网络功能虚拟化技术研究进展

王进文[1] 张晓丽[2),3),4)] 李 琦[2),3)] 吴建平[1),2),4)] 江 勇[1),3)]

[1)](清华大学清华-伯克利深圳学院 广东 深圳 518055)

[2)](清华大学网络科学与网络空间研究院 北京 100084)

[3)](清华大学深圳研究生院 广东 深圳 518055)

[4)](清华大学计算机科学与技术系 北京 100084)

摘 要 企业通常部署各种网络功能设备来实现企业网络所需的网络功能.例如,防火墙和入侵检测系统可以加强企业网络的安全性;缓存代理和广域网优化器可以提升企业网络的性能.然而,企业部署、更新和维护网络功能设备需要大量开销.不同网络功能设备之间的差异使得企业需要庞大的专业团队来管理网络设备.物理设备固定的位置和处理能力使得企业网络无法有效解决网络拥塞带来的设备失效问题.随着企业网络规模的增长,网络功能设备管理、维护和更新产生的开销急剧上升.面对日益增长的网络功能设备运维开销和管理难度,网络功能虚拟化(Network Function Virtualization,NFV)技术提出将网络功能和物理硬件设备解耦,通过在通用商用服务器上部署和管理网络功能,为企业降低了网络设备管理难度,减少了网络功能设备开销,提供了灵活的网络服务部署策略,例如,动态回收/扩展.尽管 NFV 技术能为企业带来便捷和利益,但是实现一个实用而高效的 NFV 系统存在很多挑战.针对 NFV 系统实现中存在的问题和挑战,学术界和工业界已投入大量精力对 NFV 技术进行研究和尝试.该文首先介绍欧洲电信标准协会制定的 NFV 技术标准结构,并根据其分类总结 NFV 系统实现中存在的问题和挑战,将当前研究成果分为网络功能虚拟化、网络功能虚拟化设施和管理、网络编排三个部分.软件定义网络(Software Defined Network,SDN)和 NFV 技术可以相互弥补促进,该文对其之间的关系进行了研究.然后,该文着重从虚拟化网络功能(Virtual Network Function,VNF)构建及运行环境优化、NFV 管理系统设计及优化、策略实施与验证、资源分配和迁移策略、NFV 负载均衡和状态管理技术、NFV 架构中的安全问题几个方面来深入分析 NFV 技术当前的学术研究成果.最后介绍了 NFV 技术在云计算、移动通讯以及家庭网络中的应用场景实例,同时对 NFV 技术进行总结并展望未来研究发展方向.

关键词 网络功能虚拟化技术;网络功能;虚拟化;云计算;网络安全

中图法分类号 TP302 **DOI 号** 10.11897/SP.J.1016.2019.00415

Network Function Virtualization Technology: A Survey

WANG Jin-Wen[1] ZHANG Xiao-Li[2),3),4)] LI Qi[2),3)] WU Jian-Ping[1),2),4)] JIANG Yong[1),3)]

[1)](*Tsinghua-Berkeley Shenzhen Institute, Tsinghua University, Shenzhen, Guangdong* 518055)

[2)](*Institute for Network Science and Cyberspace, Tsinghua University, Beijing* 100084)

[3)](*Graduate School at Shenzhen, Tsinghua University, Shenzhen, Guangdong* 518055)

[4)](*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

Abstract Various middleboxes have been deployed to enable diversified functionalities in enterprise networks. For example, Firewalls and Intrusion Detection Systems are used to enhance the security of the network, and Cache Proxies and WAN Optimizations are used to improve the performance of the enterprise networks. However, it is painful for enterprises to deploy, update, or maintain such physical devices in their enterprise networks. Moreover, because of the discrepancy

收稿日期:2017-11-06;在线出版日期:2018-05-15.本课题得到国家重点研发计划(2016YFB0800102)、国家自然科学基金(61572278,U1736209)、深圳市基础研究基金(JCYJ20170307153259323)资助.**王进文**,男,1994 年生,硕士研究生,主要研究方向为 NFV、网络安全、系统安全、可信计算.E-mail:wangjw16@mails.tsinghua.edu.cn.**张晓丽**,女,1993 年生,博士研究生,主要研究方向为 SDN、NFV、网络安全.**李 琦**(通信作者),男,1979 年生,博士,副教授,中国计算机学会(CCF)高级会员,主要研究方向为互联网和云安全、移动安全和大数据安全.E-mail:qi.li@sz.tsinghua.edu.cn.**吴建平**,男,1953 年生,博士,教授,博士生导师,中国工程院院士,主要研究领域为网络协议测试、网络管理、网络体系结构等.**江 勇**,男,1975 年生,博士,教授,博士生导师,主要研究领域为互联网体系结构、下一代互联网.

between network function devices of the different manufacturers, enterprises need to hire large management teams with extensive knowledge to manage the network functions devices. Because of the fixed location and network traffic process capacity of the hardware network function devices, the network traffic congestion always causes the failure of the network functions. The overhead of middlebox management, maintenance and update increase significantly as enterprise networks grow. Fortunately, the emerging Network Function Virtualization (NFV) technology can address these issues by decoupling the physical network equipment from the network functions that run on them. With the NFV technology, an enterprise can efficiently reduce complicated network function management and the cost of deploying network function equipment by deploying network functions in commercial servers. It can also enable flexible service deployment strategies for the enterprises, e. g. , dynamically scale in/out. Although NFV can bring significant benefits to enterprises, there are still many challenges in developing practical and efficient NFV. In order to solve such problems, researchers in academia and industry have devoted to the study of the NFV technology. This paper presents the first systematic study of the literature of the NFV technology. Firstly, we review the standard architecture of NFV established by the European Telecom Standards Institute Industry Specification Group for NFV (ETSI ISG NFV). We classify and summarize the problems and challenges in the developing of NFV systems based on the standard architecture of NFV. The study can be classified into the following three categories: Network Function Virtualization, Network Function Virtualization Infrastructure (NFVI), and Management And Network Orchestration (MANO). Furthermore, Software Defined Network (SDN) is a new type of network architecture which are designed to achieve flexible and intelligent network traffic control by decoupling the control plane and data plane of network devices, NFV and SDN can complement each other in various aspects, we study the relationship between the NFV and SDN. Secondly, we systematically study building blocks of NFV, i. e. , virtual network function (VNF) construction, and the running environment optimization in NFV, the design of NFV management system and its optimization, policy enforcement and verification, resource allocation and migration, load balance and state management technology, and NFV security. Thirdly, the NFV technology has been deployed in different kinds of industry fields, we show the practical deployment issues of NFV by discussing the application scenario cases of NFV, e. g. in cloud computing, mobile communication, and home network. Finally, we summarize the advantages and shortcomings of NFV by comparing the NFV technology and classical physical network function devices and present the future research directions of the NFV technology from six aspects based on the standard architecture of ETSI NFV.

Keywords network function virtualization technology; network function; virtualization; cloud computing; network security

1 引 言

现代企业通过部署各种网络功能设备来提高企业网络的安全性(例如,防火墙(Firewall)、入侵检测系统(Intrusion Detection System,IDS))和性能(例如,缓冲代理(Cache Proxy)、广域网优化器(WAN Optimization)).随着企业规模的增长,企业网络中部署的网络功能设备数量大幅增加.根据一份针对企业中网络功能设备部署数量的调研报告[1],在 58 家规模不一的企业中,拥有超过 10 万台主机的大型企业网络中平均部署 1946 台不同功能的网络功能设备以及 2850 台三层路由器,拥有不超过 1000 台主机的小型网络平均部署 10 台网络功能设备,7 台三层路由器.网络功能设备在企业网中的平均数量达到三层路由设备的 70%.在大量网络功能设备部署的企业网络环境中,出现以下问题:

(1)部署网络功能设备开销大.企业需要支付

巨额的资金来对企业网络中的网络功能设备进行部署，更新和维护；

(2) 网络功能设备管理困难. 企业网络中不同的网络功能设备通常来自于不同的供应商，其实现差异性较大，设备策略配置复杂，因而管理网络中不同的网络功能设备需要广泛的专业知识和庞大的管理团队；

(3) 网络功能设备失效率高. 在企业网络中，由于物理机器故障、电路故障、人为策略配置错误以及网络流量过载等情况引起的网络功能失效率高，所以调试解决此类问题复杂.

针对现代企业网存在的以上问题，欧洲电信标准协会(European Telecom Standards Institute, ETSI)提出了网络功能虚拟化(Network Function Virtualization, NFV). NFV 通过在工业界标准高性能服务器、交换机和存储设备上发展标准虚拟化技术来构建和部署网络功能，旨在用软件实现可在行业标准服务器上运行的网络功能，并可根据需求动态部署在网络中不同的位置而无需重新安装新的专用硬件设备. 如图 1 所示，通过使用 NFV 技术，传统的网络功能设备(例如，IDS、入侵防御系统(Intrusion Prevention System, IPS)、Firewall 和网络地址转换器(Network Address Translation, NAT)等)都能以虚拟化网络功能(Virtual Network Function, VNF)的形式在行业标准服务器上初始化并无需安装新的专有硬件设备.

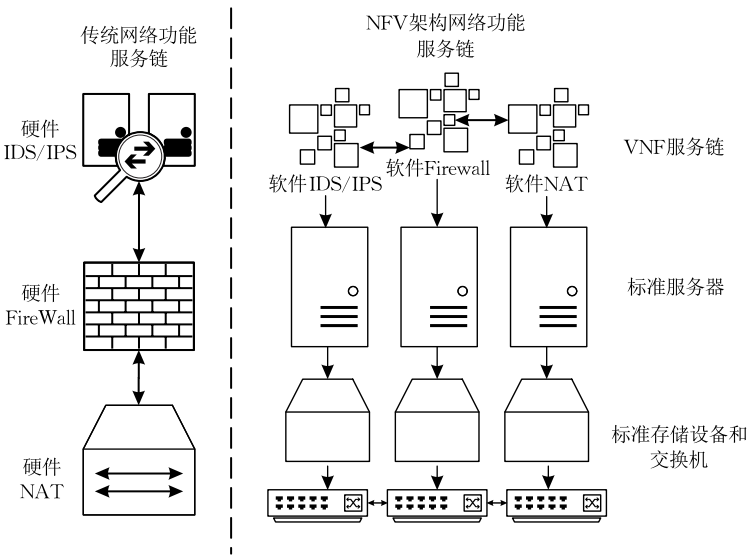

图 1　传统硬件网络功能与 NFV

NFV 技术以其部署资金开销小、管理简便与部署灵活等优势，成为了工业界和学术界关注的焦点，带动了工业界和学术界的踊跃尝试和积极研究，产生了大量的理论和实践成果. 本文首先介绍 NFV 技术的标准结构，然后着重从 VNF 构建及运行环境优化、NFV 管理系统设计及优化、策略实施与验证、资源分配及迁移策略、NFV 负载均衡和状态管理技术、NFV 架构中的安全问题六个方面来分析 NFV 技术在学术界的相关研究成果. 最后介绍 NFV 技术最新的应用场景，总结 NFV 技术并展望 NFV 技术未来研究方向.

2　NFV 标准结构介绍

2012 年 10 月，多家世界领先的内容服务提供商联合撰写了 NFV 的白皮书[2]，标志着 NFV 概念的正式诞生. 同年 11 月，这些服务提供商中的 7 家选择了 ETSI 作为 NFV 行业规范的制定组织(Industry Specification Group for NFV, ETSI ISG NFV). ETSI 的成员数目在短短两年内增长至包括 37 家世界主要服务提供商、电信运营商和 IT 制造商在内的 245 家公司. 在所有 ETSI 专家的努力下，ETSI 以两年为一个阶段，为 NFV 的规范和普及不断努力. ETSI 在 2013～2014 年预规范阶段发布了包含 NFV 使用场景用例、NFV 标准结构①、NFV 术语和虚拟化需求在内的一系列 NFV 标准，为 NFV 技术的标准化进程树立第一块里程碑. 在 2015～2016 年 NFV 标准化第二阶段内，ETSI 发布了第 2 版 NFV，其主要通过选择 NFV 系统中的关键组成部分，并

① NFV Architectural Framework. http://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf, 2013,10

在需求、接口以及数据信息模型方面对其进行详细的规范说明，实现了保证 NFV 系统中设备和服务端对端互联的规格标准化. 其中主要包括虚拟资源管理、NFV 生命周期管理、网络服务说明等多个方面. 2017 年至今，NFV 标准化的第三个阶段正在进行，ETSI 正在为第 3 版 NFV 的发布而努力，届时，完整的第 3 版 NFV 将为 NFV 技术的安全、服务计费、自动化部署和管理等方面制定新的规范和标准.

NFV 作为一种虚拟化技术，通过将软件网络功能部署在稳定的商用计算资源平台上，避免了传统的异构硬件网络功能部署过程中复杂的连接配置工作，再结合灵活的负载均衡等管理机制，有效解决了传统网络功能硬件设备存在的设备价格高、管理困难以及由连接配置和网络流量过载等引起的失效率高问题. 但使用不同的方式实现 NFV 技术会形成不同的 NFV 体系框架，混乱的 NFV 体系框架会阻碍 NFV 技术的发展和普及. 因此，ETSI 对 NFV 结构框架进行标准化，旨在形成统一的 NFV 标准结构，简化 VNF 的开发，促进 NFV 的普及和部署. 如图 2 所示，NFV 架构主要包含虚拟化网络功能、网络功能虚拟化设施以及网络功能虚拟化管理和编排三个部分.

