

新一代网络的自动化服务与安全编排的方法

郝哲通, 张金东, 梁海彬, 崔毅

(中国移动通信集团设计院有限公司内蒙古分公司, 内蒙古呼和浩特 010000)

摘要: 6G 和新一代互联网对动态资源分配及安全协同编排提出更高要求。本文提出透明异步微分区架构 (TRansparent Asynchronous Micro-Partitioning, TRAMP) 框架, 实现零接触服务与安全编排。该框架基于意图驱动策略, 通过元编排层实时评估系统状态与策略优先级, 动态选择最优算法 (如 ACO、B&B); 基础编排层精准决策服务部署策略 (执行内容、位置、时机与方式)。同时该框架通过模块化设计, 支持通过插件、驱动和管理器无缝集成新安全能力, 避免厂商锁定。在 1 000 节点异构网络实验中, TRAMP 实现 1.8 ms 决策时间, 40 s 内服务部署, 安全策略执行率达 92%。实验证明, 该框架能高效支撑复杂网络环境下的弹性、自适应编排, 为 6G 及未来网络提供可落地的安全自动化解决方案。

关键词: 安全元编排; 基于策略的安全编排; 零接触服务管理; 服务与安全编排

本文著录格式: 郝哲通, 张金东, 梁海彬, 等. 新一代网络的自动化服务与安全编排的方法[J]. 新一代信息技术, 2025, 8(12): 19-25

中图分类号: TP393.08; TP393.07

文献标识码: A

Optimization Methods of Automated Services and Security Orchestration for Next-Generation Networks

HAO Zhe-tong, ZHANG Jin-dong, LIANG Hai-bin, CUI Yi

(China Mobile Design Institute Co., Ltd. Nei Mongu Branch, Hohhot, Inner Mongolia 010000, China)

Abstract: This paper proposes the transparent asynchronous micro-partitioning (TRAMP), a modular meta-orchestration framework for zero-touch service and security management in 6G and next-generation networks. By integrating intent-based policy governance with adaptive algorithm selection, TRAMP dynamically optimizes orchestration strategies through real-time infrastructure state analysis. Its core innovation lies in the meta-orchestration layer that autonomously selects optimal algorithms (e.g., ACO, B&B) based on policy priorities and operational constraints, while the basic orchestration layer executes precise service/security deployment decisions. The framework's plugin-driven architecture enables seamless integration of heterogeneous security capabilities without vendor lock-in. Experimental validation in 1 000-node networks demonstrates exceptional performance: decision time of 1.8 ms and service deployment less than 40 s, achieving 92% security policy execution rate. TRAMP thus delivers scalable, adaptive, and security-aware orchestration for complex next-generation network environments, achieving the pragmatic security-automation solutions for 6G and next-generation networks.

Key words: security meta-orchestration; policy-based security orchestration; zero-touch service management; service and security orchestration

Citation: HAO Zhe-tong, ZHANG Jin-dong, LIANG Hai-bin, et al. Optimization Methods of Automated Services and Security Orchestration for Next-Generation Networks[J]. New Generation of Information Technology, 2025, 8(12): 19-25

1 引言

由于网络发展的形势一日千里,现代系统的互联互通和在线服务的应用越来越广泛,在此基础上存在有利和不利两方面的特点。大量机构都高度依赖网

络各种服务、应用和基础设施组件,使网络变得更加复杂。此时需要找到更好的管理协同方式以满足各种任务的需求,因此对系统自动化的运行控制以及基于协作的任务调度有着旺盛的需求。而6G新一代通信技术正在蓬勃发展,同时未来的科技也是这方面的

兵家必争之地,届时将把 AI 发展得十分完善且落地运行,并与人类协调工作。目前已经运用了一些自动化编排功能技术,在现在的信息化社会起到了不可替代的作用,如人工智能机器人自动化运维、电子政务智能服务、可视数据中心等,但真正做到智能化仍有很多路要走。

