

# 聚合型时序逻辑驱动的复合服务联动定量监测\*

周长兵<sup>1,2</sup>, 赵登<sup>1,2</sup>, 张文博<sup>3</sup>, 孙骁<sup>4</sup>, 薛霄<sup>5</sup>

<sup>1</sup>(中国地质大学(北京)信息工程学院, 北京 100083)

<sup>2</sup>(中国地质大学(北京)深时数字地球前沿科学中心, 北京 100083)

<sup>3</sup>(中国科学院软件研究所, 北京 100190)

<sup>4</sup>(清华大学自动化系, 北京 100084)

<sup>5</sup>(天津大学智能与计算学部, 天津 300072)

通信作者: 周长兵, E-mail: zbzhou@cugb.edu.cn



**摘要:**近年来,面向服务的物联网体系架构受到学术界和工业界的广泛关注.通过把物联网资源虚拟化为智能物联网服务,动态互联并融合协同这些资源受限且动态演进的物联网服务,从而构建物联网应用,已成为一种被普遍认可且灵活有效的机制.面向边缘设备资源稀缺且动态时变,物联网服务在其执行过程中可能发生QoS变化或资源失配,致使物联网应用难以继续或可能诱使故障发生.因此,实现物联网服务的实时定量监测,已成为保障物联网应用鲁棒性和系统健壮性的关键.现有相关研究工作提出了不同监测机制,但其在形式化解释上仍存不足,表现为较强的领域相关性和经验主观性.基于形式化方法,例如信号时序逻辑,可将物联网服务运行态实时定量监测问题转换为时序逻辑任务.然而,现有信号时序逻辑存在信号不可辨性、方法不鲁棒性以及场景不适用性问题.并且,在物联网服务以复合服务形式监测时,现有工作存在整体性、联动性、动态性考虑不足的问题.为解决上述问题,提出一种聚合型信号时序逻辑,以实现单个服务、服务之间以及复合服务上不同QoS约束和时间约束的实时联动定量监测.所提方法扩展基于正负偏置黎曼和的累加型时间算子,对整个时间域内所有子公式进行鲁棒性度量,实现时序鲁棒、信号可辨、动态适用的物联网服务运行态实时定量监测;并扩展基于约束类型与组合结构的聚合算子,以及随动态环境可变的动态变量,实现复合服务在动态环境中的联动定量监测.由此,物联网服务及其复合服务的多维度多约束被转换为逻辑公式,并在运行时以定性和定量化的满意度进行形式化解释.实验结果表明所提方法具有更好的监测表达能力、场景适用性和结果鲁棒性.

**关键词:**物联网服务;复合服务;定量监测;实时监测;信号时序逻辑

**中图法分类号:** TP311

中文引用格式: 周长兵, 赵登, 张文博, 孙骁, 薛霄. 聚合型时序逻辑驱动的复合服务联动定量监测. 软件学报. <http://www.jos.org.cn/1000-9825/7082.htm>

英文引用格式: Zhou ZB, Zhao D, Zhang WB, Sun X, Xue X. Compositional Signal Temporal Logic for Runtime Quantitative Monitoring of Composite Services. Ruan Jian Xue Bao/Journal of Software (in Chinese). <http://www.jos.org.cn/1000-9825/7082.htm>

## Compositional Signal Temporal Logic for Runtime Quantitative Monitoring of Composite Services

ZHOU Zhang-Bing<sup>1,2</sup>, ZHAO Deng<sup>1,2</sup>, ZHANG Wen-Bo<sup>3</sup>, SUN Xiao<sup>4</sup>, XUE Xiao<sup>5</sup>

<sup>1</sup>(School of Information Engineering, China University of Geosciences, Beijing 100083, China)

<sup>2</sup>(Frontiers Science Center for Deep-time Digital Earth, China University of Geosciences, Beijing 100083, China)

\* 基金项目: 中央高校基本科研业务费深时数字地球前沿科学中心“深时数字地球”中央高校科技领军人才团队项目(2652023001); 国家自然科学基金(62372420, 42050103); 地质调查专项“地球科学文献知识服务与决策支撑二级项目”(DD20230139)

收稿时间: 2023-05-08; 修改时间: 2023-07-21; 采用时间: 2023-10-23; jos 在线出版时间: 2024-03-13

<sup>3</sup>(Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

<sup>4</sup>(Department of Automation, Tsinghua University, Beijing 100084, China)

