换机、路由器等各类资源均可转换为虚拟资源,形成规模巨大的虚拟网络资源,为网络用户提供能力无限的资源服务。但当前的资源定价是由服务商直接定义的,缺乏一定的灵活性和公平性,而 SDP 希望通过软件的方式定义价格,从而摆脱价格的单一性,让价格可以随时间、流量等因素变化,以提高价格的灵活性。

表 1 几种编程技术的比较

技术特点	传统的桌面编程	传统的网络编程(C/S 模式)	新型的 SDN 应用编程(云平台)
操作便捷性	一般	好	好
服务质量	一般	一般	好
是否涉及网络	否	是	是
机器数量	一台 PC	一台服务器	多台服务器
地域范围/规模	一台 PC	有网络的地方/至少两台 PC 机	有网络的地方
软件收费方式	购买软件	购买软件	成功调用按次收费
软件存放位置	用户机	用户机	服务器
软件维护、升级	主动更新	自动更新	透明更新
知识产权保护能力(破解问题)	弱	中	强
用户群可扩展性(针对用户)	差	一般	好



SDP 是一种新型的价格定义模式,其核心是通过自定义算法来确定资源的价格,通过软件定义价格协议 (software defined price protocol,SDPP)在买方和卖方相互协商后确定最终的成交价。

SDPP 是 SDP 机制的核心组成,主要指服务商与用户进行交易时,在协商价格过程中定义的一种双方认可的协定。完整的 SDPP 主要包括交易对象、交易内容、交易费用以及交易对象的义务等。同时,服务商可以对用户在工作负载和资源使用方面做出规定。SDPP 可以保证用户在使用资源时的服务质量。

SDPP 的主要思想是:当用户需要访问服务商提供的资源时,服务商将资源的价格列表返回给用户,用户提供对应资源的服务价格,服务商依据自己的价格协商算法,比较并计算该用户是否有权限使用该资源,若满足,则提供相应服务水平的资源;若不满足,则不能达成协议,交易终止。SDP 协商过程如图 6 所示。