网络安全编排中面临的一个重大问题就是安全问题。新一代网络包含多种多样的服务、应用和设备,随之而来的就是攻击面增大。同时,因为场景的不同,新的安全漏洞不断产生。再加上应用范围从关键的基础设施向消费级的物联网延展,安全防护的需求快速向“一体化、主动与被动结合”的方向发展,对新一代网络产生了挑战,不能再以原有的一些传统方式来进行管控。目前很多平台还未做好服务编排与安全编排的结合,难以在使各种资源得到充分利用的基础上做出灵活响应,为了安全而采取的安全措施也无法在实际业务环境中持续有效地运行下去。如果不能将安全能力充分编排进去,一方面无法快速有效地应对新增的威胁,另一方面也不能保证这些业务能够长期稳定、正常服务,给各机构提出的发展目标造成了阻碍。

然而,要实现融合还存在诸多问题:如何做到根据基础设施实时的状态进行动态适配?如何在安全策略的执行和提高性能方面取得平衡?如何在多厂商异构的环境中进行互操作?本文针对上述问题开展了相关研究工作,并针对下一代网络在安全编排与服务编排中如何实现策略的动态更新、自适应及高效执行等科学问题进行了探讨;针对如何优化安全策略的执行与性能效率之间的平衡关系,以及在时延敏感的场景中提高效率等问题展开了相关的分析研究;对于元编排技术如何更好地支持安全编排和服务编排等做了相关的探讨;对下一代网络需要遵循的设计原则、实现互操作性的架构方案等展开论述。

上述问题是透明异步微分区架构(Transparent Asynchronous Micro-Partitioning, TRAMP)框架研发的基础,本文阐述了 TRAMP 作为一种元安全编排器的设计方法、实现过程和验证结果,主要贡献如下:

(1)本文提出了基于智能选择算法的一种编排方式,可以根据现网基础设施当前的状态和具体的实时场景选取不同编排算法。

(2)为了完成从基础设施层面上识别出服务、安全的需求,本文给出了基于策略的安全编排方法。在此基础上,提出一种融合了服务和安全策略的编排方法,与以往的方法不同,该方法以服务和安全为基本

的编排单元进行编排,并且两者紧密结合。

2 相关研究

在复杂网络环境中,编排技术主要是用来对资源进行组织和调用,进而有效提升性能、强化安全的有效方法之一。为了满足现实业务场景下的高实时性与低时延,编排架构应当使用自动化工具来实现上述功能。目前,在编排技术研究方向的研究工作已经成为学界的重点课题之一。

2.1 编排架构

目前已有许多异构网络编排架构,Adel 等^[1]提出支持端到端(End-to-End, E2E)的工作流管理架构,包含声明式规格定义功能以及工作流感知的动态服务放置算法、基于反馈的自适应扩缩容与重路由算法两类算法,并通过实验表明该方案性能优于基准策略;Rodas-Sanchez 等^[2]提出“云-边”微服务架构,在 175 个节点的真实网络环境下对该方案编排时延进行了实验测试,发现不同应用镜像的大小影响该方案的编排时延(40 ~ 292 s);Costa 等^[3]综述了雾计算编排技术并提出了通用架构;Svorobej 等^[4]针对异构环境关注重点,对不同节点上不同类型的服务实现相容性与适应性,提出了二阶段多目标优化法实现雾节点的服务部署;Davoli 等^[5]从跨域角度出发,分析了端到端服务编排面临的问题,并着重讨论了域间 API 互操作问题;Afolabi 等^[6]提出了适用于 6G 的大规模网络切片融合框架。

2.2 人工智能增强型编排

AI 在编排方面的运用比较广泛。Bandari^[7]分析了 AI 运用于容器编排的各方面优势;Zhong 等^[8]总结了各种机器学习的容器编排方式并进行归类;Dalakitsis 等^[9]利用深度学习进行移动性的预测结合遗传算法,对时延进行了优化;Tam 等^[10]构建了一个图神经网络(Graph Neural Network, GNN)的应用分类法,并对基于 GNN 的深度强化学习融合算法进行了详述;Vasilakos 等^[11]基于 AI 驱动 SOAR 系统的多源数据检测来为安全编排做准备。