<sup>5</sup>(College of Intelligence and Computing, Tianjin University, Tianjin 300072, China)

**Abstract:** In recent years, service-oriented IoT architectures have received a lot of attention from academia and industry. By encapsulating IoT resources into intelligent IoT services, interconnecting and collaborating these resource-constrained and capacity-evolving IoT services to facilitate IoT applications has become a widely adopted and flexible mechanism. Upon capacity-fluctuating and resource-varying edge devices, IoT services may experience QoS degradations or resource mismatches during their execution, making it difficult for IoT applications to continue and possibly inducing failures. Therefore, quantitative monitoring of IoT services at runtime has become the key to guaranteeing the robustness of IoT applications. Different monitoring mechanisms have been proposed in recent literature, but they are inadequate in formal interpretation with strong domain relevance and empirical subjectivity. Based on formal methods, such as signal temporal logic (STL), the problem of IoT service monitoring can be formulated as a temporal logic task to achieve runtime quantitative monitoring. However, STL and its extensions suffer from issues of non-differentiability, loss of soundness, and inapplicability in dynamic environments. Moreover, existing works are inadequate for the monitoring of composite services, with a lack of integrity, linkage, and dynamics. To solve these problems, this study proposes a compositional signal temporal logic (CSTL) to achieve quantitative monitoring of different QoS constraints and time constraints upon intra-, inter-, and composite services. Specifically, CSTL extends an accumulative operator based on positively and negatively biased Riemann sums to emphasize the robust satisfaction of all sub-formulae over their entire time domains and to evaluate qualitative and quantitative constraint satisfaction for IoT service monitoring. Besides, CSTL extends a compositional operator based on constraint types and composite structures, as well as dynamic variables that can vary with the dynamic environment, to effectively monitor QoS variations and temporal violations of composite services. As a result, temporal and QoS constraints upon intra-, inter-, and composite services, can be specified by CSTL formulae, and formally interpreted with qualitative and quantitative satisfaction at runtime. Extensive evaluations show that the proposed CSTL performs better than baseline techniques in terms of expressiveness, applicability, and robustness.

**Key words:** IoT service; composite service; quantitative monitoring; runtime monitoring; signal temporal logic (STL)

随着边缘计算的出现及广泛运用,成千上万的物联网设备正在被部署于边缘网络中,以支持边缘原生应用<sup>[1]</sup>。由于单个物联网设备的能力受限性和资源稀缺性,通常由多个物联网设备协作来共同完成复杂应用的应答实现。在面向服务的物联网架构体系中,物联网设备的功能可被封装并抽象成为物联网服务,物联网设备之间的协作可通过物联网服务的组合适配完成<sup>[2]</sup>。传统的组合技术已在近些年来被广泛研究,通常这些方法基于物联网服务的功能性匹配度和非功能性最优化,可实现边缘网络中物联网服务的有效组合适配,以应答用户需求和复杂应用<sup>[3,4]</sup>。然而,传统服务组合的一个隐含前提是网络性能指标在一段时间内保持稳定,物联网服务的功能和非功能性属性在其运行期间被假定为无变化<sup>[5]</sup>。考虑边缘设备上资源稀缺且通常难以得到有效补充,且随着应用的执行/完成和新应用的持续部署,边缘节点可用资源的状态呈现动态波动的特征,处于运行态的应用因为网络拓扑变化和资源波动,使得预配资源难以得到持续保障<sup>[6]</sup>。因此,在这种资源处于持续动态变化的边缘网络中,当前时刻可满足用户需求的物联网服务组合方案可能无法在接下来的时间内保持其 QoS 值或相关非功能性约束满足度不变,致使用户体验下降。由此,面对边缘设备资源稀缺且动态可变,及物联网应用复杂多变的情景,物联网服务组合形成复合服务后,在其执行过程中可能发生服务失配,实现复合服务的实时定量监测已成为保证物联网服务组合方案鲁棒性的关键。