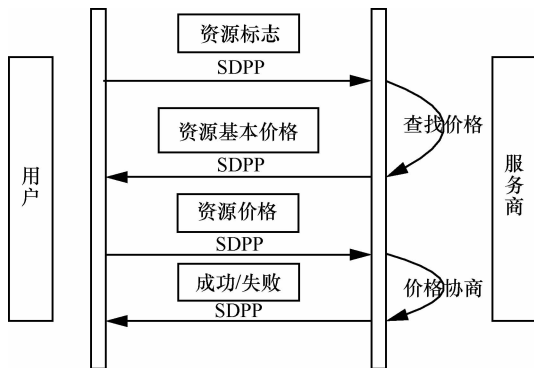


图 6 SDP 协商过程

SDPP 协议头如图 7 所示。SDPP 协议头包括:身份标识 (ID)、优先级 (level)、表达 (expression)、时间戳

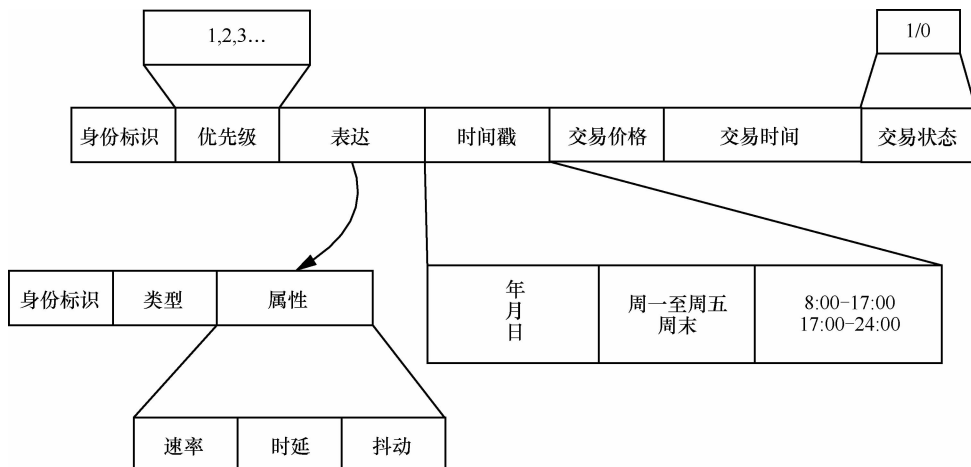


图 7 SDPP 协议头

(timespec)、交易价格 (price)、交易时间 (contract date)以及交易状态 (state),具体介绍如下。

- 身份标识:协议头的唯一标识。
- 优先级:服务优先级的标识。
- 表达:包括身份标识、类型 (type)、属性 (property),其中,身份标识是资源的唯一标识,类型用于表示资源的类型,属性用于表示资源的属性,包括速率、时延、抖动等。
- 时间戳:服务时间的一个结构体,主要指服务时间段的选择。
- 交易价格:交易时的价格。
- 交易时间:完成交易的时间。
- 交易状态:1 表示交易成功;0 表示交易失败。

4.2 基于 SDP 的 SDN 应用

SDN 应用的概念之所以被提出是由于各种网络社群的骤增。这些网络社群用户具有相同的目的,通过网络聚集到一起。基于 SDP 的 SDN 应用以 SDN 为基础,即含有 SDP 模块,通过软件编程的方式调用各种网络资源,根据用户需求特点并在满足用户需求的情况下提供具有价格信息的服务。由于在一个完全开放的竞争 SDN 应用市场中,资源服务提供商和资源消费者不能完全掌握彼此对资源和服务的真实估价,因此资源/服务的交易是信息不对称的博弈问题。交易双方的经济效益与彼此的出价策略紧密联系。因此,资源服务交易的双方就构成了双向拍卖,这是信息不对称的贝叶斯博弈问题。本文设计的基于 SDP 的 SDN 应用具有以下典型的特点。

- 既可作资源提供商也可充当资源消费者。当 SDN 应用需要调用资源或者功能时,向其他 SDN 应用付款

以获取资源 and 功能, 此时该 SDN 应用充当资源消费者; 与此同时, SDN 应用中的任何资源或功能可被定价并销售, 该 SDN 应用充当资源提供商。

- 具有 SDP 模块。SDN 应用中添加了 SDP 模块, 该模块主要包含通用价格模块和通用交易模块, 分别用于资源的定价和交易。其中, 通用价格模块是一个框架, 内置几种价值算法。用户可以选择内置的价值算法, 也可自定义价值算法。资源价值随时间、当前网络流量、资源紧缺充裕程度以及历史价格而变动, 价值算法根据这些因素重新定价出当前资源的价值。该价值是 SDN 应用资源提供商的私人信息, 只有资源提供商自己知晓。
- 交易对象可协商价格。在交易过程中, SDN 应用资源提供商根据当前的定价价值, 重新包装价格, 并呈献给 SDN 应用资源消费者。交易时, 若资源消费者不满意资源提供商的价格, 双方可协商价格。若协商成功, 则交易成功; 反之, 协商破裂, 交易失败。
- 资源可代替性。资源提供商以不同的价格提供资源, 这些资源具有可代替性, 相同单位的资源在不同提供商中的价格差异比较大^[19]。资源消费者在调用资源的过程中下放命令到不同的资源提供商, 拥有该资源的资源提供商根据自身的定价机制将价格呈现给资源消费者, 资源消费者根据价格预算和需求选择合适的资源提供商。
- 区别于其他的编程, SDN 应用基于一定特征的网络社群(如 QQ 群, 具有特定的对象), 拥有至少两个用户, 这些用户具有共同的认知感, 每个用户的行为都有相同而明确的目标和期望^[20]。SDN 应用可按照时间划为两种类型: 临时型和长期型。顾名思义, 临时型 SDN 应用指因为某种原因而临时建立起来的应用, 当达到预期效果后, 该 SDN 应用可解散; 长期型 SDN 应用则指根据某种需求或盈利目的而建立并要求长期运行维护下去的应用。

4.3 SDN 应用实例

根据 SDN 应用的典型特点, 即 SDN 应用具有 SDP 模块、同时充当资源提供者和资源消费者以及针对一定特征的网络社群, 本节给出一个典型的 SDN 应用实例——微信请求资源过程。