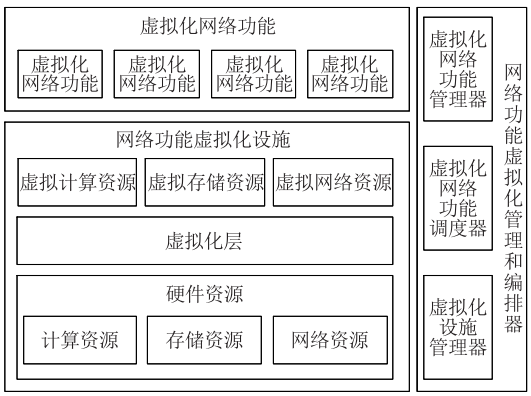

图 2 ETSI NFV 标准结构(参见本文第 3 页脚注①)

2.1 虚拟化网络功能

网络功能①(Network Function，NF)是传统网络基础设施中的功能模块，它具有固定的内部功能以及良好定义的外部接口. 例如，家庭网络中的住宅网关(Residential Gateway，RGW)、传统企业网络中的用于增强企业网安全性能的 Firewall、IDS、IPS 以及用于提升企业网络性能的代理(Proxy)、缓存(Cache)和 WAN Optimization 等.

与 NF 对应，VNF 就是能部署在虚拟资源上的各类软件 NF. 不同的 VNF 通常由相互独立的软件开发商根据 NFV 标准进行开发. 单个的 VNF 可以由多个内部组件组成，因此，单个 VNF 可以分布在多个虚拟机(Virtual Machine，VM)上，不同的虚拟机托管不同的 VNF 组件.

2.2 网络功能虚拟化设施

网络功能虚拟化设施(Network Functions Virtualization Infrastructure，NFVI)是提供 NFV 部署、管理和运行所需环境的软硬件结构总称，其中包括硬件资源、虚拟化层以及虚拟化资源.

硬件资源主要包括由计算硬件设备组成的计算资源、存储设备构成的存储资源以及由节点和连接链路组成的网络资源. 这些硬件资源通过虚拟化层(例如，VM、虚拟机管理器等)向 VNF 提供计算处理能力、存储能力以及网络连接性.

虚拟化层主要负责抽象硬件资源，并将 VNF 和底层硬件资源解耦. VNF 通过使用经过虚拟化层抽象和逻辑切分的物理资源，可以运行在逻辑独立的物理硬件资源之上.

虚拟化资源是对计算资源、网络资源和存储资源的抽象. 与硬件资源相对应，虚拟化资源包括虚拟化计算资源、虚拟化存储资源和虚拟化网络资源. 在数据中心环境中，虚拟化计算资源和虚拟化存储资源通常以虚拟机的形式向上层 VNF 提供计算资源和存储资源. 虚拟化网络资源则通常表示为虚拟节点和虚拟网络链路. 其中虚拟节点是具有托管或路由功能的软件(例如，VM 中的操作系统)；而虚拟链路则为虚拟节点之间提供相互之间的连接性，使虚拟节点拥有可以动态变化的物理链路属性[3].

2.3 网络功能虚拟化管理和编排

网络功能虚拟化管理和编排(Management And Network Orchestration，MANO)部分主要向 NFV 平台提供协调控制所有 VNF 所需要的功能和操作(例如，对 VNF 和虚拟资源的配置)，使所有 VNF 能够有序运行. MANO 主要包含虚拟化设施管理器、虚拟化网络功能管理器(VNF 管理器)和虚拟化网络功能调度器(VNF 调度器)三个部分.

虚拟化设施管理器主要负责监控 NFVI 中的资源使用情况，并对 NFVI 中软/硬件资源进行生命周期管理和分配调度，例如，对 CPU 资源的分配、网络链路带宽的分配等. VNF 管理器主要负责 VNF

① Terminology for Main Concepts in NFV. http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_NFV003v010201p.pdf 2014，12

生命周期的管理，例如，VNF 的初始化、更新、扩展、终止等. VNF 调度器主要负责协调 VNF 管理器和虚拟化设施管理器来实现网络功能服务链在虚拟化设施上的部署实施.

此外，MANO 中还设置有数据库，并提供了一系列标准接口. 数据库用来存储由网络管理员提供的 VNF 部署规则和 VNF 生命周期属性等信息. 标准接口用于实现 MANO 中不同组件之间通信交流以及 MANO 和传统网络管理系统（例如，操作支持系统（Operation Support System，OSS）以及商务支持系统（Business Support System，BSS））之间的协调合作. 通过对数据库和这些接口的结合使用，MANO 可以实现对 VNF 和传统设备的协调管理.

2.4　NFV 与 SDN

2.4.1　NFV 与 SDN 的关系

NFV 是一种用软件方式来实现传统硬件网络功能的技术. 其通过将网络功能与专有硬件分离，旨在实现网络功能的高效配置和灵活部署，减少网络功能部署产生的资金开销、操作开销、空间以及能源消耗. 软件定义网络[4]（Software Defined Network，SDN）是一种新型的网络架构，其通过解耦网络设备的控制平面和数据平面，旨在实现灵活、智能的网络流量控制. 作为两种独立的新兴网络技术，SDN 和 NFV 之间存在着相互弥补，互相促进的关系.

为了满足多样化的服务要求，SDN 数据层设备需要进行通用流量匹配和数据包转发，SDN 交换机的成本和复杂性随之增加. 另外，目前 SDN 架构缺乏对异构 SDN 控制器间交互的支持，使之无法提供灵活的跨自治域端对端服务. SDN 在数据层面和控制层面存在的软件网络架构和硬件网络设施之间的紧耦合限制了 SDN 更加广泛地应用. 目前仅在数据层面和控制层面的解耦已经无法有效地避免上述问题，软件服务功能和硬件网络设施的进一步解耦才能使得 SDN 得以更加广泛地应用. 因此，在 SDN 中使用 NFV 技术，可以为 SDN 提供更加灵活的网络服务. 例如，使用 NFV 技术实现虚拟化 SDN 控制器[5]，通过一致性接口实现异构虚拟 SDN 控制器之间的交互、使用 NFV 技术实现虚拟化 SDN 数据层面可以根据不同的服务要求实现灵活的流量匹配和数据包转发.

在 ETSI NFV 架构中，NFV 管理和编排层是整个 NFV 平台有序、高效运行的保障. 在 NFV 平台动态网络环境中，需要复杂的控制和管理机制来对虚拟资源和物理资源进行合理的分配和管理. 因此可编程的网络控制不可或缺. SDN 结合 MANO 可以高效地控制网络流量转发，向 NFV 平台提供 VNF 之间的可编程网络连接，以此来实现高效灵活的流量调度[6-7].

2.4.2　NFV 与 SDN 结合的网络系统架构

NFV 与 SDN 相结合的网络系统架构如图 3 所示，其主要包含转发设备、NFV 平台以及逻辑控制模块.

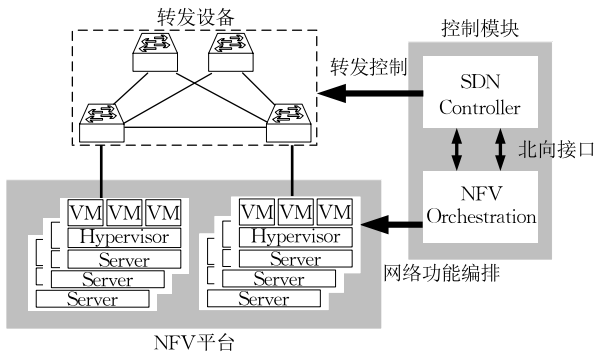

图 3　SDN NFV 结合的网络系统架构

转发设备主要负责数据包的转发，其中转发规则由 SDN 控制器决定，并以流表项的形式下发到交换机. 中心控制器和分布式转发设备一般采用高效的协议（例如，OpenFlow[8]）作为相互通信的接口.

NFV 平台利用标准商业服务器以低成本拥有高带宽的 NF. 托管 NF 的 VM 由运行在服务器上的虚拟机管理器进行支持. 因此，网络管理员只需要提供纯软件的 NF，NFV 平台即可提供可定制、可编程数据层处理功能，例如，Firewall、IDS、Proxy 等.

逻辑控制模块由 SDN 控制器和 NFV 编排系统组成. NFV 编排系统负责管理 VNF，SDN 控制器负责管理转发设备并通过标准接口和 NFV 编排系统进行交互. 控制模块通过网络管理员定义的网络拓扑和策略需求，使用最佳的资源分配方案和最优路由路径. 其中，资源分配方案由 NFV 编排系统计算执行，SDN 控制器通过在转发设备上安装转发规则来实现流量调度.

3　NFV 技术研究概况

在 NFV 技术提出之后，学术界已对其展开了大量研究，以 ETSI 提出的 NFV 标准结构层次为依据，现存的 NFV 研究成果可以分为以下 6 个领域：VNF 构建及运行环境优化、NFV 管理系统设计及优化、策略实施与验证、资源分配与迁移策略、NFV 负载均衡和状态管理技术以及 NFV 架构的安全问题.

如图 4 所示，在虚拟化网络功能层面，由于 VNF 开发复杂、出错率高、开发周期长、学术界高度关注 VNF 的高效构建方式. 因此，高效的 VNF 构建，成为了 NFV 虚拟化网络功能层面研究的目标之一. 此外，网络流量的过载以及 VNF 运行时错误引起的 VNF 失效会对企业造成严重的损失. 因此，实现灵活、鲁棒的负载均衡和状态管理机制成为虚拟化网络功能层面研究的第二个重要目标.

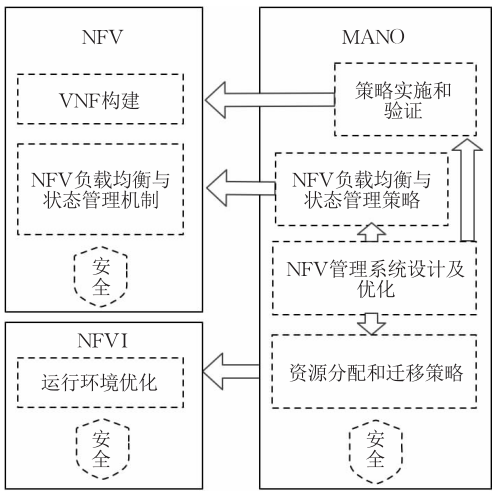

图 4 NFV 研究内容体系结构

NFV 主要依赖虚拟化技术实现软件网络功能在通用服务器上的部署. 因此，对虚拟化技术的研究成为 NFVI 层的研究核心. 如何优化运行环境(NFVI 层的 VM 和虚拟机管理器)，使得虚拟化网络功能层面的 VNF 能够高效地处理网络流量成为了 NFVI 层的研究目标. 由于 NFVI 主要为 VNF 提供运行环境，因此，对 VNF 构建和 VNF 运行环境的研究通常在相关工作中同步进行.

MANO 在 NFV 系统中主要负责对 VNF 和 NFVI 的管理与协调. 因此，MANO 层的研究内容主要包括 NFV 管理系统设计及优化、策略实施和验证、资源分配和迁移策略以及负载均衡策略. NFV 管理系统设计及优化主要负责 MANO 层对 NFV 和 NFVI 的协调管理及系统的高效整合和实现，其涉及 MANO 通过策略实施和验证策略、NFV 负载均衡策略以及资源分配和迁移策略与 NFV 层以及 NFVI 层的双向交互；其中策略实施和验证负责 MANO 对 NFV 的配置和管理策略的实现，主要涉及 MANO 与 NFV 层的交互；资源分配和迁移策略主要负责 MANO 对 NFVI 管理策略的实现，主要涉及 MANO 与 NFVI 层的交互；负载均衡与状态管理策略主要负责 NFV 平台流量的负载均衡及状态管理，其研究工作通常与 NFV 层负载均衡与状态管理机制同步进行.

通过对以上各个领域的研究，高效完整的 NFV 系统得以搭建. 但是，由于存在于 VNF 层的软件安全问题、NFVI 层的系统安全问题、MANO 层的管理安全问题以及外包 NFV 平台的隐私及安全问题，所以 NFV 架构的安全问题仍然不容忽视.

3.1 VNF 构建及运行环境优化

传统的 VNF 开发通常采用自底向上的开发方式，针对不同 VNF 定制化实现不同的功能模块. 然而，不同的 VNF 中通常包含相同的网络数据包处理模块，例如，数据包头解析、协议解析、数据包分类等. 因此，学术界提出了采用模块化开发 VNF 的思想，通过组合不同现成的 VNF 功能模块，生成定制化的 VNF. 这种模块化的 VNF 开发方式能够极大简化 VNF 开发工作，促进 VNF 的开发和创新. 此外，为了支持隔离不同 VNF 的功能，目前的 VNF 利用虚拟化技术运行于不同的 VM 之上，但无论是虚拟机管理技术，例如，Xen 或 KVM，还是托管 VNF 的操作系统，例如，Linux，都未针对 NF 进行特殊的网络处理优化，使得 VNF 的处理性能低下. 目前对 VNF 运行环境的优化主要包括对 VNF 托管环境中网络驱动的优化以及 VNF 隔离机制的优化，相关工作主要有 ClickOS[9] 和 NetBricks[10].