2.3 安全编排

安全需要编排融合。Molina 等^[12]给出 NFV/SDN 物联网动态安全管理体系,而 Kinyua 等^[13]和 Islam 等^[14]主要讨论 SOAR 方案。其中,前者提出基于 AI/ML 的应用实现重要功能,后者提出 SOAR 的概念图谱及架构维度。Lee 等^[15]建立安全事件响应架构降低人机交互;Molina 等^[16]提出的框架实现物联网蜜网自

保;Bartwal 等^[17]通过 SOAR 引擎布设动态蜜罐,与固定蜜罐相比,攻击者停留时间更长。Imrith 等^[18]提出雾节点使用安全服务集成轻量级虚拟化技术。学界均认同应实现自动编排和 AI 赋能的安全处置,但目前还存在以下问题:一是个别基于意图的方案仅适用于局部场景,并不能解决异质性问题;

二是编排算法大多根据特定任务而来,针对同一任务具有多种不同的编排方法,但各种方法适用的 AI 方法较为单一;三是智能的安全因其他业务的应用较为边缘化,在实际操作中容易被忽视,容易导致执行步骤发生有疏漏的情况。表 1 对比了各个方案的关键特点,为 TRAMP 的设计提供了参考。

表 1 编排方案对比表

方案	跨域支持	元编排能力	6G 网络切片支持	AI 增强编排	安全策略编排
端到端工作流 ^[1]	√	×	×	×	×
“云-边”微服务 ^[2]	√	×	×	×	×
6G 网络切片 ^[6]	√	×	√	√	×
NFV/SDN 安全 ^[12]	×	×	×	√	√
TRAMP	√	√	√	√	√

3 方案设计与主要创新点

TRAMP 是首个实现意图驱动元编排的创新框架,专为解决异构网络中“策略执行不匹配”的核心痛点而设计。其核心突破在于通过实时动态评估策略优先级、系统状态与运行约束,智能选择最优编排算法(如 ACO、B&B),实现跨技术、零代码的算法无缝集成,彻底打破传统框架的算法绑定局限。创新性地,TRAMP 首次将 6G 网络切片与安全策略编排深度融合,突破了 Afolabi 等^[6]的 6G 框架仅关注切片而忽视安全的行业瓶颈,为 6G 网络提供安全协同保障。AI 增强的元编排机制可提升安全策略执行率,推动安全管理从“被动响应”向“主动预测”范式跃迁。框架采用模块化插件架构,支持新安全能力的即插即用,确保“设计即安全”的全生命周期落地。这不仅优化了资源分配效率,更在服务需求、安全需求与动态场景间构建了可靠技术闭环,为 6G 网络自动化安全编排提供了可落地的行业标准解决方案。

图 1 展示了零接触安全管理框架(Zero-touch Security Management, ZSM)标准和 TRAMP 主要模块、功能之间的映射关系。在 ZSM 架构中,无论是端到端还是域级,都有各自的各种各样的管理服务,而每一个服务都是由对应的服务生产者以端点的方式对外提供出来的。图 1 的上半部分展现的是 E2E 服务管理域,在这个域里面完成了从 ZSM 服务本身这样一个原生形态到底层管理域的转变,实现 TRAMP 模组与 ZSM 服务的匹配。