SDN 控制器通过 SDP 模块将请求下发给多个云平台, 例如新浪云服务、百度云服务和阿里云服务。不同的服

务给出当前资源的价格, SDN 控制器可根据预期价格和资源选择云服务并进行价格协商。图 8 给出了基于 SDP 的微信客户端获取服务/资源的流程。微信客户端发送获取当前地址的请求给微信服务器, 微信服务器将请求发送给虚拟服务器, 虚拟服务器根据 SDP 机制, 得到当前网络状态下的价格并提供给 SDN 控制器, SDN 控制器根据不同的虚拟基础平台提供的价格择优选择服务并返回给客户端, 同时, 在虚拟服务器中的对应账号中扣除相应的云豆。

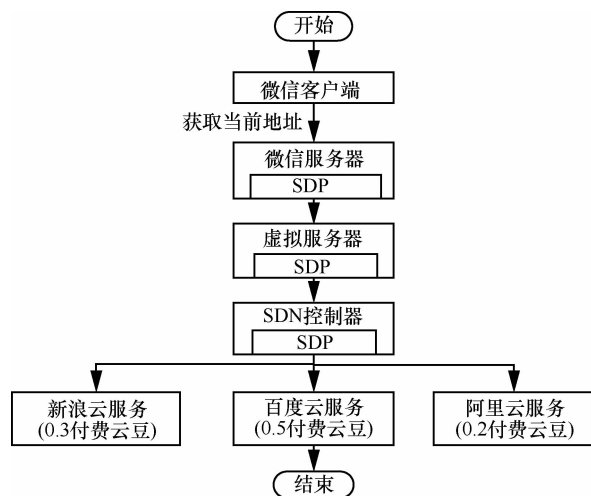


图 8 微信请求资源过程

4.4 基于 SDP 的 SDN 应用体系框架

本系统旨在基于 SDP 建立具有价格因子的 SDN 应用体系架构。新型的 SDN 应用体系如图 9 所示, 主要包括 SDN 个性化服务器、虚拟服务器、SDN 控制器以及基础服务器 4 部分。

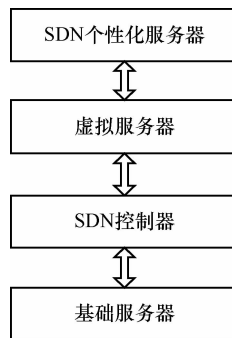


图 9 新型的 SDN 应用体系

而本文设计的 SDN 应用是基于 OpenDaylight 开源项目的, 其中, 基础服务器对应项目中的转发层, SDN 控制器属于控制层, 虚拟服务器和 SDN 个性化服务器分别对应用层中的虚拟租户网络^[20](virtual tenant network, VTN)和上层应用。



SDN 应用体系框架如图 10 所示。SDN 应用层主要包括以新浪、百度、阿里云服务器,微信平台以及以 Google 翻译、天气预报等为主体的三大类平台,各个平台都有自己的 SDP 模块,它们之间以 REST 形式进行通信。在应用层中的业务层主要包括 VTN 协调者 (VTN coordinator)、OpenStack Neutron 以及其他业务层的服务,各个服务都包含 SDP 模块,它们之间以 REST 形式进行通信。控制层和物理层的各个模块都有自己的 SDP 模块,层与层之间通过