Martins 等人[9] 通过结合基于 Xen 的操作系统 MiniOS 和模块化路由器 Click[11]，实现了高性能的模块化 VNF 构建和运行平台 ClickOS. MiniOS 通过保留 Click 运行所依赖的系统功能并精减其余 Linux 内核功能的方法，简化了系统内核，减小系统运行所需的计算资源和存储资源，使之处理网络数据包更加高效. 此外，ClickOS 使用高性能虚拟交换机 VALE[12] 替换了 OVS 虚拟交换机，并优化了 MiniOS 中的前端网络驱动以及 Xen 内核中的后端网络驱动，实现了 Xen 内核区域中数据包到 ClickOS 内存空间的直接映射. 单个 ClickOS 可以达到 41 μs 的虚拟机转发延迟，相比于 KVM 和 Xen 的网络驱动性能，减少了 62%的包处理延迟，吞吐率提升了 2～8 倍，达到了 546 Mbps[9]. 另外，ClickOS 通过命令行界面来控制 ClickOS 虚拟机的创建和销毁. 其中，虚拟机的相关信息都存储在 Xen 内核的数据库当中.

由于 VNF 开发者通常需要对不同的 VNF 功能模块进行优化，因此，即使使用模块化 VNF 构建平台，例如：ClickOS，也无法解决 VNF 开发过程中需要修改 VNF 功能模块来实现性能优化的问题.

此外，VNF 运行环境使用 VM 进行内存隔离的方式，由于 VNF 频繁穿越内存隔离边界，上下文切换产生的开销使 NFV 平台处理性能大幅下降. 针对以上问题，Panda 等人[12]提出了高性能 VNF 的构建框架 NetBricks. NetBricks 首先通过向开发者提供 VNF 模块的构建框架，使 VNF 开发者通过组合必要的 VNF 模块组件，实现高效的 VNF 功能模块，避免了开发者对现成 VNF 模块的重复修改. 然后，NetBricks 通过使用安全语言(Rust)和 LLVM[13]来实现软件内存隔离，并利用独特类型方法[14]，在编译阶段通过类型检查实现了数据包的隔离(数据包不能被其他不相关 VNF 处理). 相对基于容器的 VNF 隔离方式，这种隔离手段将 VNF 的吞吐量提高了 7 倍，相对基于 VM 的隔离手段，VNF 的吞吐量提高了 11 倍，达到了 1.6 Gbps[12].

从 VNF 构建及运行环境优化的研究工作进展中可以看出，VNF 构建方式的发展趋势是为开发者提供更细粒度，更加高效的 VNF 组件构建框架. 令 VNF 开发者从重复的模块修改开发模式转型为高效的模块组件组装开发模式，为开发者节省了大量时间，提供了极大地便利，促进了 VNF 创新和发展. VNF 的运行环境则从传统的硬件加速，逐渐转变为软件性能优化，通过软硬件结合的方式，不断提高 NFV 平台的处理性能.

3.2　NFV 管理系统设计及优化

NFV 技术旨在利用虚拟化技术，通过运行在标准服务器上的 VNF 来实现传统的专用硬件网络功能. 不同软件开发商在开发 VNF 的过程中，通常需要针对不同的 VNF 进行定制化实现其管理功能. 其中包括 VNF 的部署、网络流量过载检测；为了提升 NFV 系统资源利用率或解决网络流量过载问题而采用的负载均衡机制；以及 VNF 发生异常时所需的容错恢复机制等. 由于 VNF 的种类众多，不同的软件开发商对 VNF 管理功能以及管理接口的实现差异大. 网络管理员配置管理各种不同的 VNF 变得异常困难. 此外，软件开发商在实现 VNF 网络功能的过程中，需要针对不同的 VNF 重复实现管理功能，延长了软件开发周期，阻碍了 VNF 的软件创新. 因此，集成 VNF 管理功能的 NFV 框架成为网络管理员和 VNF 软件提供商的迫切需求. 利用集成 VNF 管理功能的 NFV 管理框架，网络管理员可以通过统一的管理接口实现对不同 VNF 的统一配置管理，VNF 开发商可以在 VNF 的开发过程中免于为不同 VNF 重复实现管理功能，如动态扩展功能、容错功能等，极大缩短了 VNF 开发周期，加快软件上市的速度.

在 NFV 管理系统实现方面，E2[15]是代表性工作之一. E2 的系统架构如图 5 所示，E2D 通过对 SoftNIC[16]进行扩展，实现了具有 0.3 μs 数据层虚拟交换机转发延迟、10 Gbps 吞吐量的高效数据转发层面[15]. 其中，SoftNIC 是一种可编程的高效软件交换机，通过对 SoftNIC 的扩展，可以实现流量监控、负载均衡、流量追踪、数据包分类以及不同 VNF 之间通信的隧道. E2 管理器和服务器代理组成了 E2 的管理层. E2 管理器负责根据网络状况和用户配置生成 VNF 管理决策，其中包括根据最小化服务器间流量传输量的原则(部署在同一台服务器上的 VNF 相互通信延迟低，开销小)、使用经过修改的经典 Kernighan-Lin 启发式算法[17]解决分图问题的方法、生成比随机部署机制高 2.25～2.59 倍吞吐量的最佳 VNF 部署策略[15]以及 VNF 动态扩展时保证流量状态一致性的状态迁移策略. E2 服务器代理主要负责执行 E2 管理器的决策并将服务器上 VNF 的相关信息向 E2 管理器上报. E2 的实现，为 NFV 平台提供了意义重大的通用 VNF 管理方案.

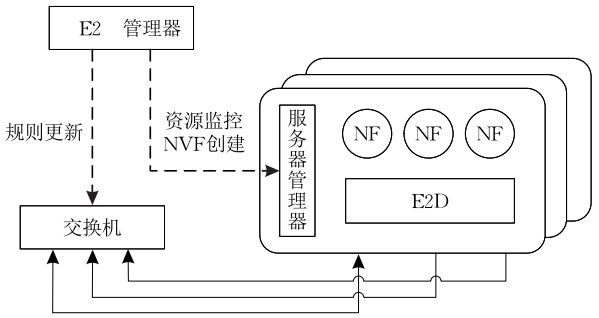

图 5　E2 系统架构[15]

在 NFV 管理系统的实现过程中，可以通过缩短数据包经过 NFV 平台所需的处理延迟、减少 NFV 平台资源部署数量等途径来提升 NFV 平台的数据处理效率和资源利用率，实现 NFV 平台的性能优化. 在 NFV 管理系统性能优化方面，目前主要有最小化资源部署量、共用不同 VNF 之间相同的数据包处理模块以及 VNF 并行处理数据包三种方式，主要工作包括 CoMB[18]、OpenBox[19]和 NFP[20].

Sekar 等人[18]实现的 CoMB，首先通过 NFV 平台中拥有网络和资源信息的控制器，将全部流量根据处理路径进行分类，并生成对应网络功能链(由指定次序的 VNF 依次连接而成)，称为超级应用. 然

后,控制器根据 VNF 平台中不同服务器的物理资源状况、不同超级应用对资源的使用情况以及每类流量的大小情况,生成流量到不同节点的分配策略.CoMB 通过最小化 NFV 平台内节点流量的峰值,减少了每个节点的最大资源部署数量,相对于传统的 VNF 部署策略节省了 1.8～2.5 倍的硬件部署资源,实现了 NFV 平台的性能优化[18].其次,CoMB 通过分析不同 VNF 之间可复用的网络功能模块,发现不同 VNF 中存在 26%～88%的可复用网络功能模块[18].共用这些可复用网络功能模块可用来缩短数据包经过 NFV 平台的处理延迟.

在利用 VNF 功能复用来缩短数据包处理延迟方面,CoMB 只分析了在会话重建和协议分析方面的复用潜力,在更细粒度的核心 VNF 处理模块(例如,包头的解析、包内容的修改、包的转发操作等)处依然存在复用优化的空间.Bremler-Barr 等人[19]提出的 OpenBox 通过将 NF 的控制层和数据层分离,利用逻辑中心化控制器来实现 NFV 平台的部署管理和优化.OpenBox 应用通过声明的方式定义 VNF 以及其在网络中的部署位置,OpenBox 控制器根据 OpenBox 应用中的 VNF 定义,通过将不同 VNF 重复的网络功能处理模块合并,实现不同 VNF 之间的相同网络功能处理模块复用.例如,如图 6 所示,数据包依次经过 Firewall 和 IDS 时都需要经过虚线所示的数据包读入和流量分类,并有相同的丢包和数据包输出操作,因此重复的数据包读入、流量分类以及相同的丢包和数据输出操作都可以进行合并.其中,重复网络功能处理模块的合并是由 OpenBox 控制器通过将不同的数据包处理模块进行分类(例如,修改数据包模块、分类模块、静态模块等),并根据不同类型处理模块的合并规则来实现.最后,实现网络数据包处理模块的 OpenBox 实例负责落实 OpenBox 控制器优化过后的 NFV 网络拓扑.经过 OpenBox 的 VNF 功能模块优化机制,NFV 平台网络吞吐量提升 86%～90%,处理延迟减少 35%～50%[19].

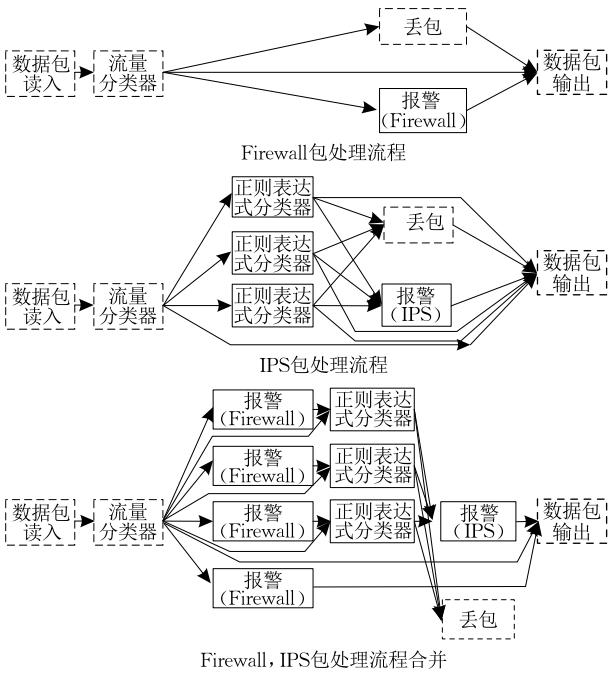

图 6 网络功能处理模块共用示意图[19]

在 NFV 平台中,网络数据包通常需要经过不同的 VNF 服务链,因此数据包经历的处理延迟通常为 VNF 服务链上所有 VNF 处理延迟之和.如图 7 所示,由于数据包在经过 IDS 和 Firewall 时内容都不会发生修改,因此可以将入侵检测系统和防火墙并行部署,使得两份相同的数据包并行通过入侵检测系统和防火墙,并保证处理结果和原始 VNF 服务链一致,减少数据包通过 NFV 平台的处理延迟.Sun 等人[20]通过分析不同网络功能之间的可并行性(例如,Firewall 和 Cache 服务器都不会对包进行修改,可以并行使用;而 NAT 和 Proxy 服务器会对包头进行修改,不能并行使用),实现了自动化并行不冲突 VNF 的 NFV 系统框架 NFP,从并行 NF 的角度优化了 NFV 系统的整体性能,减少了 45%的数据包处理延迟.

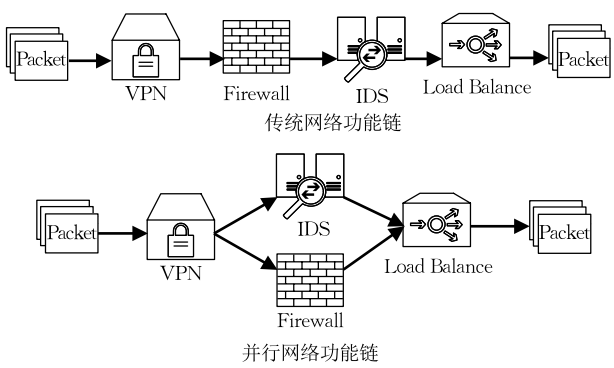

图 7 网络功能并行示意图

NFV 管理系统设计及优化的相关工作总结如表 1 所示,E2 和 OpenBox 实现了具有 NFV 系统通用管理功能的系统框架;CoMB 和 OpenBox 通过 VNF 数据包处理模块共用减小了 NFV 平台的数据包处理延迟;NPF 则通过实现 VNF 的并行处理来减小数据包的处理延迟.此外,OpenBox 还支持自定义 VNF 的实现,CoMB 的 VNF 部署考虑了最小化 NFV 平台资源利用率.