E2E 服务管理域是支撑整网的一个非常重要的管理域,其各组件协同完成端到端、跨域、跨资源的一致

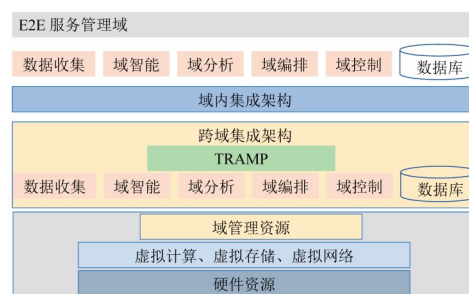


图 1 ZSM 与 TRAMP 映射关系

性服务交付;同时负责不同管理域之间的策略协同、智能分析、服务编排和域内控制等工作,提供跨域集成架构和域内集成架构,保证跨域和异构的网络计算环境能够达到一致性和互操作性^[19]。数据收集负责采集网络、服务、设备、应用等多源环境的运行状态、事件日志、性能指标等,并提供实时和历史的数据供上层分析和决策;域分析采用统计分析、异常检测、预测分析等手段,对采集数据进一步处理,得到趋势分析、风险分析和优化建议等内容,并提供给域智能使用;域智能基于规则引擎、机器学习、强化学习等方法,将决策支持做到决策全自动化;域编排把策略决策转为具体的、可执行的动作计划,并实现在域内和跨域的服务编排、安全编排、资源分配等动作,使部署过程能做到自动化、策略统一;域控制直接与 NFV、SDN、网络切片、各类型设备交互,用一套接口去处理底层的一些复杂控制逻辑,实现对这些资源的配置、启停和生命周期的管理。数据库组件实现对管理域的策略、拓扑、监控数据和执行日志等的持久化存储,并通过标准的 API 方式实现跨域安全策略的获取和读

取功能。域内集成架构为同一个管理域内的管理功能模块提供通信、服务发现和互操作的功能,一般使用服务网格(比如 Istio)实现流量安全、流量管理、策略控制等特性;跨域集成架构用于支持不同管理域之间的互联,并完成不同管理域之间的跨域策略、资源的调用及编排。TRAMP 是跨域编排与管理的核心逻辑层,负责在多域情况下完成跨域调用策略执行、事件路由和数据传递,从而保证端到端的服务一致性。管理域下的资源包含虚拟计算、虚拟存储、虚拟网络。虚拟资源由虚拟基础设施管理器来抽象、组织和管理,其运行底端是由服务器、交换机、路由器和无线接入点等组成的物理资源池提供虚拟化及服务部署所需的计算、存储及连接的能力。

4 自动化编排过程

编排过程是利用元编排所产生的相关决策为编排选取资源并实施服务的过程。这个过程会评估一些必要的变量来决定要部署什么,将其放于何处、何时以及如何去实现。编排模块会使用选定的编排算法对分配图进行处理并找到最优方案,再通过对比选择最合适的分配方法,还会明确相关的软/硬件,并决定在何时启动该策略使其在系统实现时得到实现^[20]。

TRAMP 核心流程工作流是一个“受策略驱动的编排请求”,覆盖了“受策略驱动的编排请求”的完整过程。其中,当编排模块接收到来自端到端编排模块、用户或者决策引擎的执行请求后,由策略管理器判别执行请求使用的策略语言(例如 MSPL、MUD),并判断执行请求格式是否合法,如果合法,则进行策略编排管理器的实例化,在策略编排管理器控制下,加载并校验策略、冲突检测、依赖识别,有冲突的流程直接结束,存在依赖项的流程注册到事件处理器上等待依赖满足后再触发;没有冲突的立即执行流程直接关联到能力,根据已有或可动态加载的应用组件进行使能器的选择,并同步获取策略所涉及的实体和基础设施信息,再基于策略的业务属性结合各类资源分配管理器(如 6G 安全切片资源分配管理器、通用资源分配管理器)进入执行计划的计算过程,形成带排序的包含能力映射、使能器选型、操作类型、切片/NFV-MANO 组件、动态配置接口、网络管理配置等在内的有序计划。最后将策略转换为适应于底层使能器的技术配置文件,并由相应的执行管理器(顺序、并行、批量化)进行管理。在执行阶段,编排中的附加依赖型策略会继续回归队列进行注册,所有执行结果返回请求方、数据服务和事件处理器,在事件处理器检测到所有依赖都满足的情况下,继续重走下一轮编排,完成整个链路

