

一种灵活控制服务功能链的新方法

张顺康¹, 陈 鸣¹, 许 博²

(1. 南京航空航天大学 计算机科学与技术学院, 江苏 南京 211106;

2. 陆军工程大学 指挥控制工程学院, 江苏 南京 210007)

摘 要:近年来,随着网络功能虚拟化(network function virtualization,NFV)技术的快速发展,虚拟化网络服务的种类及数量急剧增长。在NFV网络中,通常利用服务功能链(service function chaining,SFC)将一组有序的虚拟网络功能连接起来提供某种网络服务。该文研究了如何利用SFC更为灵活、细粒度地控制NFV网络服务。首先,定义了增强服务功能链(enhanced service function chaining,ESFC)的概念,给出了一种细粒度描述ESFC的方法;其次,设计了灵活控制ESFC的系统模型,能够根据用户需求及时、准确地调整ESFC,从而灵活、精细地控制NFV网络服务功能;最后,基于NFV技术实现了一个支持上述模型的原型系统。原型系统的试验结果表明,该模型可以在极短时间内响应大量用户请求,灵活、精细地调整ESFC,使得NFV网络服务呈现出多样化功能,验证了ESFC机制的可行性。

关键词:网络功能虚拟化;增强服务功能链;NFV网络服务;描述方法;灵活控制

中图分类号:TP393

文献标识码:A

文章编号:1673-629X(2020)12-0001-07

doi:10.3969/j.issn.1673-629X.2020.12.001

A Novel Method of Flexibly Controlling Service Function Chaining

ZHANG Shun-kang¹, CHEN Ming¹, XU Bo²

(1. School of Computer Science and Technology, Nanjing University of Aeronautics and

Astronautics, Nanjing 211106, China;

2. School of Command Control Engineering, Army Engineering University, Nanjing 210007, China)

Abstract: In recent years, with the rapid development of network function virtualization (NFV) technology, the variety and number of virtualized network services have grown dramatically. In NFV networks, service function chaining (SFC) is usually used to connect a set of ordered virtual network functions to provide certain network services. We study how to use SFC to control NFV network services more flexibly and finely. The concept of enhanced service function chaining (ESFC) is first defined, and a fine-grained description method of ESFC is given. Secondly, the system model for flexible control of ESFC is designed, which can adjust ESFC timely and accurately according to user needs, so as to flexibly and finely control NFV network services. Finally, a prototype system supporting the above model is implemented based on NFV technology. The test results show that this model can respond to a large number of user requests in a quite short time, and adjust the ESFC flexibly and finely, so that NFV network service presents diversified functions, which verifies the feasibility of ESFC mechanism.

Key words: network function virtualization; enhanced service function chaining; NFV network service; description method; flexible control

0 引 言

在计算机网络中,网络服务是运行在应用层的一种程序,能够提供数据存储、处理、表示、通信或其他分布式计算能力,网络服务可认为是某种网络功能的集合,是网络创新最为活跃的领域,是网络价值的体现。在网络功能虚拟化(network function virtualization, NFV)环境下,NFV通过软硬件解耦和功能抽象,形成

不依赖于物理设备的虚拟网络功能(virtual network function, VNF),VNF运行在虚拟机或容器中,通过服务功能链(service function chaining, SFC)将若干个VNF互连,形成一条具有特定功能、性能的网络路径,用于提供端到端的NFV网络服务^[1-2]。SFC可以灵活部署网络服务,有效降低网络建设与运维的成本,加快新服务的应用和升级。SFC是当前学术界的研究热

收稿日期:2020-01-10

修回日期:2020-05-13

基金项目:国家自然科学基金(61772217)

作者简介:张顺康(1995-),男,硕士,CCF会员(89140G),研究方向为计算机网络;通讯作者:陈 鸣(1956-),男,博士,教授,研究方向为网络测量、未来网络。

点,2015 年至今的 HotMiddlebox 研讨会上,收录了多篇 SFC 方面的论文,甚至计算机网络的顶级会议——ACM SIGCOMM,也将 SFC 列为专题来研讨。分析 SFC 的相关工作,对于 SFC 的研究主要分为以下 3 个方面:SFC 标准化研究、SFC 编排研究和 SFC 应用研究^[3-6]。