表 1　NFV 管理系统设计及优化工作总结

系统名称	管理功能	功能复用	最小资源部署	VNF 并行
E2[15]	√	×	×	×
CoMB[18]	×	√	√	×
OpenBox[19]	√	√	×	×
NFP[20]	×	×	×	√

3.3　策略实施与验证

在 NFV 平台中，网络管理员通过策略配置的方式来指定不同数据流量所对应的不同处理策略，使数据包按照策略依次经过按指定次序连接而成的 VNF 服务链. 区别于传统路由转发设备只依赖流表转发数据包的数据包处理方式，VNF 在处理网络流量时，具有对数据包的动态修改性以及对 VNF 状态的依赖性. 其中，VNF 对数据包的动态修改性是指 NAT 等 NF 会在处理数据包过程中修改数据包的内容. 这使得后续 VNF 无法依据修改前的数据包内容作出处理决策. 例如，如图 8 所示，连接在交换机 S1 上的 NAT，将主机 H1 和 H2 的内网 IP 地址 H1 和 H2 替换成公网 IP 地址 P1，使得后续的 IDS/IPS 无法统计不同内网主机的流连接数目. VNF 对状态的依赖性是指，由于 VNF 中存在多路复用网络连接、网络内容缓存等现象，使得 VNF 对网络流量的处理具有不确定性，VNF 通常根据内部不同的状态对网络流量进行不同的处理，同样使得后续的 VNF 无法正确地实施策略. 例如，如图 9 所示，在 H1 未访问 abc.com 时，H2 对 abc.com 的访问会受到连接在 S2 上资源管理设备(ACRL)的拦截. 但由于连接在 S1 上代理服务器的内容缓存和连接多路复用机制，代理服务器中的缓存状态在 H1 访问过 abc.com 后发生改变，使得 H2 可以在 H1 访问过 abc.com 后可以直接从代理处获取 abc.com 的内容，令 S2 处的 ACRL 无法对 H2 进行正常的内容访问控制. 因此，如何在 NFV 平台中实现正确

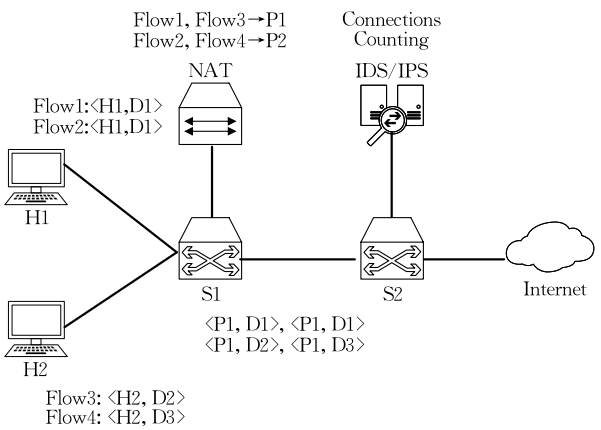

图 8　VNF 修改数据包示意图

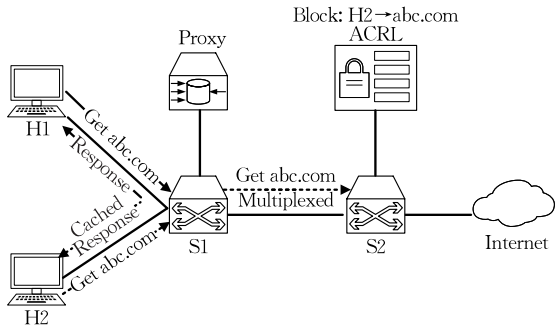

图 9　网络流量动态依赖示意图

的策略实施并进行验证成为了挑战. 目前在实现 NFV 平台策略正确实施方面，主要存在数据包携带 VNF 处理上下文信息和 VNF 服务链分解两种方法，相关工作包括 FlowTags[21] 和 Stratos[22].

Fayazbakhsh 等人[21] 采用 SDN 和 NFV 结合的网络架构，通过在数据包上携带策略执行需要的信息，实现了能正确实施策略的 NFV 系统 FlowTags，其策略实施机制对 NFV 平台性能影响甚小，仅降低 4% 的吞吐量，增加 0.5% 的运算资源[21]. 其中 VNF 通过在数据包打标签的方式，使得数据包上携带当前 VNF 处理数据包的上下文信息(数据包处理前后的 IP 地址、端口等信息)，SDN 交换机通过新增的南向接口根据标签向控制器请求相应的数据包操作，SDN 控制器根据应用通过北向接口规定的标签相关规则，对 SDN 交换机的请求作出响应. Gember 等人[22] 提出的 Stratos 系统通过克隆会改变数据包内容的 VNF，形成多条具有唯一执行路径的 VNF 服务链，结合 SDN 灵活的流量调度，解决了 NFV 平台中由于 VNF 改变数据包内容产生的策略实施困难问题. 相比而言，由于 Stratos 需要复制多个相同的 VNF 来实现策略的正确实施，因此需要消耗更多的资源，但相对于 FlowTags 需要修改 VNF 代码来携带数据包处理上下文信息，其优势在于降低了开发成本.

在 NFV 平台策略执行正确性验证方面 Panda 等人[23] 指出，相对于传统无状态网络(只包含 2/3 层转发设备的网络)的验证工作，在包含 VNF 的有状态网络中，网络验证存在一定的困难. 首先，VNF 的数据包转发行为不仅只根据转发表，还依赖于 VNF 中的配置信息以及状态信息，例如，Firewall 会根据策略配置和内部状态信息来判断是否对某条流进行转发. 因此在包含 VNF 的有状态网络中进行策略实施验证时，需要考虑如何对 VNF 进行合理建模来表示 VNF 的内部状态. 其次，VNF 对数

据包的处理并非只关注于三层和四层的网络栈信息,还会涉及到应用层数据,例如,深度包检测(Deep Packet Inspection,DPI)设备会查看数据包中的有效内容,Web 代理设备会根据数据包应用层数据来判断其转发路径.因此,对数据包的建模和验证同样需要重新思考和设计.目前 VNF 策略实施验证的手段主要包括静态验证和动态验证,相关工作包括 SLA-V[24] 和 BUZZ[25].

静态验证[23-24]通常先将 VNF 转发行为以及整个网络进行建模,然后形式化地表示数据包,最后通过符号模型检查器检查网络特定的不变量[23]违背现象来实现 VNF 策略实施的验证.其中,符号模型检查器检验的网络不变量通常包括网络可达性、网络隔离性、网络黑洞等.SLA-V 中使用的静态验证模型达到 92%的准确率[24].

由于静态验证只能检测出软件漏洞,即从策略到网络配置转化的正确性,但无法验证出真实网络环境中,由于硬件异常造成的转发错误.因此,通过在真实网络环境中运行测试流量的动态验证方法派上用场.Fayaz 等人[25]提出的 BUZZ 首先根据预期的策略和已经生成 NF 模型库对数据平面进行建模.其中,将 VNF 抽象为 FSM(有限状态机)的建模方式不仅能够表达状态信息,还可以表示与内容相关的策略(例如,恶意/正常).然后,BUZZ 根据数据平面模型,生成能够触发相关策略转发行为的抽象测试流量,进而将其翻译为具体的测试流量.其中,BUZZ 通过解耦不同层面的 NF 操作,解决了流量空间爆炸和状态空间爆炸的问题,达到了在 113s 内生成包含 600 个交换机的拓扑测试流量生成策略[25].最后,通过对比测试流量在实际数据平面中的转发行为与预期的 VNF 转发行为,便可实现 VNF 策略执行的验证.

VNF 策略实施与验证工作总结如表 2 所示,FlowTags 和 Stratos 解决了在含有 VNF 的有状态网络中策略正确实施的问题.其中 Stratos 需要消耗更多的物理资源,但避免了对 VNF 的修改.SLA-Verifier 和 BUZZ 分别使用静态验证和动态验证的手段,实现了 VNF 策略实施的验证.

表 2 NFV 策略实施与验证工作总结

系统名称	策略实施	不修改 VNF	静态验证	动态验证
FlowTags[21]	√	×	×	×
Stratos[22]	√	√	×	×
SLA-V[24]	×	√	√	×
BUZZ[25]	×	√	×	√

3.4 资源分配和迁移策略

NFV 技术建立于虚拟化技术之上,VNF 的部署和迁移都涉及资源开销,例如,网络带宽资源、服务器硬件资源、交换机 TCAM 容量等.因此,NFV 平台需要通过高效的算法来决策资源的分配以及 VNF 的动态迁移策略,使之满足期望的目标,例如,最小化 VNF 部署开销和操作开销、节约能源等.在 NFV 平台中,MANO 负责资源分配和迁移策略的计算,其根据虚拟化设施管理器以及 VNF 管理器中虚拟资源和 VNF 的相关信息,利用优化模型进行问题建模,从 VNF 服务链的构造、VNF 部署以及 VNF 调度三个角度来计算 NFV 平台的资源分配及迁移决策.其中,VNF 服务链构造是指按照网络策略的定义,根据不同 VNF 之间的依赖,依次连接 VNF,不同的 VNF 连接次序会造成 VNF 对网络带宽和节点计算能力的不同需求.VNF 部署是指将 VNF 服务链部署在实际的物理资源之上,由于不同的网络节点拥有不同的网络资源数量,不同的 VNF 对资源的需求量不同,因此,不同的 VNF 部署策略会产生不同的网络部署开销.VNF 调度是指,在保证 VNF 服务链策略依赖的前提下,在有限的物理资源上对不同 VNF 服务链上的 VNF 进行运行调度,使得不同的 VNF 服务链在有限的资源上得以有序高效有序地运行.与在有限的 CPU 资源上调度不同的进程类似,不同的调度策略将使得不同 VNF 服务链上的 VNF 在不同的时间单元运行,进而影响不同 VNF 服务链的运行效率.例如,按照优先级进行调度的策略则会优先调度优先级高的 VNF,按照吞吐量进行调度的策略则会优先调度吞吐量大的 VNF.如何在不同的 NFV 场景中,根据不同的目标进行有效的资源分配和迁移成为 NFV 研究的热点之一.针对不同的 NFV 部署场景,MANO 使用的优化模型会根据实际情况选择不同的优化模型输入参数、优化目标以及优化策略来计算资源分配和迁移决策.

优化模型的输入参数通常由不同类型的资源组成,其中包括与物理节点相关的内存资源、存储资源、计算资源和与链路相关的带宽资源、数据包传输延迟.例如,Mehraghdam 等人[26]在 VNF 服务链的部署问题中考虑了传输延迟、计算资源以及链路数据容量等资源因素;Basta 等人[27]在优化 VNF 部署位置来实现最小化网络负载开销时考虑了数据层转发延迟、数据中心网络功能部署数量以及 VNF 控制开销等因素.

在不同的 NFV 部署场景中，资源分配和迁移策略使用的优化模型有不同的优化目标. 例如，Luizelli 等人[28]以最小化 NFVI 上的 VNF 部署数量作为其优化函数的目标；Riggio 等人[29]则以最大化 NFVI 上能够部署的 NFV 服务链数量为目标. 这些目标通常与服务质量、容错、负载均衡、节能、利润收益紧密相关. 例如，提供 IP 语音功能的网络服务需要中等带宽和低延迟；电信服务提供商为了实现利益最大化，需要最小化 NFVI 的部署开销，其中包括物理资源和能源消耗；以及 NFV 平台中的容错备份节点需要最大化覆盖其服务的节点和链路.

在 NFV 资源分配和迁移的优化策略方面. 由于 NFV 资源分配和迁移策略问题可以视为虚拟网络嵌入(Virtual Network Embeding，VNE)的一般化问题，根据 VNE 问题是 NP-hard 问题可知[30]，大部分 NFV 资源分配和迁移也是 NP-hard 问题. 因此，在大中型的 NFV 平台中，由于网络节点的数量巨大，通过运行时求解优化问题最优解的解决方式是不可行的. 目前针对资源分配和 VNF 迁移的优化问题解决策略包括使用最优解算法、启发式算法和元启发式算法. 其中，最优解算法通常使用线性规划算法求解优化问题的最优解. 尽管整数线性规划问题在很多实际情况下是 NPC 问题，但通过分支定界、分支定价等算法，可以在合理的时间内，在小型网络中得出最优解. 例如，Moens 等人[31]在传统 NF 和 VNF 混合部署的网络环境中，使用求解整数线性规划问题最优解的方法来求解最小化物理节点使用数目的 VNF 部署方案. 但由于求解整数线性规划问题最优解的复杂性高，该方案只适用于小型网络. 为了在任意规模的网络中最小化资源分配和 VNF 迁移的决策计算时间，各种基于启发式的算法得以应用. 例如，Riggio 等人[32]在无线局域网(WLAN)NFV 场景中，通过使用递归贪心和最短路径算法，实现了满足约束条件的 VNF 部署；Ma 等人[33]针对会改变流量大小的 VNF，使用启发式算法，解决了在服务器硬件资源和网络链路带宽容量的限制下，如何通过选择 VNF 部署位置，来优化网络链路开销的问题. 另一方面，由于 NFV 资源分配和迁移策略的问题可以被视为在离散搜索空间内寻求最优解的最优化问题，迭代改进问题的解决方案可以接近最优解. Mijumbi 等人[34]使用基于元启发式算法的禁忌搜索算法来解决 VNF 的动态部署和调度问题.

另外，通过使用合理的 NFV 资源分配和迁移策略，可以对一些网络攻击进行防御. Fayaz 等人[35]结合 SDN 技术和 NFV 技术实现了灵活的、可扩展的 DDoS 防御系统 Bohatei. Bohatei 防御系统根据 DDoS 攻击流量大小和类型来判断防御虚拟机的部署数量和位置，利用 NFV 技术灵活地在相应位置部署相应数量和类型的防御虚拟机，实现了实时有效地防御 DDoS 攻击. SDN 技术用于在最小化用户可感知延迟和网络阻塞的前提下，将可疑流量调度到部署有防御机制的虚拟机中.