目前已有相关研究工作聚焦物联网服务监测的鲁棒性和轻量级两个方面关键要素,提出了各种不同服务监测机制,以保障物联网服务应答相关应用的健壮性<sup>[7,8]</sup>。例如, Costa 等人<sup>[9]</sup>提出雾计算环境中的资源监测机制, Trihinas 等人<sup>[10]</sup>就成本感知的物联网自适应监测展开研究, Zhang 等人<sup>[11]</sup>则构建了一种移动边缘计算中移动性和依赖性感知的 QoS 监测方法。通常,这些方法将物联网服务监测问题规约为约束满足问题<sup>[12]</sup>,并通过将其简化为数学对象或适应度评估函数来构建相关定量监测机制<sup>[13]</sup>。尽管这些研究工作采用诸如适应度之类的数学函数,来合理地定量计算物联网服务实时约束满足度,但这些监测工作在形式化解释上仍存不足,表现为较强的领域相关性和经验主观性。为了解决这一问题,本文旨在以定性和定量的方式形式化解释物联网服务上所指定约束的实时满意度,从而在物联网服务执行过程中实现实时定量监测。

相较于其他时序逻辑, 信号时序逻辑 (signal temporal logic, STL) 不仅可规约定性化的约束, 还可对约束满足度进行定量分析, 即不仅可回答“是否、有无”, 还可回答在“多大程度上”满足或违背特定约束。例如, Zhou 等人<sup>[14]</sup>构建了一种知识与数据驱动下的上下文感知的 CPS 系统安全监测机制, 其中不安全的系统场景可被信号时序逻辑形式化表达; Ma 等人<sup>[15]</sup>提出了一种带有空间聚合和空间计数逻辑算子的信号时序逻辑, 实现对智慧城市中交通、空气污染、人流分布等多种应用的定量监测; Haris 等人<sup>[16]</sup>就物联网服务部署问题, 通过信号时序逻辑规范指定其 QoS 约束, 并实现 QoS 管理器的持续监测; Antonyshyn 等人<sup>[17]</sup>在对多个移动机器人相关任务及路线规划问题的调研总结中, 发现相当大一部分工作利用信号时序逻辑及其拓展版本对其任务进行形式化规约与验证。受这些工作的启发, 为形式化表征物联网服务在边缘网络中发生失配的程度, 本文拟基于信号时序逻辑实时监测物联网服务在一个时间段内定性与定量的约束满足度。

经研究可发现传统信号时序逻辑及其最新拓展研究工作仍存在不足, 难以满足实际物联网服务多维度、多约束、多并发的实时定量监测需求, 具体可归纳为如下 3 个方面: ① 传统的信号时序逻辑存在信号值迥异、满足度可能无差异的不可分辨性问题。这是由于信号时序逻辑本身仅依赖于使用求极值 (包括最大值和最小值) 函数, 获得一段时间内信号值的时序满足度。最新拓展方法已注意到该缺陷, 针对该缺陷, 现有工作提出均值算子<sup>[18]</sup>, 或构建平滑算子<sup>[19]</sup>, 对信号时序逻辑进行有效增强。② 然而, 这些工作存在不满足约束的信号值被满足约束的信号值覆盖所导致的不鲁棒性问题, 如均值为正值并不能代表所有信号值均为正值, 除非其大于特定预定义阈值。针对该不鲁棒性缺陷, 最新的研究工作对满足约束的信号值和不满足部分分别定义其定量化语义, 例如, 对满足和不满足部分信号值做不同均值计算<sup>[20]</sup>, 或分别构建满足和不满足两部分信号值的平滑算子<sup>[21]</sup>。③ 需要强调的是, 这些最新研究均基于固定采样率的假定, 难以适用于动态变化边缘场景下信号采样速率难以预先设定的问题。例如, 在物联网服务应答应用需求中可能由于在其执行过程中出现的潜在不确定因素或环境干扰, 使得其信号值采样间隔发生变化。因此, 现有方法由于其自身采样率固定的限定, 难以直接运用于边缘场景中物联网服务的实时监测。为了解决上述问题, 本文将面向非均匀信号采样速率的边缘场景, 对信号时序逻辑拓展一种基于黎曼和的累加型时间算子, 实现时序鲁棒、信号可辨、动态适用的物联网服务运行态实时定量监测机制。这种改进可有效解决传统信号时序逻辑本身的信号不可辨问题, 并采用正负偏置的黎曼和累加型算子保证其鲁棒性。此外, 由于其对于不确定性的或非周期性信号的有效处理, 该方法在物联网服务监测方面表现出更强的适用性。