SDPP 进行通信。

5 基于 SDP 的虚拟租户网络系统的实现与测试

5.1 测试方案设计

底层物理拓扑设计如图 11 所示,系统测试的整体拓扑由 8 台主机和 7 台交换机组成,图 11 中已标出路由器之间的链路价格,其测试用例见表 2。

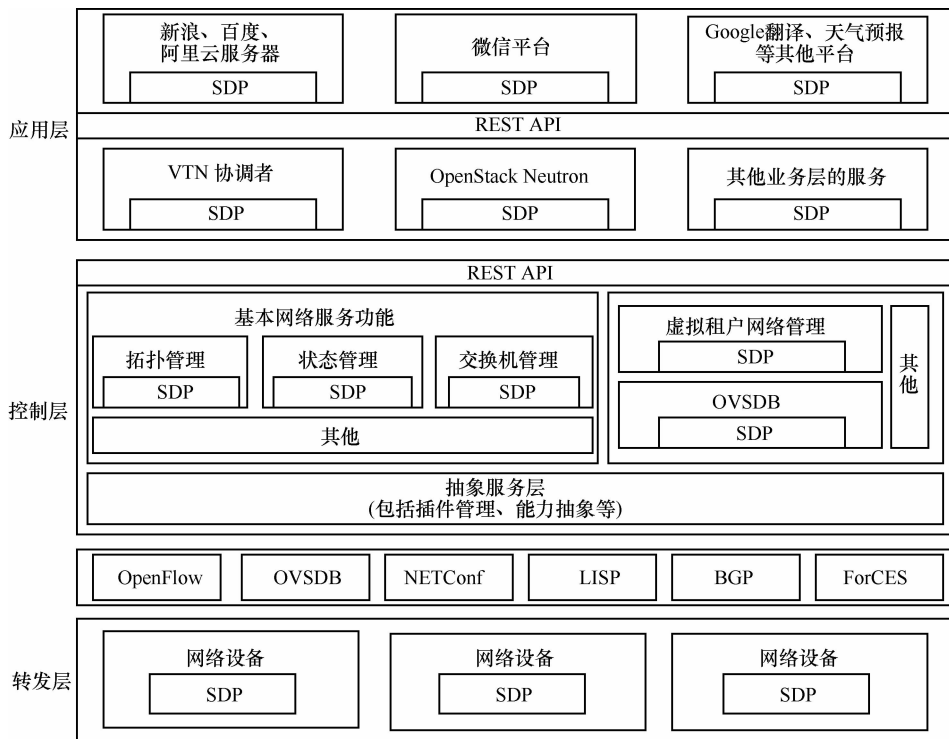


图 10 SDN 应用体系框架

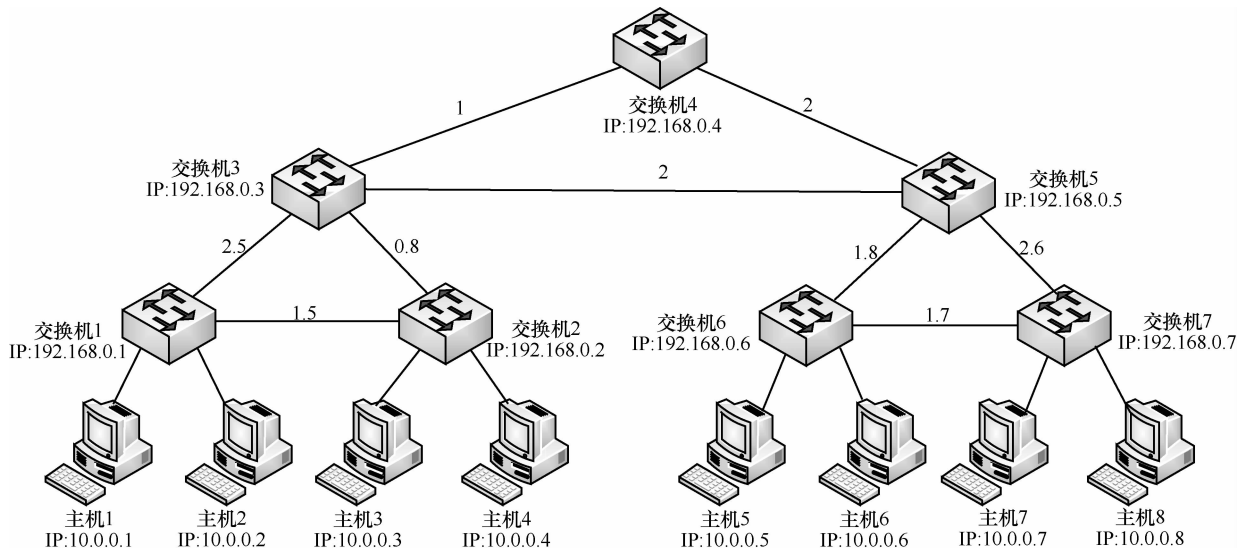


图 11 底层物理拓扑设计

表 2 系统测试用例

测试用例	说明
用例名称	基于 SDP 的虚拟租户网络整体测试
用例描述	在主机 1 与主机 8 之间建立一条最合理的通信链路
用例入口	在 Ubuntu 终端界面上,通过 curl 进行创建
测试场景	在主机 1、主机 3、主机 5 和主机 8 之间创建一条虚拟租户网络
测试步骤	通过输入 curl 命令创建相应价格的 VTN 命令
预期结果	创建成功
备注	价格符合要求,创建成功