表 3 为 NFV 资源分配及迁移策略部分相关工作的总结，从表 3 中我们可以看出，尽管在不同的 NFV 部署场景中优化目标和限制条件有所不同，但优化模型的建立都从 VNF 服务链构造、VNF 部署以及 VNF 调度三个阶段入手，并采取最优解算法、启发式或元启发式算法进行求解.

表 3　NFV 资源分配及迁移策略部分相关工作

文章引用	NFV 部署场景	优化角度	优化策略
Mehraghdam 等人[26]	TSP 网络	VNF 服务链构造、VNF 部署	最优解、启发式
Basta 等人[27]	移动网络	VNF 部署	启发式
Luizelli 等人[28]	一般 NFV 网络	VNF 部署	最优解、启发式
Riggio 等人[29]	无线网络	VNF 部署	最优解
Moens 等人[31]	混合网络	VNF 部署	最优解
Riggio 等人[32]	无线网络	VNF 部署	最优解
Ma 等人[33]	一般 NFV 网络	VNF 部署	启发式
Mijumbi 等人[34]	TSP 网络	VNF 部署、VNF 调度	启发式、元启发式
Fayaz 等人[35]	一般 NFV 网络	VNF 部署	启发式

3.5　NFV 负载均衡及状态管理技术

NFV 技术建立在虚拟化技术之上，因此，NFV 可以根据需求通过灵活动态扩展/合并 VNF 实例的方式来进行网络流量的负载均衡，实现对 NFV 平台资源的高效利用，提高网络服务的效率，保证网络服务的正确执行. 由于 VNF 在处理网络流量的过程中通常依赖于流的状态信息，例如，NAT 映射表、流的数量统计、流计时信息等，在 VNF 扩展/合并过程中，为了保证 VNF 处理结果的一致性，需要将流的状态进行管理和同步. 在结果一致性和低

延迟的条件下正确实现 NFV 的负载均衡成为挑战. 目前,学术界对 NFV 负载均衡策略、NFV 负载均衡系统实现以及 VNF 状态管理方面进行大量研究. 相关工作包括 Stratos[22]、FreeFlow[36]、OpenNF[37]、StateAlyzr[38]以及 VFW[39]. 此外,物理机器故障、电路故障等情况引起的网络功能异常会严重影响 NFV 平台的稳定性[40],因此,VNF 如何快速、高效地从失效状态恢复也成为 NFV 技术需要解决的问题之一,相关工作包括 FTMB[41].

在 NFV 平台负载均衡策略方面, Gember 等人[22],采用 SDN/NFV 网络架构,根据整个 NFV 平台的网络资源状况,通过流量分配,VNF 迁移以及 VNF 扩展的多阶段负载均衡策略,实现具有高效,可扩展性负载均衡机制的 NFV 系统框架 Stratos. 其中流量分配阶段用于高效解决网络中短暂的流量突发状况,VNF 迁移和扩展两阶段用于缓解 NFV 平台中持续的计算瓶颈和网络瓶颈. Stratos 采用的多阶段负载均衡策略,逐步定位和解决网络瓶颈,减少了不必要的 VNF 动态扩展,将整个 VNF 平台的系统计算资源利用率提高了 5%～17%[22].

由于 VNF 通常需要根据相关状态来处理网络流量,因此,在通过 VNF 扩展来实现负载均衡时,主要挑战在于如何保证 VNF 扩展前后状态的一致性. 在实现 VNF 正确动态扩展方面. 如图 10 所示, Rajagopalan 等人[36]实现的 FreeFlow 系统分类了 VNF 处理网络流量过程中的相关状态,通过结合 FreeFlow 库、编排器、虚拟机管理器以及 OpenFlow 虚拟交换机,实现了根据网络流量状况自动对 VNF 进行扩展和合并的负载均衡系统框架 FreeFlow,其

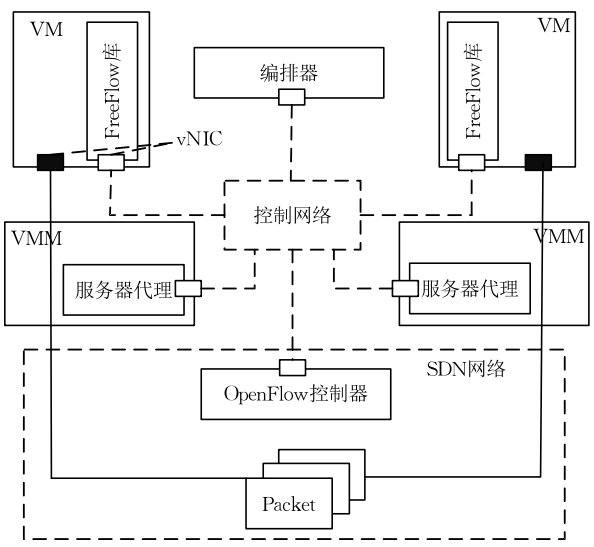

图 10 FreeFlow 系统架构图[36]

通过负载均衡策略以及灵活的扩展机制,将 NFV 平台中 VNF 处理流量的标准偏差降低了 73%,运算资源扩展和回收的速度提升了 50%,极大提高了资源利用率[36]. 其中,VNF 开发商通过框架提供的 FreeFlow 库来对 VNF 中的状态进行分类管理,并对状态迁移过程中的事务需求进行明确定义. 编排器负责通过 FreeFlow 提供的接口监控网络流量,并根据网络状况作出 VNF 扩展/合并的决策,虚拟机管理器负责 VNF 生命周期的管理,OpenFlow 交换机负责网络流量的动态调度.

FreeFlow 虽然实现了 NFV 平台中的负载均衡策略,但其在不同 VNF 状态迁移过程中会发生数据包丢失和乱序,导致部分恶意流量漏检及 VNF 误报,降低 NFV 系统性能. 如何解决在多个 VNF 实例上协作处理网络流量过程中产生的 VNF 内部状态和网络流量转发状态的一致性问题成为了 NFV 状态管理问题研究的目标之一. Gember-Jacobson 等人[37]实现了能协调控制 NFV 平台中 VNF 内部状态和网络转发状态的系统 OpenNF. OpenNF 通过使用事件和两阶段状态更新机制,保证在 VNF 动态扩展,容错恢复的状态迁移过程中,数据包不发生丢失和乱序. 以此来实现不同 VNF 之间状态迁移过程中高效、准确、细粒度的流量分发,其能在 215 ms 内实现 500 条流状态无丢失的转移与复制[37]. OpenNF 使用类 SDN 的三层结构,其中,北向接口用于控制应用进行 VNF 的状态管理操作(例如,状态迁移、状态复制等),南向接口定义了 OpenNF 控制器从 VNF 获取或向 VNF 注入 VNF 内部状态的 VNF 标准接口.

由于网络功能的种类众多,且在 VNF 中涉及状态的代码通常高达 1 万～10 万行,使得通过人工修改 VNF 代码来兼容各种网络系统框架(例如,OpenNF)变得异常困难. Khalid 等人[38]提出的自动化 VNF 状态识别系统通过设计相关算法,结合前向程序分片[42]、指针分析[43]和程序切分[44]等静态程序分析技术,经过流相关状态识别、可更新状态识别和状态流空间识别三个阶段,实现了精准、高效的自动化状态识别. 其可在弹性扩展、容错恢复等过程中. 自动化识别 VNF 需要迁移的相关状态. 其能够将人为代码修改周期缩减 20 倍,将代码修改量减少 600～8000 倍[38]. 极大地降低了 VNF 开发者兼容 NFV 框架的难易程度,缩短了 VNF 开发周期,促进了 VNF 的创新.

除了适用于不同 VNF 的一般性负载均衡系统

框架的设计和实现，不同的 VNF 在使用这些负载均衡框架进行动态扩展的过程中，也存在不同的挑战，例如对 VNF 中不同规则之间的依赖处理等. Deng 等人[39]在 NFV 和 SDN 技术的基础上，通过分析防火墙规则之间的依赖关系、SDN 交换机上流表项之间的依赖关系以及防火墙规则和 SDN 流表项之间的依赖关系，实现了确保虚拟防火墙动态扩展过程中保持语意一致性，流表更新正确性，并具有缓冲区溢出避免机制和最优资源利用率的高效可扩展虚拟防火墙控制器 VFW，其可以在 0.75 s 时间内完成对拥有 1200 条规则 Firewall 中 1200 条数据流以及依赖状态的迁移[39].

在目前高强度的动态网络环境下，VNF 提供的网络服务必须具有高可用性. 因此，VNF 能否从意外情况导致的失效中精准、高效地恢复决定了 NFV 平台的稳定性和可用性. 在 VNF 错误恢复系统设计实现方面，Sherry 等人[44]基于经典的回滚错误恢复方法，采用系统快照和数据包缓存的系统信息备份方式，通过设计使用在短时间内备份信息的“顺序日志”和多核环境下重现正常系统状态的“平行释放”算法，实现了精准，高恢复速度的通用 VNF 错误恢复机制 FTMB，其在能产生 5%～30% VNF 吞吐量开销的前提下，在 40 ms～275 ms 内实现 VNF 系统丢失状态的恢复[41].

NFV 负载均衡和状态管理技术相关工作总结如表 4 所示，Stratos 采用多阶段负载均衡的策略，大幅度提升了 NFV 负载均衡系统的扩展性. FreeFlow 首次实现了 VNF 在负载均衡阶段的正确动态扩展. OpenNF 弥补了 FreeFlow 动态扩展 VNF 时出现的数据包丢包乱序的问题. StaterAlyzr 通过程序分析技术，实现了自动化分析负载均衡和容错系统中需要迁移的相关状态，方便了 VNF 对各种 NFV 系统框架的兼容. VFW 则解决了防火墙高效动态扩展中存在的规则依赖问题. FTMB 实现了 VNF 高效、精准的错误恢复机制.

表 4　NFV 负载均衡和状态管理技术工作总结

系统名称	状态一致性	数据包丢包乱序	多阶段负载均衡	自动化状态分析	VNF 规则依赖	错误恢复
Stratos[22]	×	×	√	×	×	×
FreeFlow[36]	√	×	×	×	×	×
OpenNF[37]	√	√	×	×	×	×
StateAlyzr[38]	×	×	×	√	×	×
VFW[39]	√	×	×	×	√	√
FTMB[41]	×	×	×	×	×	√

3.6　NFV 架构中的安全问题

随着 NFV 技术的日渐成熟，NFV 技术逐渐被工业界采用. 但 NFV 技术中存在的安全问题依然不容忽视. 从 NFV 技术的实现方式着手，NFV 技术中的安全问题可以分为传统 NFV 架构中的安全问题和 NFV 外包场景下的安全问题.

NFV 架构中的安全问题主要包括 NFVI 中的安全问题和 VNF 中的安全问题. 其中，NFVI 中的安全问题分为 NFVI 外部管理安全问题和NFVI自身安全问题. NFVI 外部管理威胁主要来自 NFV 平台设施管理人员的不恰当操作，此类安全问题通常可以通过制定严格的设施操作流程来避免. NFVI 的自身威胁通常来自 NFVI 软硬件的设计和实现缺陷. 因此，NFVI 的设备应该通过安全认证过程来减少自身威胁. VNF 中的安全威胁主要来自于 VNF 管理和实现的缺陷. 因此，VNF 应该采用标准安全机制来对 VNF 的管理和运行进行认证、授权、加密和验证，以增强 VNF 的安全性，例如，使用标准认证授权机制执行虚拟机的访问控制、使用标准加密算法加密敏感数据以及使用 TLS、IPSec 等安全网络协议增强网络通信的安全性.

Sherry 等人[1]对现代企业网中网络功能设备的部署情况进行了调研，发现企业网络中，网络功能的数量与 L2/L3 层网络转发设备（路由器和交换机）相近，进而提出了企业网络功能外包的概念. 其基于 DNS 的方式将企业网络流量以最小延迟重定向到云平台，实现了在云平台上配置网络功能拓扑，并拥有简单扩展功能的 NFV 平台 APLOMEB，该方案利用云计算平台的规模经济降低企业网内网络功能部署的开销. 由于将 NFV 平台外包到第三方云服务平台后，VNF 的运行完全依赖于不可信第三方提供的物理资源和人员管理. 因此，在第三方云平台上部署 VNF 存在以下问题：(1) 云服务提供商可能为了节约成本而将用户期望的 VNF 替换为廉价的 VNF 以及对 VNF 正常运行的恶意干涉；(2) 云平台内的 VNF 部署可能存在策略配置错误的情况；(3) 流经云平台的企业网流量隐私无法得到保障. 所以，在 NFV 外包场景下，如何验证网络功能

和网络功能服务链(Service Function Chain,SFC)的执行是否与用户预期一致以及外包流量的隐私保护成为了挑战.目前学术界主要通过设计基于密码学和统计理论的验证协议来实现对 NF 和 SFC 的验证,相关工作包括 Bringing Execution Assurance[45]和 VSFC[46];对流量隐私的保护则主要通过加密搜索和可信硬件来实现,相关工作主要包括 Blind-Box[47]、Embark[48]以及 SGX-Assisted VNF[49-52].