尽管目前已有较多技术实现对物联网服务的监测, 经研究可发现现有工作缺乏对于物联网服务整体性、联动性、动态性的考虑: ① 现有技术主要聚焦于单个或多个物联网服务的监测, 缺乏对于物联网服务组合形成的复合服务的整体监测<sup>[22]</sup>。但实际上应考虑复合服务的监测, 因为复合服务的满足, 不等同于复合服务中每一个物联网服务必须满足某些硬性约束; 而单个物联网服务的失配, 可能不代表复合服务的失配; 同时, 用户可能对于单一物联网服务有一定容错度, 复合服务整体满足用户要求即可。② 另一方面, 一个物联网服务的失配可能导致相关有依赖关系的物联网服务失配, 应实现物联网服务的联动监测。在实际的物联网服务网络中, 服务之间可能存在时间上或 QoS 方面的依赖关系<sup>[23]</sup>; 例如, 后置服务可能被限制在前置服务执行结束后 10 min 内开始执行; 并且, 复合服务中前置服务的执行情况可能影响后置服务的执行约束监测<sup>[24]</sup>。③ 此外, 现有监测方案大多基于静态或预设的规则、阈值和参数等, 未考虑随实际动态场景进行自适应调整, 而难以运用于动态变化边缘网络下物联网服务监测问题<sup>[25]</sup>。物联网服务监测的动态性, 表现在其监测可能受其他服务影响, 并且在物联网服务执行过程中充满潜在的不确定性或干扰<sup>[26,27]</sup>。为了解决上述缺陷, 本文面向复合服务监测场景, 对信号时序逻辑扩展一种聚合算子和动态变量, 实现对复合服务的联动定量监测。这种扩展可有效提升信号时序逻辑对于复合服务监测表达能力, 和在边缘动态变化场景下的适用性。

综上所述, 本文针对物联网服务及其复合服务在其执行过程中对于 QoS 变化的实时联动定量监测需求, 以及现有监测方法存在的信号不可辨性、方法不鲁棒性、场景不适用性问题, 和面向复合服务整体性、联动性和动态性考虑不足的问题, 提出一种聚合型信号时序逻辑, 在传统信号时序逻辑基础上, 拓展了累加型时间算子、聚合算子和动态变量, 以实现物联网服务及其复合服务运行态实时联动定量监测。单个服务、服务之间以及复合服务上多维度多约束被转换为逻辑公式, 并在运行时以定性和定量化的满意度进行形式化解释。实验结果表明本文提出

的方法拓展了信号时序逻辑的监测表达能力和场景适用性,并有效提升了物联网服务及其复合服务实时联动监测的结果鲁棒性。

本文第1节介绍物联网服务监测的相关方法和研究现状,第2节以应用示例说明现有工作不足及本文研究动机,第3节对复合服务联动定量监测问题进行定义,第4节介绍所提出的聚合型信号时序逻辑,第5节介绍构建的基于聚合型信号时序逻辑的复合服务联动定量监测机制,第6节通过对比实验验证了所提方法的有效性,最后第7节总结全文。

## 1 物联网服务监测相关工作

物联网范式已被认为是一个关键的驱动力,它使物联网设备的互连和互操作性成为可能,并为最终用户提供越来越多的物联网服务。因此,选择满足用户需求的合适的物联网服务已成为学术界和工业界的共同研究课题,即通过服务调用、组合、适配等技术,实现服务请求或应用程序的应答。目前,已有不少工作研究了不同物联网服务组合技术来满足物联网应用,功能匹配的物联网服务在 QoS 约束下,通过最大化非功能约束满足度,可得到最优物联网服务组合方案<sup>[28,29]</sup>。例如, Chattopadhyay 等人<sup>[30]</sup>和 Palade 等人<sup>[31]</sup>提出了基于 QoS 约束的服务组合技术, Haris 等人<sup>[16]</sup>介绍了 QoS 感知的物联网服务部署方案,以在不同 QoS 约束下实现相应最佳解决方案。然而,现有大多数工作都仅在当前状态下寻找最优解,例如通过有目标函数计算出的满意度最高的组合方案。事实上,物联网设备的资源(例如,通讯带宽、CPU 资源等)在执行过程中可能会动态变化。因此,这些物联网设备可能没有足够的剩余资源来支持其上托管的物联网服务,尽管在求得物联网服务组合方案时刻下可以。因此,在物联网服务执行期间,其 QoS 约束满足程度可能发生动态变化,这使得对其进行实时定量监测已成为保证物联网服务组合方案鲁棒性的关键。