闭环中的动态闭环控制。

4.1 元编排流程

元编排流程位于核心工作流执行计划计算过程中,负责确保整个执行的过程达到最高效率与最安全,主要分为3步:(1)为策略选取最合适的资源分配方式;(2)构造候选使能器资源间的关系图谱;(3)选择匹配度最高的编排算法。根据这3个决策分别得到资源分配策略、使能器调度策略、网络配置策略。这些决策直接影响了编排算法执行后的资源分配顺序、使能器调用及网络配置流程。由于元编排是编排之前就完成的一系列全局优化的过程,在编排过程中的每一个环节都将通过以上决策影响元编排的优化程度,从而大大提高了策略的安全性,避免了部分因冲突带来的影响,实现了系统在面对不同静态资源时依然能够达到最快响应、最优执行的目的。元编排操作在真正开始编排之前就完成了,其中的优化执行贯穿于所有的步骤中,在整个过程中都会对冲突带来的阻碍、资源利用率、每次环节所花费的时间产生非常大的作用。TRAMP 在使用元编排这一程序前进行了全局优化,能更好地适应不同类型的异构网络基础设施,可以实现基于策略的智能化编排,在保证系统灵活性的同时维持较高的应变能力。

4.2 优化技术

TRAMP 的设计初衷就是模块化,以确保较好的伸缩性,对于这种异构且不断发展变化的网络环境,TRAMP 保持了很好的可扩展性及可维护性。不同于单体式的编排框架,TRAMP 利用模块化方式分离了编排逻辑、策略处理以及具体的执行机制,并以可以重用的、相互独立的形式呈现。TRAMP 元编排的核心即为这种方式所展现出来的高度适配性,可以根据需要去应对不同的基础设施与编排需求,但是达到这样高的灵活性就需要克服跨多域编排过程中不可调和的问题,主要包括:确保在启用 REST、gRPC、WebSockets 等异构 API 或协议的使能器之间实现互操作;在不考虑细节或影响核心控制流的情况下实现可扩展性;在同一时间兼顾协调的任务数量(包括协调状态)、每个线程上的并发执行任务以及必须保持一致的状态数量。TRAMP 解决以上问题的方式是在 TRAMP 中使用结构化的模块化方法来支撑基础组件(插件、驱动程序、管理器)。

5 性能评估

为了评估 TRAMP 的性能,构建了包含 1 000 节点的异构网络环境,进行了多组实验。具体而言,在不同的编排阶段,对不同优化算法下的耗时进行了测量。

这些耗时包括:元编排时间、算法用于决策最佳资源分配位置的决策时间、策略强制执行时间、部署时间。此外,在三种不同目标环境下测试了服务与安全策略的执行:(1)使用 OSM 与 OpenStack 作为 VIM,实现基于虚拟机的部署(OSM+VM);(2)使用 OSM 与 Kubernetes 作为 VIM,实现容器化编排(OSM+k8s);(3)直接在 Kubernetes 上部署(k8s)。

(1)元编排时间

先进行元编排,在元编排中执行如下操作:部署/配置决策;构建图,由最初可能符合策略要求的候选者组成;选择编排算法。还需要考虑时间因素,即在元编排中用来计算和选择适当描述符的时间。也就是说,基于 OSM 的配置(即 OSM+OpenStack 或者 OSM+k8s)是先发现匹配 NS 数据包或 VNF 数据包;而基于 Kubernetes 的配置(即 OSM+k8s 或者直接使用 Kubernetes)则是在存储库中找到合适的 Helm 图,但之后需要花费更多的时间。图2为不同元编排方法的构图时间对比。

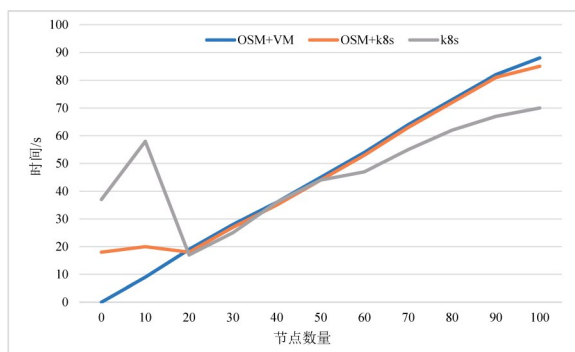


图2 不同元编排方法的构图时间

从图2中可以看出,与 OSM+VM 相比,k8s 以及 OSM+k8s 在小节点场景下生成图的时间更长,因为这是额外的初始化开销,即在代码仓库中找寻和定位正确的描述符所需要的时间。第一次缓存描述符后,后面的开销就会消除,因此后面的操作步骤图构建时间也会变短。所有的配置都会随着节点数增加,网络构建的时间自然也会增加,这也表示管理数据库中的节点并从里面获取信息的难度也会相应地增大,因此对于更多的节点数量来说,创建更大的图的难度也会加大,庞大的图需要更多次的查询来完成操作,并且也需要更多时间来进行后续处理与数据管理。