在 NFV 平台外包到第三方云服务平台之后,云平台高级别系统软件(例如,虚拟机管理器、虚拟机以及操作系统)拥有对 VNF 运行环境的完整控制能力,能够恶意改变 VNF 运行的控制流和数据流,来更改 VNF 的行为.针对第三方云服务平台恶意篡改 VNF 控制流、数据流以及相关配置来破坏 VNF 执行结果正确性的问题.Yuan 等人[45]针对基于模式匹配的网络功能(例如,Firewall、IDS),通过设计不受字典攻击的响铃机制[53],实现了在低开销前提下,具有高可信度的基于模式匹配的 VNF 正确执行验证,其能在 200ms～500ms 延迟,360Mbps 吞吐量的前提下,验证 1000 个数据包被基于模式匹配的 VNF 正确处理[45].Zhang 等人[46]通过使用在 VNF 之间逐跳认证数据包来源的方式,实现了外包场景下,仅增加每个包 48μs 的处理延迟,降低 10% 吞吐量的前提下,对 SFC 链的执行异常进行检测[46],例如,跳过防火墙来逃避检测等行为.

针对 NFV 外包过程中的流量隐私保护问题,Sherry 等人[47]通过设计轻量级加密搜索机制,结合混淆电路和遗忘传输机制,实现了比现存可搜索加密机制快 3～6 个数量级的加密流量深度包检测(Deep Packet Inspection,DPI)系统 BlindBox.其支持的 DPI 功能包括指定位置的恶意关键词匹配;由于 BlindBox 只能实现对加密流量的关键字匹配操作,其支持的 VNF 种类过于局限.Lan 等人[48]在 BlindBox 系统的基础上,通过设计能在包含 1 万条规则的虚拟 IDS 中进行拥有 1.5Mpps 吞吐量的支持前缀匹配的加密搜索机制[48],实现了支持在加密流量上进行关键词匹配和前缀匹配操作的 VNF 系统 Embark.其支持的 VNF 种类包括 DPI、HTTP 代理、IP 防火墙、NAT 以及三/四层负载均衡,极大增加了 BlindBox 所支持的 VNF 种类.

通过使用密码学手段和统计学理论来解决 NFV 平台外包场景下的流量隐私问题和 VNF 执行正确性验证问题通常局限性大,实用性窄,设计难度高.随着可信计算的发展,可信硬件的兴起为 NFV 平台外包场景下的安全问题提供了新的解决方案.Intel 的 Software Guard eXtension[54](SGX)技术作为目前最新的可信计算技术.能够在高级别操作系统不可信的情况下,为程序提供隔离的运行内存环境[55]以及远程验证的功能[56].以保证程序在不可信第三方云服务平台上部署的正确性,以及运行的完整性和隐私性.Shih 等人[49]首次提出通过使用 SGX 技术来保护 VNF 中流量的敏感状态来增强 VNF 的安全性.Coughlin 等人[50]基于模块化路由器构建框架,首次用 SGX 技术实现吞吐量为 14.77 Gbit/s 的安全 VNF.Han 等人[51]在使用 SGX 技术保护 VNF 的同时,封装了底层网络操作,例如,TLS 握手信息解析、数据包重组和加解密等操作,为开发人员在构建 SGX 支持的 VNF 时提供了方便的底层网络操作.Duan 等人[52]针对 SGX 技术提供的安全区域(Enclave Page Cache,EPC)大小有限、系统调用开销大等局限性.通过高效的 VNF 划分,使用层次化状态存储结构,批处理数据包传送,实现了使用 SGX 技术的高效安全 VNF,其在处理 1.5Mb 个并发数据流时,仅增加每个数据包 3μs 的处理延迟[52].

NFV 安全问题相关工作总结如表 5 所示.Bringing Execution Assurances 通过密码学和统计理论的手段,解决了 VNF 异常检测问题;VSFC 通过加密认证的方式,解决了 VNF 链执行异常的检测问题;Blindbox 和 Embark 使用密码学和加密搜索的相关理论基础,通过密文操作来解决流量隐私保护问题;由 SGX 辅助的 VNF 则利用可信硬件来实现 VNF 执行的完整性和保密性.目前,通过密码学实现的加密搜索和使用可信硬件的方式都能用于解决 NFV 技术在外包场景下的流量隐私和功能执行正确性验证问题,但其各有千秋.加密搜索的方式无须额外的硬件支持,但其实现复杂,支持 NF 操作少,效率低下,消耗资源量大.而可信硬件的方式实现简便,效率高,支持任意的 NF 操作,但其需要特殊的硬件支持,因而有额外的硬件开销.

表 5 NFV 安全问题工作总结

系统名称	流量隐私保护	VNF 执行异常执行检测	VNF 服务链执行异常检测	VNF 执行保护
Bringing Execution Assurance[45]	×	√	×	×
VSFC[46]	×	×	√	×
BlindBox[47]	√	×	×	×
Embark[48]	√	×	×	×
SGX-Assisted VNF[49-52]	√	×	×	√

4　NFV 部署场景

NFV 因其节省开销、操作管理简单以及部署灵活等优点，被广泛应用于各种网络场景中. 无论在数据层处理方面还是控制层管理方面，NFV 都有相关应用. 本节通过 4 个 NFV 工业应用实例来说明传统网络架构的局限性以及基于 NFV 的网络架构相对与传统网络架构的优势.

4.1　云数据中心网络功能虚拟化

云计算[57]是一种能够随时随地，通过网络访问可配置共享资源池中资源（例如，网络、服务器、存储、应用和服务等）的资源使用模式. 其利用多样的网络接入方式（移动设备、主机、服务器）、资源池共享、灵活扩展以及按需付费等关键特性，可以通过最少的管理操作和服务提供商交互实现资源的快速供应和及时释放. 云计算通过设施即服务（Infrastructure as a Service，IaaS）、平台即服务（Platform as a Service，PaaS）、软件即服务（Software as a Service，SaaS）三种服务模式，为用户提供可配置的设施、平台以及软件支持而无需用户关心云数据中心设施的具体实现和管理.

传统的硬件网络功能由于其固定性，无法在支持多租户的云数据中心为用户提供相互隔离的、可灵活扩展的网络功能服务. 随着 NFV 技术的兴起，这一问题得以解决，云数据中心可以通过部署 VNF 来为云数据中心和用户提供相互隔离、高效灵活的网络功能服务[58].

云数据中心通过使用 NFV 技术，不仅可以为自己的数据中心网络提供灵活可靠的网络功能来处理所有经过云数据中心的网络流量，还可以通过在不同租户的虚拟机上运行 VNF 为租户提供独立的可配置网络服务，达到网络功能的多租户支持.

4.2　蜂窝基站虚拟化

传统蜂窝网络的无线接入网络（Radio Access Network，RAN）通常由独立的基站组成. 这些基站用于为移动手机处理、传输无线信号，并通过回程链路将移动手机的数据转发至核心网络. 然而，传统 RAN 网络架构拥有一些局限性. 首先，蜂窝网络运营商为了使基站能够处理网络负载峰值，只能部署拥有处理最大网络负载能力的基站. 但由于基站使用模式的不同和用户的流动性，基站需要处理的流量在不同的时间段波动幅度很大. 因此，基站的处理能力通常得不到完全使用，大部分时间出现处理能力过剩的现象. 而由于基站地理位置的分散性，当某些基站出现流量高峰时，处理能力过剩的基站也无法分享空闲的处理能力；其次，鉴于在有限的频谱范围内，基站需要复用无线电频率. 因此，基站部署的规划和优化变的复杂且困难；第三，每个基站都需要独立的回程链路设备、环境监控系统、冷却系统和备用电能设备. 整套基站系统设施的部署需要占用巨大的空间.

经过一定时间的发展，RAN 网络架构从一体化基站发展为分布式基站. 分布式基站将无线电功能单元（Remote Radio Head，RRH）和数字功能单元（Building Baseband Unit，BBU）分离，BBU 主要实现天线阵列系统功能以及 MAC 层和物理层功能，而 RRH 则负责无线电信号的转换和增强. 其中，基带无线电信号通过光纤在 RRH 和 BBU 之间传播. 分布式基站使得 RRH 和 BBU 的分散部署成为可能.

如图 11 所示，运营商利用分布式基站的网络结构，以云 RAN（Cloud RAN）[59, 60]的方式，通过在数据中心对 BBU 池资源的虚拟化，实现了灵活的虚拟化蜂窝网络基站. 云 RAN 通过使用 NFV 架构（对 BBU 的集中和虚拟化），减少了硬件基站的部署，显著降低了运营商的运营，能源和房屋租赁成本. 此外，利用 NFV 技术灵活的动态扩展、资源共享等优势，云 RAN 节省了大量的计算资源部署开销. 据 Bhaumik 等人[61]统计，云 RAN 可以通过基站流量的动态共享处理，减少 22％的计算资源部署.

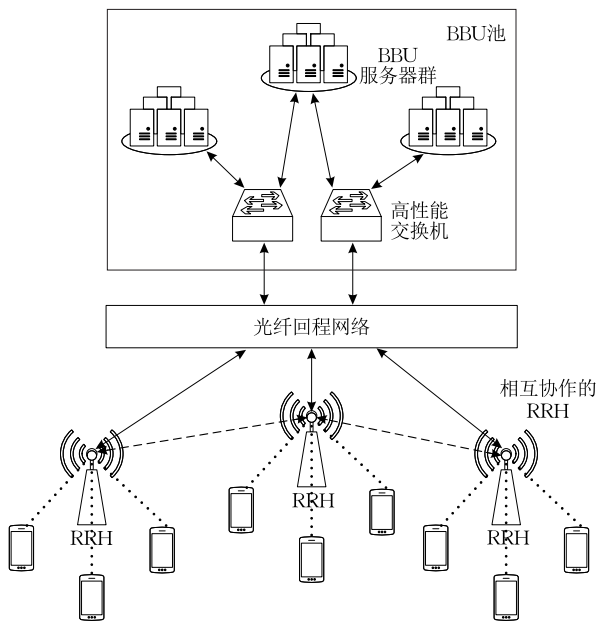

图 11　云 RAN 网络架构图

4.3 移动核心网虚拟化

传统移动核心网络由于大量部署昂贵的专有设备和使用硬态信令协议,使得网络管理和更新变得复杂困难.当网络功能设备的一些功能无法使用时,蜂窝网络运营商必须更换现有的硬件网络功能设备.这种方式使得网络功能设备的频繁升级和网络服务的及时更新扩展变得异常困难.由于被替换网络功能设备的大部分功能尚可正常使用,因此造成了大量不必要的网络设备开销.目前移动核心网络主要使用隧道机制和中心网关(例如,4G 核心网(Evolved Packet Core,EPC)中的包数据网络网关(PDN GateWay,PGW))交换用户数据流量,蜂窝网络运营商对这些长距离永久隧道的维护异常昂贵和困难.

云 EPC(Cloud EPC)[62-63]通过应用 NFV 技术来解决传统移动核心网络中存在的以上问题.如图 12 所示,云 EPC 实现的 VNF 目标包括移动性管理实体(Mobility Management Entity,MME)、规约签署用户服务器(Home Subscriber Server,HSS)、PGW 以及策略和计费规则功能(Policy and Charging Rules Function,PCRF).此外,蜂窝网络运营商可以通过虚拟化 IP 多媒体子系统(IP Multimedia Subsystem,IMS)中的部分组件,例如,呼叫会话控制功能(Call Session Control Function,CSCF)等,来实现更好的 LTE 语音功能(VoLTE)支持.

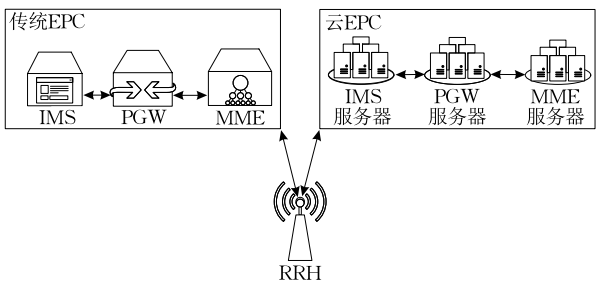

图 12 传统 EPC 与云 EPC 共存网络架构图

云 EPC 通过实现上述的 VNF,消除了分布式网络资源的地理限制,构建了一个更加智能、灵活且可扩展的网络架构.其通过利用 NFV 动态扩展、资源共享等优势,在降低设备部署开销的同时,能在遭遇资源、性能瓶颈时灵活的共享,分配硬件资源.并确保本地资源发生故障时,依然可以提供可靠稳定的网络服务.通过 NFV 技术,云 EPC 可以实现 PGW 的灵活部署,例如,将 PGW 和基站(eNodeb)一起部署,可以避免使用长距离的隧道.

4.4 家庭网络虚拟化

网络服务提供商通常通过专用客户终端设备(Customer Premise Equipment,CPE)向用户提供家庭服务.传统的 CPE 包括用于连接互联网的住宅网关(Residential Gateway,RG)和多媒体服务的机顶盒(Set-Top-Box,STBs).由于网络协议电视(IPTV)的交互式网络流控制功能(例如,快进和倒带),使 IPTV 服务的传输变得复杂.