在 SFC 应用研究领域,目前研究者比较关注的是如何降低时延、增强服务可靠性以及灵活调整服务功能。该文研究如何灵活调整 NFV 网络服务的功能,提出了两个问题并给出了解决方案。第一,目前 SFC 仅通过定义 VNF 序列的执行顺序来控制网络服务功能,而事实上,VNF 的运行参数及执行时间段等都会对 NFV 网络服务的功能产生影响;第二,如何让系统理解用户对 NFV 网络服务的要求,并且能够自动将其转换为对 VNF 序列的控制操作。针对第一个问题,文中提出了增强服务功能链(enhanced service function chaining, ESFC)的概念,并给出了一种基于 XML 语言的描述方法,通过自定义用户义的标签描述 VNF 序列的执行顺序、运行参数以及执行时间段等,从而精细地定制 ESFC,满足用户细粒度控制 NFV 网络服务的需求;对于第二个问题,文中设计了一种灵活控制 ESFC 的系统模型,可以自动解析控制描述文件,并生成对 VNF 序列的控制任务,执行相应的任务,就可以控制特定 VNF 在特定时间内,以特定的参数运行,从而自动调整 ESFC,实现 NFV 网络服务的精细化控制。

1 SFC 相关研究

目前,SFC 的标准化工作主要由 IETF 下的 SFC 工作组完成,该组织在草案^[7]中详细定义了 SFC 的框架结构、适用场景、数据包分组格式等。SFC 编排问题已经被证明是一个 NP 难问题^[8],可将 SFC 编排问题转化为整数线性规划模型,然后设计启发式算法求最优解,如 Beck 等人^[9]设计了一种基于弹性约束的启发式算法,根据 SFC 策略需求可以快速得到最优解,有效降低计算资源消耗;李丹等人^[10]则提出了一种基于最优加权图匹配的算法,将 SFC 抽象为功能拓扑图,利用特征向量分解算法获得匹配结果,并利用爬山算法对结果进行优化,可以显著降低带宽资源消耗,并且算法复杂度很低,具有较高的时效性。

当前对于 SFC 应用问题的研究主要集中在如何降低时延以及保证服务的可靠性,如 Oljira 等人^[11]通过一些仿真试验测试了虚拟化对网络时延的影响,仿真结果表明,对于端到端的网络时延,虚拟化环境下的数值比非虚拟化环境要大;陈卓等人^[12]抽象了一个面向业务流服务时延优化的数学模型,并基于遗传算法提出了一种启发式 SFC 重迁移配置策略,并通过对比

试验证明了模型和算法的有效性。而为了提高服务的可靠性,通常采用 VNF 备份机制,如 Fan 等人^[13]基于 VNF 备份机制提出了一种新颖的在线算法,可以在资源消耗最小的同时,确保网络服务的可靠性;兰巨龙等人^[14]通过将网络拓扑抽象为有向图进行建模分析,提出了一种基于路径选择的可靠性算法;韩青等人^[15]提出了一种最优备份拓扑生成算法,可以在保证服务可靠性的前提下,减少 VNF 备份数量,提高资源利用率。

综上所述,虽然目前关于 SFC 标准化以及 SFC 编排的研究已经相对成熟,在 SFC 应用方面也取得了一定的研究成果,提出的一些算法和模型也具备一定的可行性,但是,由于上述算法基本都是通过仿真系统进行验证的,在真实网络中的效果还有待商榷。另外当前也缺少利用 SFC 灵活、精细地控制 NFV 网络服务的研究。文中针对这一问题进行了探讨,提出了两个问题并给出了解决方案,同时在 NFV 网络中进行试验,确保了结果的准确性及可靠性。

2 增强服务功能链机制

2.1 相关定义

定义 1:NFV 环境下,所谓 VNF 是一段能够在虚拟机或容器等环境中运行的代码或软件。

定义 2:NFV 环境下,虚拟中间盒(virtual middle-box, VMB)是指具有特定网络功能的 Linux 容器(Linux container, LXC),在 LXC 环境中运行一个或多个 VNF 提供某种网络功能,这种 LXC 即被称为 VMB。

定义 3:NFV 环境下,某 VNF_i 的定义为: $VNF_i = \{F, P\}$,其中 F 是性能与功能集合, P 是运行参数集合。当分组序列通过 VNF_i 时,所呈现的功能及性能集合称为 VNF_i 行为。当前的 VNF 通常认为其行为是不变的,但事实上,若 P 取不同的子集 p_i ,则其可能呈现不同的功能和性能 $F_i = \{f_s | p_i\}$ 。

定义 4:NFV 环境下,某 SFC_j 的定义为: $SFC_j = \cap_i^n x_{Ai}(x) VNF_{ij}$,其中 $x_{Ai}(x)$ 为 VNF_i 的示性函数,表示此 VNF 开启(1)或关闭(0)。当分组序列通过该 SFC_j 时,所呈现的功能及性能集合称为 SFC_j 行为或确定性网络服务。