5.2 测试过程及结果

实验采用 Ubuntu11.10 操作系统,系统的整体测试环境包括 OpenDaylight 控制器和 Mininet 仿真环境。测试分为以下 4 个部分。

(1)在 Ubuntu 终端创建虚拟租户网络。

若返回“HTTP/1.1 200 OK”,表示创建成功。创建 VTN 成功如图 12 所示。

```
yao@yao-Founder-PC:~$ curl -v -POST -H 'content-type:application/json' -H 'username:admin' -H 'password:admin' -H 'ipaddr:127.0.0.1' -d '{"vtn":{"hosts":["10.0.0.1", "10.0.0.3", "10.0.0.5", "10.0.0.8"], "price":10.0}}' http://127.0.0.1:8080/vtn-sdp/vtn
* About to connect() to 127.0.0.1 port 8080(#0)
* Trying 127.0.0.1...
* Connected to 127.0.0.1(127.0.0.1) port 8080(#0)
> POST /vtn-sdp/vtn HTTP/1.1
> User-Agent: curl/7.29.0
> Host: 127.0.0.1:8080
> Accept: */*
> content-type: application/json
> username: admin
> password: admin
> ipaddr:127.0.0.1
> Content-length: 128
* upload completely sent off: 128 out of 128 bytes

< HTTP/1.1 200 OK
< Server: Apache-Coyote/1.1
< Content-Type: application/json;charset=UTF-8
< Content-Length: 0
< Date: Mon, 11 Oct 2014 14:14:03 GMT
* Connection #0 to host 127.0.0.1 left intact
```

图 12 创建 VTN 成功示意

(2)通过 Mininet 终端验证结果的正确性。

Mininet 终端测试结果验证界面如图 13 所示。打开 Mininet 终端,在终端输入 pingall 指令,可以看到,主机 1、主机 3、主机 5 和主机 8 之间可以相互通信。

(3)通过基于 SDP 的虚拟租户网络日志文件验证结

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> X h3 X h5 X X h8
h2 -> X X X X X X X
h3 -> h1 X X h5 X X h8
h4 -> X X X X X X X
h5 -> h1 X h3 X X X h8
h6 -> X X X X X X X
h7 -> X X X X X X X
h8 -> h1 X h3 X h5 X X
```

图 13 Mininet 终端测试结果验证界面

果的正确性。

VTN 日志文件如图 14 所示,在主机 1、主机 3、主机 5 和主机 8 之间存在 3 条虚拟租户网络,根据 VTN 的定价机制可得到 3 条虚拟租户网络 (VTN1、VTN2、VTN3),对应的价格为:10.7、9.7 和 8.6,当客户端发出创建 VTN3 命令时,调用协商算法 (BaseConsult) 进行协商;然后,调用南向接口创建对应的 vBridge、interface 以及 portmap;最后,将创建成功的信息返回给用户。