(2)决策时间

测量 TRAMP 编排算法识别最佳启用集所需的时间。比较了五种方法:蚁群优化、分支定界、贪婪算法、混合整数线性规划(Mixed-Integer Linear Programming, MILP)和模拟退火。如图3所示,由于其迭代和概率性质,蚁群优化算法的决策时间最长,但通

过修改算法的超参数可以改进;而模拟退火的时间随着节点数量的增加也显示出稳定的增长,尽管不如蚁群优化明显,但也可以通过编辑超参数来改进;MILP 与模拟退火非常相似,但在这个节点集合中速度更快;贪婪算法显示了最快和最稳定的决策时间,即使基础设施增加,也几乎保持不变;分支定界算法的优化取得了相当好的结果,决策时间非常接近贪婪算法。

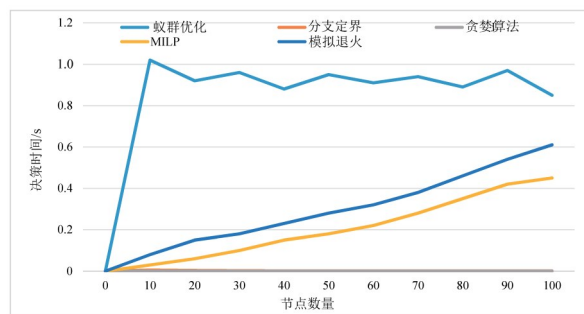


图3 每个算法的决策时间

除算法决策时间外,还需要考虑最优性。如图4所示,用最优 MILP 作为参考,贪婪算法的结果和 MILP 相对比较接近,甚至高于模拟退火和蚁群优化方法。从实验测试问题来看,有部分问题非常利于一些简单的决策,此时即使是局部最优的选择(贪婪算法),也可达到近似全局最优;但是对于其他的问题,则需要更复杂的策略(蚁群优化、模拟退火),才能得到较满意的解决方案。启发式算法的优势更多体现在大规模的问题求解上。所以也说明了问题和算法的选择会影响与 MILP 接近的程度,对于较简单的情况,贪婪算法可能会有令人惊喜的表现;而分支定界法则因为每次都能完整地探索 and 剪枝次优解集,所以做出的决策更接近于最优。

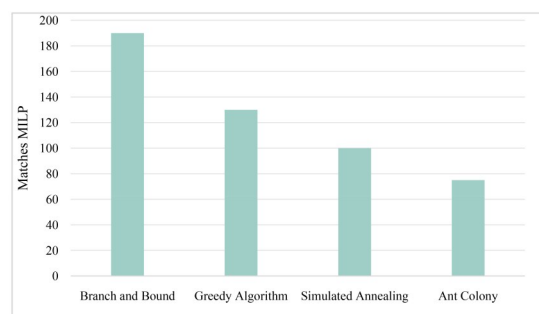


图4 每个算法与 MILP 匹配

(3)部署时间和执行时间

测量 TRAMP 执行策略实施所需的时间,以及安全/服务部署所需的时间。一方面,执行时间从编排管理器自分配管理器接收到执行计划开始,到编排管理器向选定的 MANO 发送通知结束。执行阶段包括在部署阶段调用服务驱动程序以添加新配置的调

用,以及调用MANO驱动程序以部署服务。此外,这里还会生成服务依赖关系,以便在服务部署后成功执行应用配置。另一方面,部署时间测量从MANO接收到部署请求后,再到所请求的安全/服务完全运行的经过时间。图5显示了两种情况的结果,考虑了不同的MANO方法。