当前的 SFC 较之传统方式提供网络服务的优势,就在于它可以定制 VNF 的次序,增强网络服务的灵活性。但是 SFC 一旦确定并部署完成,SFC 的行为就是固定的,且 $x_{Ai}(x)$, $x \in A$ 取值为 1。而事实上, VNF_i 的行为是可变的,因此 $x_{Ai}(x)$ 可能取值为 0,并且由定义 3 可知,每个 VNF 的行为都可能随着运行参数的改变而不同。

定义 5:NFV 环境下,当某 SFC 的运行参数 P 可控调整,且 $x_{Ai}(x)$ 可按时间 t 取值 0 或 1,该 SFC 称为增

强服务功能链(ESFC)。

SFC 与 ESFC 的本质区别是,前者仅能定制 VNF 次序,且一旦定制完成,则其行为固定不变;而后者则可通过 VNF 顺序、运行参数集合 P 和随时变化的 $x_{Ai}(x)$ 等要素来定制 SFC,从而灵活、精细地控制 SFC,使得 NFV 网络服务呈现丰富的功能多样性。

定义 6: NFV 环境下,一个网络服务由一个或多个 SFC_j 构成,某网络服务的定义为: $NS = \bigcup_j^m SFC_j = \bigcup_j^m \bigcap_i^n x_{Ai}(x) VNF_{ij}$ 。当分组序列通过该 NFV 网络服务时,所呈现的功能及性能集合称为 NFV 网络服务行为。

2.2 灵活控制 ESFC 的系统模型

引入上述概念使得灵活、精细地控制 NFV 网络服务成为可能,但由于用户对网络服务的需求不断变化,通过人工修改程序来满足网络服务行为调整的方法是不可取的,需要研发一种系统:理解人类用户需求,进而自动将其转换为可执行的网络任务。

为此,基本思路是:定义一种形式化描述 ESFC 的方法,用户利用该方法描述所需的网络服务,并形成控制描述文件;系统读入并解析控制描述文件,进而将其分解为对 VNF 序列的控制任务。

为了具有灵活、精细地控制网络服务的能力,需要系统提供相关的功能支持。该系统的层次模型如图 1 所示,它分为服务定制层、服务编排层以及任务执行层。其中,服务定制层主要供用户定制 NFV 网络服务的控制描述文件,用于解决提出的第一个问题;服务编排层是控制描述文件的解释执行环境,负责编排控制描述文件并生成任务队列,用来解决提出的第二个问题;任务执行层的主要功能就是执行具体的任务以控制 VNF 呈现特定行为,从而灵活调整 ESFC 的功能,实现灵活控制 NFV 网络服务的目标。

对于该灵活控制 ESFC 的系统模型,有以下 3 条假设:

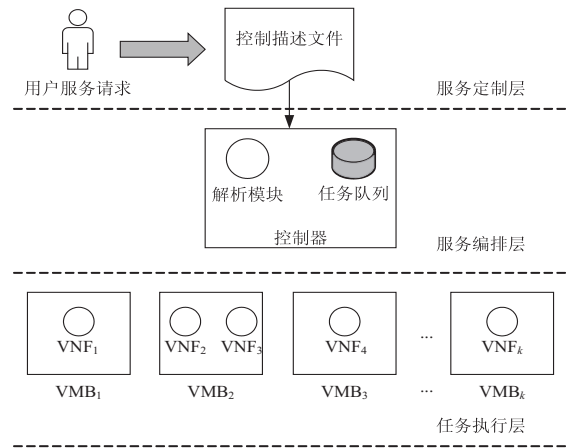


图 1 灵活控制 ESFC 的系统模型

- (1) NFV 网络服务由 ESFC 提供。
- (2) ESFC 已经部署在 NFV 网络中。
- (3) 用户明确 NFV 网络服务需求。

该系统中,执行一个控制任务指调整一个 VNF 的行为,任务队列中的所有任务全部被执行之后,即完成了相应的 ESFC 的调整,即实现了用户的 NFV 网络服务需求。控制任务具有“控制描述”、“解析调度”和“任务执行”3 种状态,系统调度控制任务执行时的有限状态机如图 2 所示。其中,当新的网络服务需求到达时,任务处于“控制描述”状态,采用形式化方法描述 NFV 网络服务行为,定义构成网络服务行为的控制任务可通过描述 ESFC 实现,而事实上是描述构成 ESFC 的 VNF 序列的行为,任务保持在“控制描述”状态直至描述完毕为止;而当完成控制描述文件时,提交控制描述文件,任务处于“解析调度”状态。控制任务处于“解析调度”状态时,当任务队列中某个任务的调度时间到了,提交该任务,转向“任务执行”状态;如果当任务调度时间到了而没有任务可调度,则返回。当任务处于“任务执行”状态,若某个控制任务是一次性任务,则执行完毕后删除该任务;而当该任务需要重复执行时,则需要将该任务重新插入队列的适当位置。