```
2014-10-11 14:14:01 org.opendaylight.vtn.sdp.endpoint.VTNAPI createVTN
Info: hosts:10.0.0.1, 10.0.0.3, 10.0.0.5, 10.0.0.8 price:10.0
2014-10-11 14:14:05 org.opendaylight.vtn.sdp.pricing.impl.BasePricing setPricing
Info: {"vtn": [{"vtn_name": "VTN1", "price": 10.7}, {"vtn_name": "VTN2", "price": 9.7}, {"vtn_name": "VTN3", "price": 8.6}]}
2014-10-11 14:14:10 org.opendaylight.vtn.sdp.consult.BaseConsult consult
Info Select VTN3 Successfully!
2014-10-11 14:14:15 org.opendaylight.vtn.sdp.consult.southbound.BaseSouthBound post
Info HTTP/1.1 200 OK
Info {"vbridge": [{"vbridge_name": "vbridge1", "if_name": "if1", "logical_port_id": "PP-00-00-00-00-00-00-01-s1-eth1"}, {"vbridge_name": "vbridge2", "if_name": "if1", "logical_port_id": "PP-00-00-00-00-00-00-02-s2-eth2"}, {"vbridge_name": "vbridge3", "if_name": "if1", "logical_port_id": "PP-00-00-00-00-00-00-03-s3-eth2"}, {"vbridge_name": "vbridge4", "if_name": "if1", "logical_port_id": "PP-00-00-00-00-00-00-05-s5-eth1"}, {"vbridge_name": "vbridge5", "if_name": "if1", "logical_port_id": "PP-00-00-00-00-00-00-07-s7-eth1"}, {"vbridge_name": "vbridge6", "if_name": "if1", "logical_port_id": "PP-00-00-00-00-00-00-09-s9-eth2"}, {"vbridge_name": "vbridge7", "if_name": "if1", "logical_port_id": "PP-00-00-00-00-00-00-11-s11-eth1"}, {"vbridge_name": "vbridge8", "if_name": "if1", "logical_port_id": "PP-00-00-00-00-00-00-13-s13-eth2"}]}
Info Post OK!
2014-10-11 14:14:18 org.opendaylight.vtn.sdp.endpoint.VTNAPI createVTN
Info: VTNPrice: 8.6
Info: Create VTN Successfully!
```

图 14 VTN 日志文件

(4)通过图形界面验证结果的正确性。

基于 SDP 的虚拟租户网络系统的 UI 界面如图 15 所示。各个链路上的数值对应当前链路的价格,加阴影的主机是当前虚拟租户网络所覆盖的主机。VTN1、VTN2 和 VTN3 为第 1 步创建 VTN 时显示的 3 条虚拟租户网络。由图 15 可知,VTN3 对应的价格最便宜,所以用户可以选择创建 VTN3 来实现自己的虚拟租户网络系统。

6 结束语

SDN 因其具有转发与控制分离、集中控制和开放接口功能而被广泛应用。本文首先介绍了 SDN 的产生背景和发展现状;在此基础上,通过比较 SDN 和云计算应用的区别和特点,借鉴云计算的经济模型,提出了 SDP 的概念;并在 SDN 应用体系框架的基础上,将 SDP 机制与 SDN 应用体系框架结合起来,提出了基于 SDP 的 SDN 应用和 SDN 应用体系框架。在本文所提出的 SDN 应用中,SDN 应用具有经济模型,可以对网络资源进行动态定价;而资源提供者和资源消耗者可以通过协商机制对资源的价格进行协商,从而从用户的角度调度网络资源。实验证明当前可以实现基于 SDP 的虚拟租户网络。在未来的工作中,将针对 SDN 架构中的每一层,进一步完善和实现 SDP 和 SDN 应用。

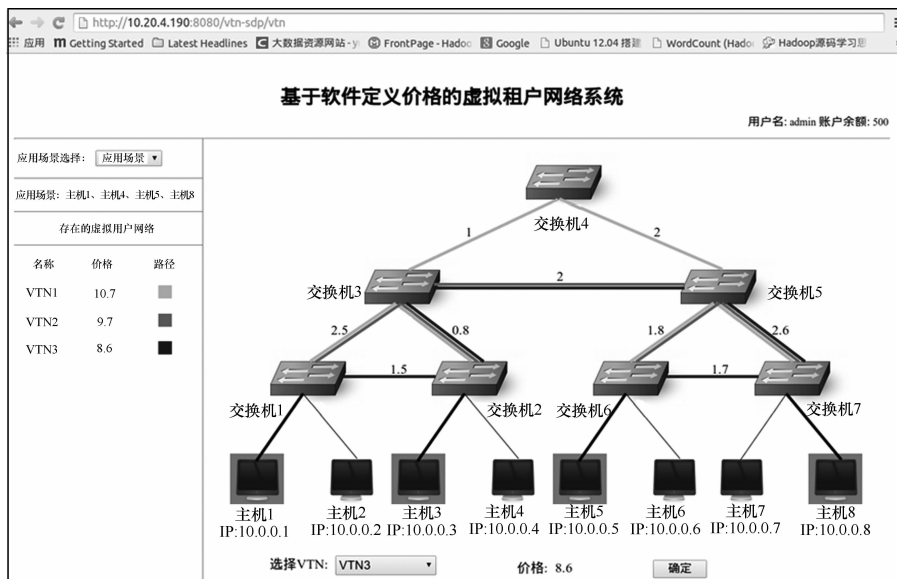


图 15 VTN 图形界面

参考文献

- 1 Stanford University. Clean slate program. <http://cleanslate.stanford.edu>, 2006
- 2 McKeown N. Software-defined networking. Proceedings of IEEE INFOCOM, Rio de Janeiro, Brazil, 2009
- 3 张朝昆, 崔勇, 唐嵩祎等. 软件定义网络(SDN)研究进展. 软件学报, 2015, 26(1): 62~81
Zhang C K, Cui Y, Tang H Y, et al. State of the art survey on software defined networking. Journal of Software, 2015, 26(1): 62~81
- 4 郑毅, 华一强, 何晓峰等. SDN 的特征、发展现状及趋势. 电信科学, 2013, 29(9): 102~107