考虑边缘设备上资源稀缺且通常难以得到有效补充,且随着应用的执行/完成和新应用的持续部署,边缘节点可用资源的状态呈现动态波动的特征,处于运行态的应用因为网络拓扑变化和资源波动,使得预配资源难以得到持续保障。目前已有相关研究工作提出不同服务监测机制,以保障物联网服务应答相关应用的健壮性<sup>[7,8]</sup>。例如,雾计算环境中的资源监测机制<sup>[9]</sup>、物联网自适应监测策略<sup>[10]</sup>、移动边缘计算中 QoS 监测方法<sup>[11]</sup>。通常,这些方法将 QoS 感知的物联网服务监测问题规约为约束满足问题<sup>[12]</sup>,并利用适应度评估函数或其他数学对象实现定量监测<sup>[13]</sup>。然而,这些监测工作在形式化解释上仍存不足,表现为较强的领域相关性和经验主观性。为了解决这一问题,本文旨在以定性和定量的方式形式化解释物联网服务上所指定的所有或部分 QoS 约束的实时满意度,从而在物联网服务执行过程中实现实时定量监测。

另外,对目前应用较广泛的主动监测 (active monitoring) 和被动监测 (passive monitoring) 技术进行了调研<sup>[32]</sup>。通常,主动监测会向网络中注入探针,并根据测量的参数推断网络的性能。它的缺点在于会带来额外的流量,并且该方法由于模拟用户在网络上的行为方式,导致所收集的数据并非真实用户的数据。相比之下,被动监测是非侵入式的,并且不会引入额外的流量<sup>[33]</sup>。该方法收集网络中已经存在的数据,并从真实数据中推断出网络的性能。然而,该方法通常需要完整或大量的数据来进行数据分析、处理与决策等,其过程耗时且低效,可能不适用于资源受限的物联网服务监测场景。总的来说,主动和被动监测各有利弊,但在应用到物联网服务实时监测中仍存缺陷。

近年来,研究学者提出了不同的形式化方法来解决相关监测问题<sup>[34,35]</sup>。考虑到物联网服务动态组合的时序性,可将其转换为时序应用问题,从而将之转化为时序逻辑任务,即基于时序逻辑对其进行规约与验证。传统的时序逻辑包括线性时序逻辑、计算树时序逻辑、度量时序逻辑,以及信号时序逻辑<sup>[36]</sup>等,可对连续时间片内的动态系统进行语义规约,将时序逻辑任务转换为可以处理的约束范式。考虑多重约束规划生成可满足任务需求的解决方案,并已被广泛应用于路径规划、协同控制、实时监测等实际业务场景中。例如,Chen 等人<sup>[37]</sup>在支持物联网的系统中采用时序逻辑进行语义故障诊断;黄钰翀等人<sup>[38]</sup>利用时序逻辑实现了多机器人的路径协同规划;Ma 等人<sup>[39]</sup>结合信号时序逻辑和循环神经网络实现了动态环境的预测式监测;Bartocci 等人<sup>[40]</sup>对时序逻辑及其相关拓展工作与应用领域做了较全面的梳理。但是,目前基于相关时序逻辑对物联网服务实现逻辑规约与形式化验证工作较少,大多

数工作仅以在线或离线监测方式实现不同场景下的物联网服务多维属性和时间约束等的实时监测. 考虑到边缘设备上资源的稀缺性, 能耗低、代价小是物联网服务动态监测的绩效目标之一, 并考虑到时序逻辑本身形式化验证优点和轻量级特性, 本文提出基于时序逻辑的物联网服务运行态实时监测方法, 可提高物联网服务监测的逻辑可解释性和形式化验证的可信性. 其中, 信号时序逻辑相较于其他时序逻辑, 不仅可规定个性化的约束, 还可对约束满足度进行定量分析. 为量化感知物联网服务在网络动态变化与时序演进中发生失配的程度大小, 可基于信号时序逻辑实时监测物联网服务在一段时间内定性与定量的约束满足度.