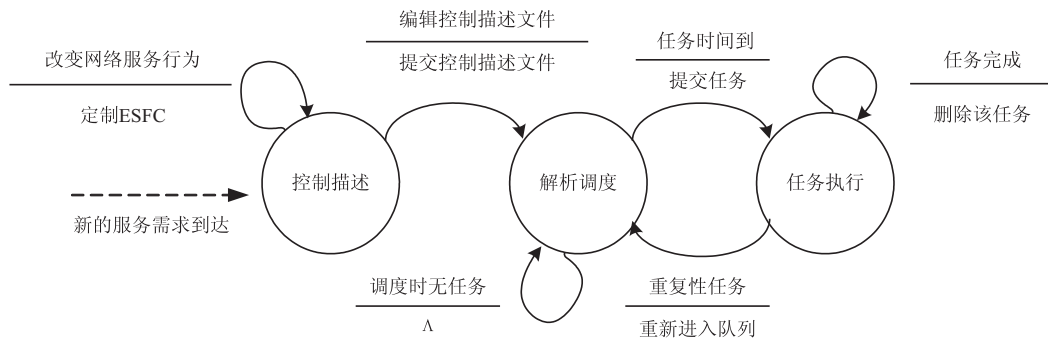


图 2 系统执行控制任务的有限状态机

2.3 形式化描述方法

XML 是一种可扩展标记语言,允许用户自定义标签,同时具有自描述性,可定义各种标识描述信息的所

有元素^[16]。因此为了解决文中提出的第一个问题,提出了一种基于 XML 语言描述 ESFC 的形式化方法,用户可根据 NFV 网络服务需求自主定义标签,从而能够

灵活地基于多个约束条件定义一个 ESFC,满足用户灵活、精细地描述 NFV 网络服务的需求。

文中给出了从 VNF 次序、VNF 执行时间、VNF 运行参数等因素共同定义的 ESFC 的控制描述文件 control.xml。该文件共3层,第1层为根标签,表示此文件用来描述 NFV 网络服务的控制信息,其标签定义

为<NSVMBControl></NSVMBControl>;第2层标签为<strategy ID=“...”></strategy>,定义了一条控制信息的序号;第3层定义了一个 VNF 的控制信息,包括 VNF 所在 VMB 的名称、VNF 名称、VNF 执行时间、VNF 动作、VNF 运行参数等,其具体标签定义如表1所示。

表1 VNF 控制信息的标签定义

标签	含义
<VMBname>...</VMBname>	定义某个 VNF 所在的 VMB 的名称
<VNFname>...</VNFname>	定义该 VNF 的名称
<action>...</action>	定义该 VNF 的行为,可为“start”、“stop”和“restart”,分别表示启动 VNF、停止 VNF 运行和重启 VNF
<category>...</category>	定义该任务的类别,可为“0”、“1 x”和“2 x”,其中“0”和“1 x”都是一次性任务,“1 x”表示 x 秒后执行任务,“2 x”是周期性任务,x 表示任务周期,单位为秒
<trigger>...</trigger>	定义该 VNF 的运行时间
<VNFparams>...</VNFparams>	定义该 VNF 的运行参数
<priority>...</priority>	定义该 VNF 的优先级,用来确定 VNF 的执行次序,在运行时间相同时生效,取值为正整数,数值大的 VNF 优先执行

2.4 控制器设计

上述控制描述文件可解决灵活、细粒度地描述 ESFC 的问题,而为了解决文中提出的第二个问题,基于图1所示的灵活控制 ESFC 的系统模型,设计实现了一个控制器,用来解析控制描述文件,并将其自动转换为控制相应 VNF 行为的任务队列,调度队列执行任务即可及时、准确地调整 ESFC,从而实现 NFV 网络服务的精细化控制。

控制器分为解析模块和任务队列模块两部分,解析模块负责检测控制描述文件的语法格式,并解析描述文件得到 NFV 网络服务的控制描述信息;任务队列模块根据控制描述信息构建任务队列,并调度任务队列的任务在特定时间下发到任务执行模块执行。其中,解析模块的语法检测功能由检测函数 XMLGrammarD()提供,基于 DTD 文件实现,描述文件解析功能由解析函数 analysisXML()提供,基于 Java DOM4j 函数库实现,逐行解析控制描述文件即可得到 NFV 网络服务的控制描述信息;任务队列基于双向链表实现,构造任务队列时,有 VNF 运行时间和 VNF 优先级两个约束条件,因此得到的任务队列中,排在前面的都是执行时间靠前且优先级较高的控制任务。控制器的工作流程如下:

(1) 系统读入控制描述文件,执行函数 XMLGrammarD()检查控制描述文件的语法,若语法正确,执行解析函数;否则,给出错误提示,由人工修改控制描述文件后再重新读入。

(2) 执行完解析函数之后得到 NFV 网络服务的

控制信息,根据控制描述信息生成任务队列,执行调度函数 taskControl()处理任务队列。

(3) 若已到达队首任务的执行时间 startTime,则让队首控制任务出队,获取控制任务信息,然后执行队列函数 taskAdjustment(),此过程中根据任务类别,对该任务进行不同操作,然后开启线程下发任务参数到任务执行模块,从而控制特定 VMB 中特定的 VNF 执行特定的操作;若未到达队首任务的执行时间,则继续等待。

调度任务队列的详细流程如图3所示。

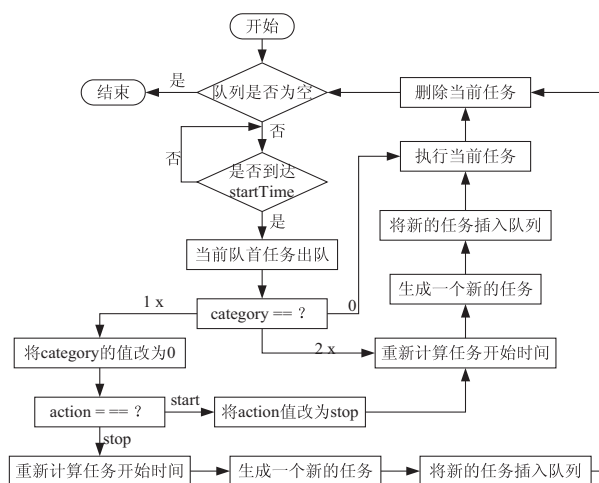


图3 任务队列调度流程

2.5 任务执行模块设计

任务执行模块,用来执行具体的控制任务,改变特定 VNF 的行为,其基于 shell 脚本实现,主要接收任务队列下发的控制任务信息。首先,根据任务信息确定

操作的 VMB 以及 VNF;然后,根据 VNF 的动作信息调整 VNF 行为。任务执行脚本 `control.sh` 位于 `/usr/local/` 目录下,共有 3 个函数:`start()`、`stop()` 和 `restart()`,分别用来执行启动 VNF、停止 VNF 运行以及重启 VNF 等操作。

(1) 执行 `start()` 函数需要传递的参数有 VMB 名称、VNF 名称以及 VNF 运行参数(可选)。

(2) 执行 `stop()` 函数需要传递的参数有 VMB 名称和 VNF 名称。

(3) 执行 `restart()` 函数需要传递的参数有 VMB 名称、VNF 名称以及 VNF 运行参数。

其中 VNF 的运行参数存储在文件 `config.txt` 中,以“参数名=参数值”的格式存储,每个参数占一行,位于相应 VMB 中的 `/usr/local/<VNFname>` 目录下。启动 VNF 实际上就是根据该配置文件使 VNF 的功能生效;停止 VNF 运行实际上是根据 `VNFname` 的值关闭相应的进程;重启 VNF 实际上就是更新该配置文件,并以新的配置文件为参数让 VNF 的功能生效。

3 原型系统试验及结果分析

根据第 2 节提出的灵活控制 ESFC 的系统模型,本节基于 LXC 技术在一台服务器上建立了原型系统,通过对原型系统的试验,验证了基于 XML 语言描述 NFV 网络服务需求的方法的可行性,以及 ESFC 机制的有效性。

3.1 原型系统组成

原型系统运行在一台型号为 Intel(R) Xeon(R) X5647 的服务器上,服务器的主频为 2.93 GHz,内存为 32 G,CPU 为四核,安装 Ubuntu 16.04 server 版系统。原型系统的 NFV 网络拓扑结构如图 4 所示,NFV 网络中的所有虚拟网络节点都是通过 LXC 中运行 VNF 得到的,所有链路的带宽均为 1 000 Mbps,在 LXC 中运行 Quagga^[17] 软件得到虚拟路由器 $R_1 \sim R_5$,所有路由器中都运行 OSPF 协议^[18];在 LXC 运行 Open