Zheng Y, Hua Y Q, He X F, et al. Characteristics, development and future of SDN. Telecommunications Science, 2013, 29(9): 102~107
- 5 雷葆华, 王峰, 王茜等. SDN 核心技术剖析和实战指南. 北京: 电子工业出版社, 2013
Lei B H, Wang F, Wang Q, et al. Deciphering SDN: Core Techniques and Practical Guide. Beijing: Publishing House of Electronics Industry, 2013
- 6 Network Functions Virtualization: an Introduction, Benefits, Enablers, Challenges & Call for Action. White Paper, 2012
- 7 OpenDaylight. <http://www.opendaylight.org/>, 2015
- 8 Nadeau T D, Gray K. SDN: Software Defined Networks. Sebastopol: Oreilly Media, 2013
- 9 赵慧玲. SDN 应用之资源管理平台. <http://network.51cto.com/art/201312/424814.htm>, 2013
Zhao H L. Resource management platform of SDN application. <http://network.51cto.com/art/201312/424814.htm>, 2013
- 10 赵慧玲, 史凡. SDN/NFV 的发展与挑战. 电信科学, 2014, 30(8): 13~18
Zhao H L, Shi F. Development and challenge of SDN/NFV. Telecommunications Science, 2014, 30(8): 13~18
- 11 Huang S, Griffioen J, Calvert K L. Network hypervisors: enhancing SDN infrastructure. Computer Communications, 2014, 46(6): 87~96
- 12 宋平, 刘轶, 刘驰等. 一种支持细粒度并行的 SDN 虚拟化编程框架. 软件学报, 2014, 25(10): 2220~2234
Song P, Liu Y, Liu C, et al. Fine-grained parallel SDN virtualization programming framework. Journal of Software, 2014, 25(10): 2220~2234
- 13 王淑玲, 李济汉, 张云勇等. SDN 架构及安全性研究. 电信科学, 2013, 29(3): 117~122
Wang S L, Li J H, Zhang Y Y, et al. Research on SDN architecture and security. Telecommunications Science, 2013, 29(3): 117~122
- 14 黄孙亮. SDN 的应用场景分析与探讨. http://www.zte.com.cn/cndata/magazine/zte_technologies/2013/7_2013/magazine/201307/t20130715_402140.html, 2013
Huang S L. The analysis and discussion of SDN application scenarios. http://www.zte.com.cn/cndata/magazine/zte_technologies/2013/7_2013/magazine/201307/t20130715_402140.html, 2013
- 15 第一届全国高校软件定义网络(SDN)应用创新开发大赛. <http://www2.scut.edu.cn/s/139/t/153/91/59/info102745.htm>, 2014
The 1st innovation and development competition of SDN application in national colleges. <http://www2.scut.edu.cn/s/139/t/153/91/59/info102745.htm>, 2014

- 16 陈康, 郑纬民. 云计算: 系统实例与研究现状. 软件学报, 2009, 20(5): 1337~1348
Chen K, Zheng W M. Cloud computing: system instances and current research. Journal of Software, 2009, 20(5): 1337~1348
- 17 黎春兰, 邓仲华, 张文萍. 云服务的定价策略分析. 图书与情报, 2013(1): 36~41
Li C L, Deng Z H, Zhang W P. Pricing the cloud services. Library and Information, 2013(1): 36~41
- 18 张云勇, 李素粉, 吴俊等. 面向云服务提供商的服务选择方法研究. 通信学报, 2012(9): 66~76
Zhang Y Y, Li S F, Wu J, et al. Research on the cloud services provider-oriented services selection method. Journal on Communications, 2012(9): 66~76