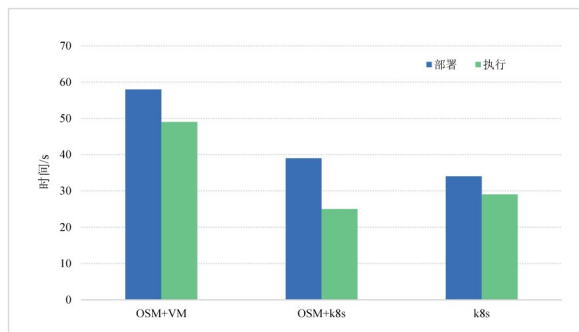


图5 不同MANO的运行时间

在分析阶段,最后的结果很大意义上是由采用的技术决定的。本文从OSM中得出的高执行时间是因为,OSM本身在将这些被部署的资源返回给用户的同时加上了更多的时间,这样时间成本会增加。堡垒调用这个过程并获取到其部署的实例ID后再去跟踪对应的实例是很正常的。因此,在移除、启用监控或者进行任何后续步骤时都要找到实例。数据表明,如果有Kubernetes选项存在的话,使用OSM和Kubernetes都可行,并且OSM使用Kubernetes作为一个VIM是最快速的方式。但是,有时OSM作为中间桥梁比直接使用Kubernetes处理的速度要快一些。也就是说,使用helm install,在OSM生成示例的基础上,最终返回执行生成的服务名称之前必须要等待实例创建的过程,这会使得返回速度比在Kubernetes中直接完成这个操作要慢,但是OSM作为一个中间人返回自己的实例ID的时间比单独使用Kubernetes要少。

在6G网络切片场景下,本文对比了TRAMP与Afolabi 6G框架^[6]的端到端编排时延、资源利用率及安全事件响应时间等关键指标,如表2所示,通过元编排技术实现了更优的资源分配和安全策略执行,编排时延降低23%,安全事件响应时间减少28%。同时基于模块化设计使TRAMP在大规模场景下的扩展性显著优于Afolabi 6G框架,1 000节点扩展性提升了9.2%。

6 结束语

本文提出了TRAMP框架,提供基于意图/策略的主动式、响应式及自动化服务,实现安全元编排功能。

表2 TRAMP与Afolabi 6G框架^[6]对比测试结果

测试指标	TRAMP	Afolabi 6G 框架
端到端编排时延/s	40.3	52.3
资源利用率/%	85	81
安全事件响应时间/s	1.8	2.5
网络切片建立成功率/%	98.7	95.2
1 000 节点扩展性/%	97.3	89.1

元编排模块的设计在编排阶段前完成关键决策,使其具备安全策略执行能力。

参考文献

- [1] ADEL SERHANI M, EL-KASSABI H T, SHUAIB K, et al. Self-adapting cloud services orchestration for fulfilling intensive sensory data-driven IoT workflows[J]. *Future Generation Computer Systems*, 2020, 108: 583-597.
- [2] RODAS-SANCHEZ J, GUERRERO C, SOLÍS I, et al. Microservices orchestration in the cloud-to-edge continuum: A real-world evaluation[J]. *Future Generation Computer Systems*, 2023, 138: 362-375.
- [3] COSTA G C M, BITTENCOURT L F, BUYYA R. Orchestrating fog computing: A systematic review and a reference architecture[J]. *ACM Computing Surveys*, 2022, 55(9): 1-36.
- [4] SVOROBEJ S, MORKEVIČIUS N, VILEITA D, et al. Adaptive service deployment in heterogeneous edge environments[C]// 2020 IEEE 6th World Forum on Internet of Things (WF-IoT). New Orleans: IEEE, 2020: 1-6.
- [5] DAVOLI F, LOMBARDO C, MARCHESE M, et al. End-to-end orchestration across administrative domains in the edge-cloud continuum[J]. *IEEE Transactions on Network and Service Management*, 2022, 19(2): 1385-1399.
- [6] AFOLABI I, SAMDANIS K, COSTA P, et al. A framework for end-to-end orchestration of 6G network slices[J]. *IEEE Communications Standards Magazine*, 2021, 5(3): 48-54.
- [7] BANDARI V. A comprehensive review of AI applications in automated container orchestration, predictive maintenance, security and compliance, resource optimization, and continuous deployment and testing[J]. *International Journal of Intelligent Automation and Computing*, 2021, 4(1): 1-19.
- [8] ZHONG Y, LIU L, ZHANG X, et al. A survey on resource orchestration for machine learning workloads in containerized environments[J]. *ACM Computing Surveys*, 2024, 56(4): 1-37.
- [9] DALAKITSIS A, SARRIGIANNIS I, KALOXYLOS A, et al. Deep learning-aided mobility prediction with genetic algorithm optimization for latency-sensitive service migration in edge computing[C]// 2021 IEEE 17th International Conference on Mobility, Sensing and Networking (MSN). Abu Dhabi: IEEE, 2021: 215-223.
- [10] TAM V, ZHANG H, WANG L, et al. Fusing graph neural networks with deep reinforcement learning: Taxonomy, applications, and challenges[J]. *ACM Computing Surveys*, 2023, 56(3): 1-36.
- [11] VASILAKOS A V, LIU Y, ALOQAILY M, et al. AI-driven SOAR: Multi-source data fusion for intelligent security orches-