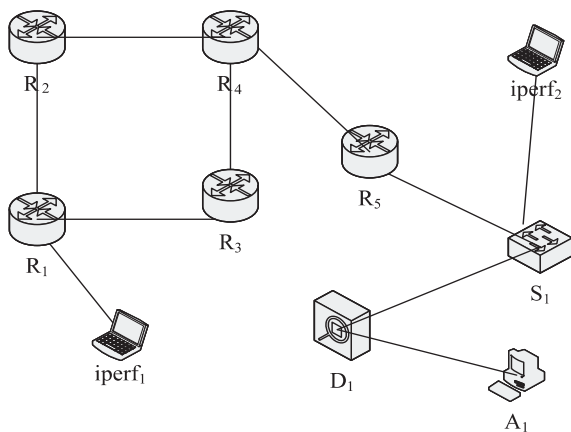


图 4 测试原型系统的 NFV 网络

vSwitch 软件得到交换机 S_1 ;在 LXC 中运行 IPERF 软件得到虚拟主机 `iperf1` 和 `iperf2`,用来发送或接收 TCP/UDP 流量;在 LXC 中运行 TFN2K^[19] 软件得到可以发动 DDoS 攻击中间盒 A_1 ,可发送 DDoS 攻击流;在 LXC 中运行 DDoS 防御程序^[20] 得到 DDoS 防御中间盒 D_1 ,可用来检测并防御 DDoS 攻击。

3.2 系统功能验证及分析

为了验证灵活控制 ESFC 的系统能够根据控制描述文件准确地调整 ESFC,从而按需灵活控制 NFV 网络服务,在此 NFV 网络中,部署了两条 ESFC:`iperf2→iperf1` 和 $A_1 \rightarrow D_1 \rightarrow \text{iperf}_1$,通过以下试验来验证灵活控制 ESFC 的系统功能,试验开始前,ESFC 的功能节点中的 VNF 都未运行。

试验过程如下:

(1) 在 $T_0=0$ s 时,启动 `iperf2` 和 `iperf1`,并让 `iperf2` 向 `iperf1` 发送带宽为 100 Mbps 的 TCP 流。

(2) 在 $T_1=10$ s 时,改变 `iperf2` 中 IPERF VNF 的运行参数,让其向 `iperf1` 发送带宽为 200 Mbps 的 UDP 流。

(3) 在 $T_2=20$ s 时,启动 A_1 中的 TFN2K VNF,让其对 `iperf1` 发动 DDoS 攻击,攻击方式为 TCP flood。

(4) 在 $T_3=30$ s 时,启动 D_1 中的防御程序。

(5) 在 $T_4=40$ s 时,结束试验。

编辑表示此次试验过程的控制描述文件,并将其提交给原型系统执行,统计这段时间内,`iperf1` 端口捕获分组速率的变化情况,试验重复了 30 次,取其平均值作为 `iperf1` 端口捕获分组速率的数值。图 5 展示了 `iperf1` 端口捕获分组速率的变化曲线。

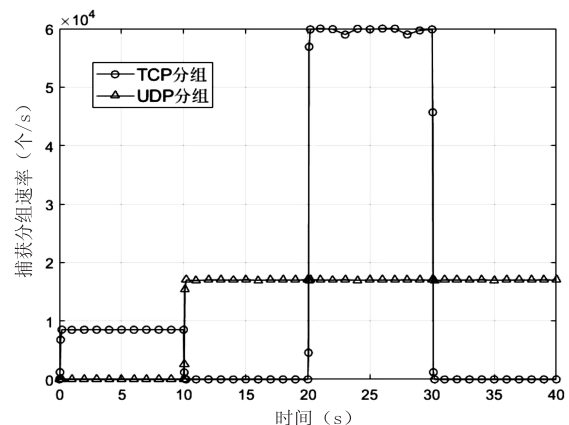


图 5 通过系统控制 ESFC 的行为

由图 5 可见,在 $T_0 \sim T_1$ (0 ~ 10 s) 时间段内,`iperf1` 端口捕获分组的速率大约在 8 500 个报文/秒左右,报文类型为 TCP;在 $T_1 \sim T_2$ (10 ~ 20 s) 时间段内,`iperf1` 端口捕获分组的速率约为 17 000 个报文/秒,报文类型为 UDP;在 $T_2 \sim T_3$ (20 ~ 30 s) 时间段内,`iperf1` 端口捕获 UDP 分组的速率基本不变,仍为 17 000 个报文/