通过在数据中心部署虚拟化机顶盒以及住宅网关的相关虚拟化组件(例如,Firewall、动态主机配置服务器(Dynamic Host Configuration Protocol,DHCP)、虚拟专用网络网关(VPN 网关)以及 NAT 等),网络服务运营商只需要提供廉价和低维护成本的网络连接设备来保证用户到互联网的物理连接性.

这种虚拟化的网络架构为网络运营商和用户带来了很多好处.首先,运营商通过避免 CPE 设备的频繁维护和更新,减轻了运营商产品维护的负担;其次,NFV 架构通过资源共享,不仅可以提供极大的存储能力,还能实现在不同的地点共享多台设备的内容,帮助服务提供商提高用户体验质量;再者,NFV 架构可以通过动态服务质量管理和应用流量之间的受控共享,实现内容供应商对用户的可控内容供应能力;最后,NFV 架构可以最小化不同服务对 CPE 的依赖性,实现新服务的平稳引入.

5 总结及展望

5.1 NFV 技术的优势

NFV 利用虚拟化技术降低了企业网络建设的成本,减小了网络功能的失效率,提供了更灵活的 NF 扩展方式.与传统的硬件网络功能设备相比较,NFV 技术拥有以下技术优势:

5.1.1 降低网络功能部署开销

传统网络功能设备由开发商配套的软件和专有硬件组成,因此,部署不同的网络功能需要购买不同的高价网络功能硬件设备,造成大量部署开销.相对而言,NFV 技术通过利用虚拟化技术,实现了 VNF 在商用服务器上灵活的廉价部署,大幅减少了企业网 NF 部署开销.

5.1.2 提供灵活的弹性扩展功能

企业网络为了保证能在网络流量峰值时正常提供网络服务,需要在网络中部署足够数量的不同

NF 设备来满足网络流量峰值下的网络服务需求. 但在大部分网络流量正常的情况下，NF 设备存在资源过剩的情况，浪费了资源和资金. NFV 技术通过虚拟化技术，可根据需求动态扩展和合并 VNF 实例，通过灵活的资源共享，提高资源利用率，提升网络处理性能.

5.1.3　简化网络功能更新

由于传统网络功能设备的软硬件绑定. 当企业需要对 NF 进行部分功能的更新时，通常需要替换整个 NF 硬件设备，造成了大量不必要的额外开销. 此外，NF 设备软硬件绑定的产品开发模式使得 NF 设备制造商开发周期长，成本高. 相对而言，VNF 在更新时仅需要修改相关代码，大幅度缩减了 NF 开发商的更新周期，促进了 NF 的创新，并为企业网节省了 NF 更新时不必要的额外硬件开销.

5.1.4　降低网络功能管理复杂度

企业在使用传统网络功能设备时需要庞大的管理团队来管理不同的硬件设备，相对而言，通过利用 NFV 技术，在云平台上搭建虚拟网络的方式，将硬件计算资源设备的管理交给云平台，大幅度减小了企业网络硬件设备管理的难度，同时避免了由于管理失误而产生的网络服务失效.

5.2　NFV 技术的不足

尽管 NFV 技术拥有很多优势，但它相对于传统的网络功能设备，也同样存在以下一些不足：

5.2.1　网络功能处理性能下降

传统网络功能设备通过结合专有硬件和软件方式，实现了网络流量处理过程中的相关优化，提升了 NF 处理网络流量的性能. NFV 技术旨在将软件 VNF 部署在一般性商用硬件上实现传统 NF 的功能，因此，相对于传统网络功能缺少特殊的硬件优化，VNF 处理网络流量的性能有所下降. 另外，传统的硬件网络功能在设计生产时通常会综合考虑 NF 硬件的扩展性，使之在庞大的硬件集群环境下依然保持高效的处理能力. 然而，部署在通用服务器上的 NFV 平台是否能在扩展的 VNF 集群中依然保持高效的处理能力以及低廉的部署开销尚待考察.

5.2.2　网络流量数据隐私暴露

NFV 技术在外包场景下依赖于不可信第三方云计算平台提供的计算资源. 由于用户流量在不可信计算平台上进行处理，流量数据完全暴露在云计算平台中，使得用户数据隐私无法得到保障.

5.2.3　软件攻击面扩大

由于 NFV 利用虚拟化的手段通过软件来实现传统硬件设备中的网络功能，因此，VNF 中存在软件漏洞的可能性相对传统硬件网络功能设备有所增加. 软件攻击面的扩大，需要更加小心谨慎的 VNF 开发过程和更加全面的软件测试步骤.

5.3　NFV 技术未来研究方向展望

NFV 技术作为一种新颖的网络功能实现方案，在学术界和工业界引起广泛的关注，其优点获得了学术界和工业界的认可. 我们认为 NFV 技术在未来的研究工作会偏重于以下几个方面：

5.3.1　NFV 系统性能优化

在 NFV 技术兴起之后，电信产业由传统网络体系架构逐渐向虚拟网络架构转型. 但为 VNF 提供高效稳定的网络性能依然困难. Wang 和 Ng 等人[64]测试了亚马逊 EC2 云服务的端对端网络性能，报告指出虚拟化环境中处理器的共享会导致 TCP/UDP 吞吐量在几十毫秒的范围内出现 0～10 Gbps 的波动，且不同 EC2 实例之间的传输延迟差异超过普通延迟的 100 倍. 因此，由虚拟化技术引起的不稳定网络特征会影响到虚拟设施的性能与部署. 为了使 NFV 平台达到实际网络环境的性能需求，NFV 性能优化仍然是未来研究的重要方向之一. NFV 性能优化包括 VNF 运行环境性能优化和 VNF 处理性能优化两个部分.

在 VNF 运行环境方面，首先可以利用现成 Linux 的 NAPI 和 Intel 的 DPDK 来提升 VNF 的网络性能. 其能够在网络负载高的环境下屏蔽数据包中断，并采用轮询网络设备的方式进行数据包的处理. 通过避免频繁处理数据包处理中断，可以大幅度提升 VNF 的网络性能. 其次，针对不同的虚拟机和虚拟机管理器，可以根据需求从网络驱动和系统网络栈的实现方面进行一些定制优化. 通过两种方式的结合，可以实现 NFV 平台的网络性能优化.

在 VNF 处理性能优化方面，针对不同的 VNF，可以通过 VNF 内部更细粒度包处理模块的复用和并行来实现不同 VNF 性能的进一步优化.

5.3.2　VNF 负载均衡和状态管理

高效的 VNF 负载均衡和容错技术是 NFV 平台提供稳定网络服务的有效保障. 目前 VNF 负载均衡和容错恢复机制都采用实时监测、及时响应的模式. 这种模式无法对网络流量过载和 VNF 运行

时错误做出及时的预测并尽早采取行动.

通过使用统计学习理论和机器学习方法,可以对 VNF 运行的历史数据进行分析和挖掘,对经过 VNF 的网络流量峰值进行预测,提前做好负载均衡的准备工作(例如,资源分配、VNF 扩展等),达到高效的网络流量负载均衡.同理,对系统进行基于历史数据的异常检测可以及时发现 VNF 的运行异常,进而及时采取相应措施,避免出现 VNF 完全失效的情况.

5.3.3 资源分配和调度策略

NFV 管理平台需要合理部署、控制和管理 VNF 来满足不同网络环境中的动态需求,例如,资源利用率的最优化、对网络攻击的动态防御等.如何根据 NFV 平台不同部署场景下具体的限制条件和优化目标,作出最优的 VNF 部署和迁移决策,以提升 NFV 平台整体性能,依然是 NFV 技术研究的热点.

如何综合考虑 VNF 服务链的构造、VNF 部署以及 VNF 调度三个相互关联的资源分配阶段,以协调一致的方式执行三个阶段来实现合理的 VNF 部署和资源分配仍然是一个挑战.目前尚未提出相关的算法协调解决三个阶段的资源分配问题.此外,目前的 VNF 部署策略鲜有考虑 NFV 平台的能源消耗问题,在未来的 VNF 部署策略中,我们应该综合考虑 NFV 平台中的能源消耗问题,以减少整个 NFV 平台的能源消耗.

由于 VNF 需要根据变化的网络环境来进行动态的实例化和迁移,因此如何高效精准地实现 VNF 迁移成为难题.在 VNF 迁移策略方面,由于 VNF 在迁移的过程中会存在资源开销,例如,迁移 VNF 镜像和状态所需要的网络带宽等,如何根据具体的网络环境,设计高效精准的迁移算法来权衡 VNF 迁移过程中的资源开销和收益成为解决此类问题的关键.在系统实现方面,虽然 ClickOS 等 NFV 运行平台对 VNF 的性能进行了优化,实现了极小的内存占用空间和快速的 VNF 初始化.但他们尚未实现 VNF 的高效迁移.因此,通过对轻量级的 VNF 运行平台,虚拟机管理器以及虚拟机进行系统层面的优化,实现高效的 VNF 迁移成为未来的研究目标.

5.3.4 NFV 策略实施和验证

在策略实施和验证方面,目前的研究工作已经能够解决在 NFV 平台中策略的正确实施和验证问题.然而,在策略实施验证方面,还存在一些问题尚待解决.

网络流量处理延迟往往决定网络服务质量.在 NFV 外包场景下,由于 NFVI 层的所有设备都在云服务提供商的控制之下,用户无法得知 VNF 运行设备的实际情况.因此用户如何在 NFV 外包场景下全面、高效地验证云平台上 NFV 策略实施的性能成为有待解决的问题.

5.3.5 NFV 管理系统设计及优化

在 NFV 管理系统设计及优化方面,目前提出的 NFV 平台管理框架可以胜任 NFV 平台所需的对 VNF 和 NFVI 资源的管理功能.但在 NFV 系统的性能方面,仍然存在优化提升的空间.

目前针对 NFV 平台性能优化的工作在方法上都相互独立,从不同的角度对 NFV 平台的网络流量处理性能进行优化.然而,不同方面的性能优化可以进行有效地结合,例如,VNF 功能模块复用和并行执行可以有效地进行结合,以进一步提升 NFV 平台的性能,但各种功能模块之间的可复用程度以及并行算法仍旧需要进一步研究总结.

5.3.6 NFV 技术中的安全问题

NFV 技术中的 VNF 运行于由软硬件组成的 NFVI 之上,当 NFVI 中的任意部分出现安全问题时,例如,软硬件安全漏洞等,整个 NFV 平台极有可能受到相关恶意行为的影响.且与由不同硬件网络功能组成的传统网络设施相比,使用标准服务器和虚拟化技术实现的 NFV 平台更容易传播恶意行为.因此,如何在 NFV 平台中检测恶意行为并减小恶意行为对 NFV 平台的影响需要引起学术界的关注.

另外,NFV 技术的兴起使得越来越多不同的软硬件开发商进入网络设施市场来提供用于搭建 NFV 平台的软硬件产品.因此,NFV 平台中的软硬件组件通常由不同开发商提供.如何评估 NFV 平台中软硬件组件的可靠性与安全性成为了未来研究的方向之一.

最后,由于在支持多租户的 NFV 平台中,不同租户共享平台上的网络资源和计算资源.因此,协调控制网络数据转发层和 NFV 平台控制层,高效安全地实现多租户之间的资源隔离,成为增强 NFV 平台安全性的挑战.

5.3.7 NFV 技术外包模式流量隐私保护

利用 NFV 技术,对网络功能进行外包为现代

企业带来了诸多好处.对于大中型企业来说,网络功能外包减轻了巨大的网络功能运维负担.对于小型企业来讲,网络功能外包为其提供了享受更多样化网络服务的机会,由于相关网络功能设备的复杂性和昂贵的价格,小型企业通常无法在企业网中部署昂贵的硬件网络功能设备.但 NFV 技术的外包模式存在企业及用户隐私问题.其中主要包含隐私问题以及外包决策问题.

首先,企业需要向外包服务平台提交需要搭建的网络拓扑,保证企业网络拓扑的隐私性是提高企业网络安全的关键措施之一;其次,由于用户网络流量需要经过不可信云服务提供商管理的 VNF 处理,因此,用户流量内容的隐私无法得到保障.针对 NFV 技术外包场景中涉及的用户流量隐私暴露问题,目前有工作提出使用加密搜索的方式[50-51]和可信计算硬件的方式[58-61]来保护用户流量隐私.但加密搜索的方式仅能实现少部分网络功能,而使用可信硬件的方式受到可信硬件局限性的限制.例如,系统调用开销大、能同时部署的 VNF 数量少等.因此,实现一种针对所有网络功能的高效流量隐私保护方案对 NFV 技术的实际部署有着重要的意义.

6　结束语

本文首先介绍了 NFV 技术的标准结构,然后对构建及运行环境优化、NFV 管理系统设计及优化、策略实施与验证、资源分配和迁移策略、NFV 负载均衡和状态管理技术、NFV 架构中的安全问题几个方面的学术研究状况进行了着重分析,最后介绍了 NFV 部署场景,总结了 NFV 技术相对于传统网络功能设备的优势和不足,并对 NFV 技术未来的研究方向进行了展望.综上所述,NFV 技术会逐渐被现代企业所接受,为企业提供优质可靠的网络服务.