- 19 谢剑. 云计算服务产品的组合双向拍卖模型研究 (硕士学位论文). 浙江工商大学, 2013
Xie J. Study on the model of combinatorial double auction for the cloud computing services and products (master dissertation). Zhejiang Gongshang University, 2013
- 20 诸葛斌, 邓丽, 戴国伟等. 基于双边市场多归属结构的 SDN 资源管理机制. 电信科学, 2014, 30(5): 55~64
Zhuge B, Deng L, Dai G W, et al. SDN resource management mechanism based on bilateral market's multi-homed architecture. Telecommunications Science, 2014, 30(5): 55~64
- 21 梁宁宁, 兰巨龙, 程国振等. 基于拍卖博弈的可重构服务承载网动态构建算法. 电信科学, 2015, 31(5)
Liang N N, Lan J L, Cheng G Z, et al. Dynamic auction game-based reconfigurable service carrying network construction algorithm. Telecommunications Science, 2015, 31(5)
- 22 杨明. 基于组合双向拍卖和信任的网格资源分配与定价算法研究及应用 (硕士学位论文). 北京邮电大学, 2010
Yang M. Research and application on the algorithm of grid resource allocation and pricing based on combinatorial double auction and trust (master dissertation). Beijing University of Posts and Telecommunications, 2010
- 23 李俊武. 云计算网络珠玑. 北京: 电子工业出版社, 2015
Li J W. Cloud Computing Pearls. Beijing: Publishing House of Electronics Industry, 2015
- 24 张卫峰. 深度解析 SDN: 利益、战略、技术、实践. 北京: 电子工业出版社, 2014
Zhang W F. SDN: Interest, Strategy, Technology and Practice. Beijing: Publishing House of Electronics Industry, 2014
- 25 李俊武. Neutron 结合 SDN 的架构分析. <http://www.sdnlab.com/8583.html>, 2015
Li J W. Architecture analysis of the Neutron combined SDN. <http://www.sdnlab.com/8583.html>, 2015
- 26 中外对比: 国内云计算平台的三大特点. <http://news.expeak.com/cloud/2012/0904/3526.html?1346829435>, 2012
Chinese-foreign contrast: the three characteristics of domestic cloud computing platform. <http://news.expeak.com/cloud/2012/0904/3526.html?1346829435>, 2012
- 27 胡永凯, 左保河. 基于 Maven 的 J2ME 项目管理插件的开发. 电脑与电信, 2009(3): 54~56
Hu Y K, Zuo B H. Development of Maven plugin for J2ME project. Computer and Telecommunication, 2009(3): 54~56
- 28 张俊. 用 Maven 简化你的项目管理. 程序员, 2006(9): 124~125
Zhang J. Simplify project using Maven. Programmer, 2006(9): 124~125
- 29 David E, Jon K. Networks, crowds and markets: reasoning about a highly connected world. Journal of the Royal Statistical Society, 2010, 175(4)
- 30 OpenDaylight Virtual Tenant Network (VTN). <https://github.com/opendaylight/docs/blob/stable/lithium/manuals/user-guide/src/main/asciidoc/vtn/vtn-user.adoc>, 2015

[作者简介]



诸葛斌,男,博士,浙江工商大学信息与电子工程学院副院长、教授,主要研究方向为网络与通信技术、互联网技术和网络安全。



王保霞,女,浙江工商大学硕士生,主要研究方向为软件定义网络和未来网络体系结构。



王怡宁,女,浙江大学在读,主要研究方向为计算机网络、算法与编程和软件定义网络。

吴春明,男,博士,浙江大学计算机科学与技术学院教授、博士生导师,主要研究方向为互联网体系结构、可重构网络、软件定义网络和动态网络主动防御技术。

姚敏辉,男,浙江工商大学硕士生,主要研究方向为软件定义网络和未来网络体系结构。