秒,捕获 TCP 分组的速率则迅速增加,约为 60 000 个报文/秒;在 $T_3 \sim T_4$ (30 ~ 40 s) 时间段内,iperf₁ 端口捕获 UDP 分组的速率保持不变,仍为 17 000 个报文/秒,而捕获 TCP 分组的速率则迅速下降到 0。并且在时刻 T_0 、 T_1 、 T_2 、 T_3 ,捕获分组的速率变化基本都在极短时间内完成,此后基本保持稳定。分析捕获分组速率的变化曲线可知,在 T_0 时刻,iperf₁ 接收到带宽为 100 Mbps 的 TCP 报文;在 T_1 时刻 iperf₁ 接收到带宽为 200 Mbps 的 UDP 报文;在 T_2 时刻 iperf₁ 遭受了 TCP flood 攻击;在 T_3 时刻启动了 DDoS 防御中间盒 D_1 ,因此,DDoS 攻击报文不能通过交换机 S_1 转发到 iperf₁。可以看到,试验过程中,相关 VNF 的行为与预定义的控制描述文件完全一致。

试验小结:基于 XML 语言编辑控制描述文件可以表示复杂、多变的 NFV 网络服务需求,并且该系统读入控制描述文件,可以按照预期调控相关 VNF 的行为,从而按需调整 ESFC,进而实现灵活、精细地控制 NFV 网络服务的目标。

3.3 系统性能验证及分析

为了测试通过系统控制 VNF 行为的及时性,将 VNF 行为预定发生时间与实际执行时间的间隔称为调度时间,以其作为评价指标。通过该系统反复测试了控制不同的 VNF 执行不同操作时的调度时间,最后得到如表 2 所示的试验结果。

表 2 三种 VNF 的调度时间 ms

VNF 操作	IPERF	TFN2K	DDoS 防御
start	38	41	40
stop	40	39	41
restart	100	101	102

由表 2 可见,调度时间与 VNF 类型基本无关,但与 VNF 行为有关。启动或停止 VNF 的调度时间约为 40 ms,而重启 VNF 的调度时间大致为 100 ms。这说明,当同时刻仅有一个控制任务需要被执行时,该系统可以在极短时间内完成特定 VNF 行为的控制。

但在实际应用中,用户可能需要同时改变大量 VNF 的行为,为此,让该系统在同一时刻分别调度 100、200、300、400、500 和 800 个任务进行试验,这些控制任务并发执行,此时统计处理完成同一时刻执行的所有任务所需要的调度时间。图 6 显示了并发执行任务的测试结果。

由图 6 可见,任务调度时间与同一时刻需要执行的任务数量大致呈线性增长的关系。该系统在 1 秒内可以同时处理大约 200 个动作为“restart”的 VNF 操作,800 个动作为“start”或“stop”的 VNF 操作,因此该系统应当能满足大部分实际网络服务的应用需求。

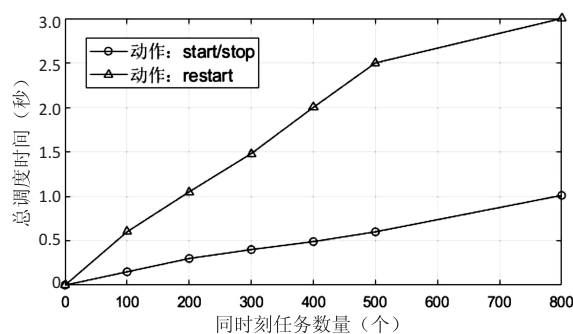


图 6 并发执行多个任务所需的调度时间

为了测试该系统工作时对宿主机的影响,在试验过程中,统计了宿主服务器资源的使用情况。图 7 展示了系统运行过程中宿主服务器的 CPU 使用率及内存占用率的变化情况。

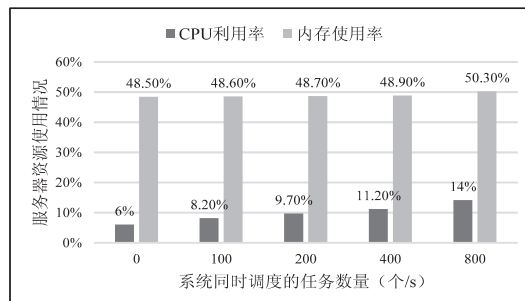


图 7 宿主服务器资源与任务数量的关系

由图 7 可见,尽管宿主服务器的 CPU 利用率和内存占用率随任务数量的增加而上升,但即使在调度 800 个任务时,内存占用率的增长也没有超过 1%,CPU 利用率最大也没有达到 15%。上述试验结果表明,系统总的调度时间随着任务数量的增加而大致呈线性增长趋势,该系统能够同时处理 200 个动作为“restart”的 VNF 调度,或 800 个动作为“start”或“stop”的 VNF 调度,而不影响响应的及时性,并且这些调度操作对系统资源的影响较小。说明基于普通宿主服务器,原型系统能够满足完成处理大量并发任务调度的需求。