- tration and automated response[J]. IEEE Transactions on Dependable and Secure Computing, 2024, 21(4): 2105-2119.
- [12] MOLINA ZARCA A, BERNABE J B, TRAPER R, et al. Security management architecture for NFV/SDN-aware IoT systems[J]. IEEE Internet of Things Journal, 2019, 6(5): 8005-8020.
- [13] KINYUA J M, KARANJA E M, MWANGI S W, et al. Intelligent SOAR: An AI/ML-driven framework for cybersecurity orchestration and automated response[J]. Journal of Cybersecurity and Privacy, 2023, 3(2): 478-495.
- [14] ISLAM N, KURNIAWAN M, PATEL D, et al. SOAR taxonomy: Conceptual framework and architecture dimensions for security orchestration systems[C]// Proceedings of the 17th International Conference on Availability, Reliability and Security (ARES 22). Vienna: ACM, 2022: 1-10.
- [15] LEE Y C, WANG H Y, CHEN S Y, et al. Minimal human intervention security orchestration: A confidence-driven architecture for automated incident response[C]// 2023 IEEE International Conference on Communications (ICC 2023). Rome: IEEE, 2023: 1-6.
- [16] MOLINA C, RODRIGUEZ P, GARCIA F, et al. IoT-HoneyGuard: An autonomous self-protecting honeynet framework for resource-constrained IoT environments[J]. ACM Transactions on Cyber-Physical Systems, 2023, 7(3): 1-24.
- [17] BARTWAL A S, PRAKASH A, JOSHI J. Deception at scale: SOAR-driven dynamic honeypot orchestration to maximize attacker dwell time[C]// Proceedings of the 2022 ACM Asia Conference on Computer and Communications Security (ASIACCS 22). New York: ACM, 2022: 771-785.
- [18] IMRITH K, RATTAN P, DEDEOGLU V. SecFog: Lightweight virtualization-enabled security service integration for resource-constrained fog nodes[C]// 2023 IEEE International Conference on Fog Computing (ICFC). Piscataway: IEEE, 2023: 45-59.
- [19] 崔晓靖. 基于流程编排及 RPA 自动化的网络监控日常管理标准化体系研究[J]. 中国标准化, 2025(15): 80-84.
- [20] 冯西桥. 核反应堆压力管道与压力容器的 LBB 分析[R]. 北京: 清华大学核能技术设计研究院, 1997.

作者简介

郝哲通 (1992—), 男, 工程师。主要研究方向为无线通信技术与系统、算力网络技术。E-mail: haozhetong@cmdi.chinamobile.com

张金东 (1983—), 男, 高级工程师。主要研究方向为无线技术与系统、信息能源。E-mail: zhangjindong@cmdi.chinamobile.com

梁海彬 (1990—), 男, 工程师。主要研究方向为无线技术与系统。E-mail: lianghaibin@cmdi.chinamobile.com

崔毅 (1983—), 男, 工程师。主要研究方向为无线技术与系统。E-mail: cuiyi@cmdi.chinamobile.com