参　考　文　献

[1] Sherry J, Hasan S, Scott C, et al. Making middleboxes someone else's problem: Network processing as a cloud service. ACM SIGCOMM Computer Communication Review, 2012, 42(4): 13-24

[2] Guerzoni R. Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action, introductory white paper//Proceedings of the SDN and OpenFlow World Congress. Darmstadt, Germany, 2012: 5-7

[3] Mijumbi R, Serrat Fernández J, Gorricho Moreno J L. Self-managed resources in network virtualisation environments//Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management. Ottawa, Canada, 2015: 1099-1106

[4] Kreutz D, Ramos F M V, Verissimo P E, et al. Software-defined networking: A comprehensive survey. Proceedings of the IEEE, 2015, 103(1): 14-76

[5] Bifulco R, Canonico R, Brunner M, et al. A practical experience in designing an OpenFlow controller//Proceedings of the 2012 European Workshop on Software Defined Networking Software Defined Networking (EWSDN). Darmstadt, Germany, 2012: 61-66

[6] Hawilo H, Shami A, Mirahmadi M, et al. NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC). IEEE Network, 2014, 28(6): 18-26

[7] Drutskoy D, Keller E, Rexford J. Scalable network virtualization in software-defined networks. IEEE Internet Computing, 2013, 17(2): 20-27

[8] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: Enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 2008, 38(2): 69-74

[9] Martins J, Ahmed M, Raiciu C, et al. ClickOS and the art of network function virtualization//Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation. Seattle, USA, 2014: 459-473

[10] Panda A, Han S, Jang K, et al. NetBricks: Taking the V out of NFV//Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Savannah, USA, 2016: 203-216

[11] Kohler E, Morris R, Chen B, et al. The Click modular router. ACM Transactions on Computer Systems, 2000, 18(3): 263-297

[12] Rizzo L, Lettieri G. Vale, a switched Ethernet for virtual machines//Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies. Nice, France, 2012: 61-72

[13] Lattner C, Adve V. LLVM: A compilation framework for lifelong program analysis & transformation//Proceedings of the International Symposium on Code Generation and Optimization: Feedback-Directed and Runtime Optimization. Palo Alto, USA 2004: 75-86

[14] Gordon C S, Parkinson M J, Parsons J, et al. Uniqueness and reference immutability for safe parallelism. Association for Computing Machinery's Special Interest Group on Programming Languages Notices, 2012, 47(10): 21-40

[15] Palkar S, Lan C, Han S, et al. E2: A framework for NFV applications//Proceedings of the 25th Symposium on Operating Systems Principles. Monterey, USA, 2015: 121-136

[16] Han S, Jang K, Panda A, et al. SoftNIC: A software NIC to augment hardware. Electrical Engineering and Computer Science Department, University of California, Berkeley, CA, USA: Technology Report: UCB/EECS-2015-155, 2015

[17] Kernighan B W, Lin S. An efficient heuristic procedure for partitioning graphs. The Bell System Technical Journal, 1970, 49(2): 291-307

[18] Sekar V, Egi N, Ratnasamy S, et al. Design and implementation of a consolidated middlebox architecture//Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. Vancouver, Canada, 2012: 24-24

[19] Bremler-Barr A, Harchol Y, Hay D. OpenBox: A software-defined framework for developing, deploying, and managing network functions//Proceedings of the Conference of the ACM Special Interest Group on Data Communication. Salvador, Brazil, 2016: 511-524

[20] Sun C, Bi J, Zheng Z, et al. NFP: Enabling network function parallelism in NFV//Proceedings of the Conference of the ACM Special Interest Group on Data Communication. Los Angeles, USA, 2017: 43-56

[21] Fayazbakhsh S K, Chiang L, Sekar V, et al. Enforcing network-wide policies in the presence of dynamic middlebox actions using FlowTags//Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. Seattle, USA, 2014: 533-546

[22] Gember A, Krishnamurthy A, John S S, et al. Stratos: A network-aware orchestration layer for virtual middleboxes in clouds. arXiv preprint, 2013, 1305(0209): 1-13

[23] Panda A, Lahav O, Argyraki K, et al. Verifying isolation properties in the presence of middleboxes. Archive (arXiv), 2014, 1409(7687): 1-12

[24] Ying Z, Wu W, Banerjee S, et al. SLA-Verifier: Stateful and quantitative verification for service chaining//Proceedings of the IEEE International Conference on Computer Communication. Atlanta, USA, 2017: 328-341

[25] Fayaz S K, Yu T, Tobioka Y, et al. Buzz: Testing context-dependent policies in stateful networks//Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16). Boston, USA, 2016: 275-289

[26] Mehraghdam S, Keller M, Karl H. Specifying and placing chains of virtual network functions//Proceedings of the Cloud Networking (CloudNet). Luxembourg, The Grand Duchy of Luxembourg, 2014: 7-13

[27] Basta A, Kellerer W, Hoffmann M, et al. Applying NFV and SDN to LTE mobile core gateways, the functions placement problem//Proceedings of the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges. Chicago, USA, 2014: 33-38

[28] Luizelli M C, Bays L R, Buriol L S, et al. Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions//Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). Ottawa, Canada, 2015: 98-106

[29] Riggio R, Bradai A, Rasheed T, et al. Virtual network functions orchestration in wireless networks//Proceedings of the 2015 11th International Conference on Network and Service Management. Barcelona, Spain, 2015: 108-116

[30] Amaldi E, Coniglio S, Koster A M C A, et al. On the computational complexity of the virtual network embedding problem. Electronic Notes in Discrete Mathematics, 2016, 52: 213-220

[31] Moens H, De Turck F. VNF-P: A model for efficient placement of virtualized network functions//Proceedings of the 10th International Conference on Network and Service Management (CNSM). Rio de Janeiro, Brazil, 2014: 418-423

[32] Riggio R, Rasheed T, Narayanan R. Virtual network functions orchestration in enterprise WLANs//Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). Ottawa, Canada, 2015: 1220-1225

[33] Ma W, Sandoval O, Beltran J, et al. Traffic aware placement of interdependent NFV middleboxes//Proceedings of the IEEE International Conference on Computer Communication. Atlanta, USA, 2017: 631-645

[34] Mijumbi R, Serrat Fernández J, Gorricho Moreno J L, et al. Design and evaluation of algorithms for mapping and scheduling of virtual network functions//Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft). Bologna, Italy, 2015: 1-9

[35] Fayaz S K, Tobioka Y, Sekar V, et al. Bohatei: Flexible and elastic DDoS defense//Proceedings of the USENIX Security Symposium. Washington DC, USA, 2015: 817-832

[36] Rajagopalan S, Williams D, Jamjoom H, et al. Split/Merge: System support for elastic execution in virtual middleboxes//Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13). Lombard, IL, 2013: 227-240

[37] Gember-Jacobson A, Viswanathan R, Prakash C, et al. OpenNF: Enabling innovation in network function control. ACM SIGCOMM Computer Communication Review, 2015, 44(4): 163-174

[38] Khalid J, Gember-Jacobson A, Michael R, et al. Paving the way for NFV: Simplifying middlebox modifications using StateAlyzr//Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16). Boston, USA, 2016: 239-253

[39] Deng J, Li H, Hu H, et al. On the safety and efficiency of virtual firewall elasticity control//Proceedings of the 24th Network and Distributed System Security Symposium (NDSS'17). San Diego, USA, 2017: 235-248

[40] Gill P, Jain N, Nagappan N. Understanding network failures in data centers: measurement, analysis, and implications// Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM). Toronto, Canada, 2011: 350-361

[41] Sherry J, Gao P X, Basu S, et al. Rollback-recovery for middleboxes. ACM SIGCOMM Computer Communication Review, 2015, 45(4): 227-240

[42] Horwitz S, Reps T, Binkley D. Interprocedural slicing using dependence graphs. ACM Transactions on Programming Languages and Systems, 1990, 12(1): 26-60

[43] Andersen L O. Program Analysis and Specialization for the C Programming Language[Ph. D. dissertation]. University of Copenhagen, Copenhagen, Denmark, 1994

[44] Steensgaard B. Points-to analysis in almost linear time// Proceedings of the 23rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. St. Petersburg Beach, USA, 1996: 32-41

[45] Yuan X, Duan H, Wang C. Bringing execution assurances of pattern matching in outsourced middleboxes//Proceedings of the 2016 IEEE 24th International Conference on Network Protocols. Singapore, 2016: 1-10

[46] Zhang X, Li Q, Wu J, et al. Generic and agile service function chain verification on cloud//Proceedings of the 2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS). Barcelona, Spain, 2017: 1-10

[47] Sherry J, Lan C, Popa R A, et al. Blindbox: Deep packet inspection over encrypted traffic. ACM SIGCOMM Computer Communication Review. 2015, 45(4): 213-226

[48] Lan C, Sherry J, Popa R A, et al. Embark: Securely outsourcing middleboxes to the cloud//Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16). Santa Clara, USA, 2016: 255-273

[49] Shih M W, Kumar M, Kim T, et al. S-NFV: Securing NFV states by using SGX//Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization. New Orleans, USA, 2016: 45-48

[50] Coughlin M, Keller E, Wustrow E. Trusted click: Overcoming security issues of NFV in the cloud//Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization. Scottsdale, USA, 2017: 31-36

[51] Han J, Kim S, Ha J, et al. SGX-Box: Enabling visibility on encrypted traffic using a secure middlebox module//Proceedings of the 1st Asia-Pacific Workshop on Networking. Hong Kong, China, 2017: 99-105

[52] Duan H, Yuan X, Wang C. LightBox: SGX-assisted secure network functions at near-native speed. Archive (arXiv). 2017, 1706(06261): 0-10

[53] Golle P, Mironov I. Uncheatable distributed computations// Proceedings of the Cryptographers' Track at the RSA Conference. Berlin, Germany, 2001: 425-440

[54] Hoekstra M, Lal R, Pappachan P, et al. Using innovative instructions to create trustworthy software solutions// Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy. Tel-Aviv, Israel, 2013: 11-19

[55] McKeen F, Alexandrovich I, Berenzon A, et al. Innovative instructions and software model for isolated execution// Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy. Tel-Aviv, Israel, 2013: 20-28

[56] Anati I, Gueron S, Johnson S, et al. Innovative technology for CPU based attestation and sealing//Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy. Tel-Aviv, Israel, 2013, 40-51

[57] Armbrust M, Fox A, Griffith R, et al. A view of cloud computing. Communications of the ACM, 2010, 53(4): 50-58

[58] Kumar S, Tufail M, Majee S, et al. Service function chaining use cases in data centers. Internet Engineering Task Force Service Function Chain Work Group, 2015, 1(1): 1-18

[59] Yang M, Liy Y, Jiny D, et al. OpenRAN: A software-defined RAN architecture via virtualization. ACM SIGCOMM Computer Communication Review, 2013, 43(4): 549-550

[60] Yang M, Li Y, Hu L, et al. Cross-layer software-defined 5G network. Mobile Networks and Applications, 2015, 20(3): 400-409

[61] Bhaumik S, Chandrabose S P, Jataprolu M K, et al. CloudIQ: A framework for processing base stations in a data center// Proceedings of the 18th Annual International Conference on Mobile Computing and Networking. Istanbul, Turkey, 2012: 125-136

[62] Yang M, Li Y, Jin D, et al. Software-defined and virtualized future mobile and wireless networks: A survey. Mobile Networks and Applications, 2015, 20(1): 4-18

[63] Yang M, Li Y, Li B, et al. Service-oriented 5G network architecture: An end-to-end software defining approach. International Journal of Communication Systems, 2016, 29(10): 1645-1657

[64] Wang G, Ng T S E. The impact of virtualization on network performance of amazon EC2 data center//Proceedings of the IEEE International Conference on Computer Communication. San Diego, USA, 2010: 1-9

WANG Jin-Wen, born in 1994, M. S. candidate. His research interests include NFV, network security, system security, trusted computing.

ZHANG Xiao-Li, born in 1993, Ph. D. candidate. Her research interests include SDN, NFV, network security.

LI Qi, born in 1979, Ph. D., associate professor. His research interests include Internet and cloud security, mobile security, big data security.

WU Jian-Ping, born in 1953, Ph. D., professor, Ph. D. supervisor, member of Chinese Academy of Engineering. His research interests include network protocol test, network management, network architecture.

JIANG Yong, born in 1975, Ph. D., professor, Ph. D. supervisor. His research interests include network architecture, next generation Internet.

Background

Enterprises improve their network performance and securities by deploying different kinds of middleboxes, such as firewall, Intrusion Detection System (IDS), WAN Optimization, etc. However, with the increase of the size of the enterprise network, the cost of middleboxes deployment and maintenance increases significantly. To address these problems, Network Function Virtualization (NFV) has been proposed to deploy the software network functions on the standard commercial server with high performance to save the hardware cost and simplify the network function maintenance. In recent years, NFV technology has attracted significant attention from academic and industrial communities. In this paper, we provide a comprehensive survey of existing studies of NFV technology.

In this paper, we first introduced the standard NFV architecture proposed by ETSI. Then we classify the existing research results of NFV technology. And we discuss the research works on NFV technology from following aspects: building blocks of NFV, i.e., virtual network function (VNF) construction, and the running environment optimization in NFV, the design of NFV management system and its optimization, policy enforcement and verification, resource allocation and migration, load balance and state management technology, and NFV security. Finally, this paper summaries the advantages and shortcomings of NFV technology compared with traditional network functions, introduces the application scenario cases of NFV and predicts the research directions of NFV in the future, according to the ETSI standard architecture of NFV.