4 结束语

当前,通过 SFC 定制 NFV 网络服务,仅能定义 VNF 次序,并且在 SFC 部署完成之后,不能再调整相应的 VNF 行为。文中研究了 SFC 定制 NFV 网络服务时所面临的两个问题,并给出了解决方案。首先,提出了 ESFC 的概念,给出了一种基于 XML 语言的细粒度描述 NFV 网络服务的表示方法;其次,设计并实现了灵活控制 ESFC 的系统。原型系统的试验结果表明,该系统通过解析控制描述文件,能够灵活、精细地控制 NFV 网络服务。并且该系统基于 NFV 技术实现,所得的试验结果真实可靠。接下来,将进一步优化控制器设计,使所提架构与机制能够更好地满足多种 NFV

网络服务的需求,以及研究该系统如何应用到真实网络中。

参考文献:

- [1] GHASEM M, LI Z Q. Optimal network function virtualization and service function chaining: a survey [J]. Chinese Journal of Electronics, 2018, 27(4): 704–717.
- [2] MEDHAT A M, TALEB T, ELMANGOUSH A, et al. Service function chaining in next generation networks: state of the art and research challenges [J]. IEEE Communications Magazine, 2017, 55(2): 216–223.
- [3] BHAMARE D, JAIN R, SAMAKA M, et al. A survey on service function chaining [J]. Journal of Network and Computer Applications, 2016, 75: 138–155.
- [4] 熊 钢, 王科人, 韩杰思. 网络服务功能链技术概述 [J]. 电信技术研究, 2017(1): 28–38.
- [5] 李源灏, 毛玉明, 韦云凯. 网络服务功能链技术研究建模 [J]. 计算机应用研究, 2019, 36(9): 2735–2738.
- [6] 刘益岑, 卢 昱, 陈兴凯, 等. 动态服务功能链(SFC)技术研究进展 [J]. 计算机工程与应用, 2018, 54(1): 32–41.
- [7] RFC 7665. Service function chaining (SFC) architecture [S]. [s. l.]: IETF, 2015.
- [8] BARI F, CHOWDHURY S R, AHMED R, et al. Orchestrating virtualized network functions [J]. IEEE Transactions on Network & Service Management, 2016, 13(4): 725–739.
- [9] BECK M T, BOTERO J F. Scalable and coordinated allocation of service function chains [J]. Computer Communications, 2017, 102(9): 78–88.
- [10] 李 丹, 兰巨龙, 王 鹏, 等. 基于最优加权图匹配的服务功能链部署方法 [J]. 通信学报, 2019, 40(3): 10–18.
- [11] OLJIRA D B, BRUNSTROM A, TAHERI J, et al. Analysis of network latency in virtualized environments [C]//2016 IEEE global communications conference (GLOBECOM). Washington, DC: IEEE, 2016: 1–6.
- [12] 陈 卓, 冯 钢, 刘 蓓, 等. 运营商网络中面向时延优化的服务功能链迁移重配置策略 [J]. 电子学报, 2018, 46(9): 2229–2237.
- [13] FAN J Y, GAO X J, YE Z L, et al. GREP: guaranteeing reliability with enhanced protection in NFV [C]//ACM SIGCOMM workshop on hot topics in middleboxes & network function virtualization. New York: ACM, 2015: 13–18.
- [14] 兰巨龙, 金子晋, 孙鹏浩, 等. 基于可靠性的服务功能链构建算法 [J]. 通信学报, 2019, 40(1): 64–70.
- [15] 韩 青, 兰巨龙, 江逸茗. 一种保证 NFV 可靠性的最优备份拓扑生成方法 [J]. 计算机应用研究, 2017, 34(11): 3384–3387.
- [16] SUN W H, XU Y, LIU F. The role of XML in service level agreements management [C]//International conference on services systems & services management. Chongqing, China: IEEE, 2005: 1118–1120.
- [17] JAKMA P, LAMPARTER D. Introduction to the quagga routing suite [J]. IEEE Network, 2014, 28(2): 42–48.
- [18] MICHALSKI M, KAMIL C, POLAK M. The system for large networks emulation with OSPF/BGP routers based on LXC [C]//IEEE 16th international conference on high performance switching and routing - HPSR 2015. Budapest: IEEE, 2015: 1–4.
- [19] 何翔慧. DDoS 典型工具分析—TFN2k [J]. 才智, 2009(23): 74.
- [20] 钱红燕, 薛 昊, 陈 鸣. UDM: 基于 NFV 的防止 DDoS 攻击 SDN 控制器的机制 [J]. 通信学报, 2019, 40(3): 116–124.