然而, 传统信号时序逻辑的表现力不足以指定某些场景中的复杂需求, 因此现有工作就信号时序逻辑提出了不同拓展型工作. 例如, Ma 等人<sup>[15]</sup>提出了一种新型的空间聚合信号时序逻辑, 以实现智慧城市中的空间感知监测. 当前这些工作构建新型的运算符以扩展传统信号时序逻辑, 来表达特定场景下的某些属性或约束<sup>[41,42]</sup>. 然而, 这些工作仍使用传统信号时序逻辑中的求最大值和最小值函数来实现其他运算符. 正如 Mehdipour 等人<sup>[20]</sup>所述, 传统信号时序逻辑所提出的鲁棒性度量方法是非凸的和不可微的. 为了解决信号时序逻辑的这一不可微性问题, 已经提出了不同改进或拓展技术<sup>[18,19]</sup>. 另外, 为了保证监测的鲁棒性, 可以提出满足和违反部分的不同运算操作, 例如不同的基于平均的鲁棒性函数和平滑整流器<sup>[20,21]</sup>. 需强调的是, 目前工作主要依赖于一个假设, 即以固定的间隔采样信号, 而使其难以应用于非固定采样率场景或动态场景中. 为了解决这些限制, 本文将新构建一种基于黎曼和算子的鲁棒性度量方式来拓展信号时序逻辑, 以允许任意间隔的信号, 同时保证可微性和鲁棒性.

尽管目前已有较多技术对信号时序逻辑进行扩展与优化, 但这些拓展型信号时序逻辑仍无法很好解决物联网服务的定量监测问题. 经研究可发现现有工作缺乏对于物联网服务整体性、联动性、动态性的考虑, 具体包括: (1) 现有技术主要聚焦于单个或多个物联网服务的监测, 未对物联网服务组合形成的复合服务进行整体监测<sup>[22]</sup>. 但实际中应考虑复合服务的监测, 因为单个服务的失配, 可能不代表复合服务的失配; (2) 另一方面, 一个物联网服务的失效可能导致相关有依赖关系的物联网服务失效, 应实现物联网服务的联动监测<sup>[23]</sup>; (3) 现有监测方案大多基于静态或预设的规则、阈值和参数等, 未考虑随实际动态场景进行自适应调整, 而难以运用于动态变化和时序演进的物联网服务监测问题<sup>[25]</sup>. 为解决这些不足, 本文将对信号时序逻辑扩展聚合算子和动态阈值, 构建更适用于物联网服务智能化监测的方法, 实现复合服务的联动定量监测.

## 2 复合服务实时监测应用示例

图 1 展示了一个应用案例来说明基于聚合型信号时序逻辑的物联网服务实时定量监测过程. 具体而言, 如图 1 左下部 (“物联网服务约束”) 所示, 物联网服务在边缘网络实时监测中考虑了多种物联网服务 QoS 约束, 包括空间相关性、时间约束、能量有限, 和资源限制等. 这些约束可能会在其整个生命周期 (从几个月到几年) 或相对较短的持续时间 (从几分钟到几天) 内受到监测.

如图 1 左中部 (“资源失配监测需求”) 所示, 用户可指定物联网服务上的监测需求, 如 “物联网服务  $s_i$  的数据量在未来 24 h 内总是不大于 10 KB” (这是为了保证除物联网服务  $es_i$  之外还有足够的存储空间来执行  $es_i$  所部署的边缘节点上其他所承载的物联网服务). 此监测需求可以形式化表示为信号时序逻辑公式, 即  $\phi_{es_i}^{ds} = G_{[0,24(\text{hours})]} es_i \cdot ds \leq 10 \text{ (KB)}$ . 由此, 可根据一段时间段内的信号值判断该物联网服务是否能够满足该监测需求 (定性化满足度), 并给出其满足度或失配程度大小 (量化化满足度).

如图 1 右中部 (“一段时间内存储空间信号值”) 所示, 监测期间每个时间点下物联网服务  $es_i$  占用的存储空间不同, 如图蓝色和橙色线示例. 其中, 可满足用户需求区 (即 satisfying part, S.P.) 和不满区 (即 violating part, V.P.) 如图 1 所示. 由图 1 可看出, 蓝色和橙色所示信号轨迹线不同但具有相同的极值点 (即指信号轨迹中的最大值或最小值). 例如, 在时间点 0 时物联网服务占用空间大小均为 15 KB. 此外, 橙色线所示信号轨迹以固定时间间隔 (即 1.5 个时间单元) 记录, 而蓝色线所示信号轨迹则在时间点 0、1、3、4 和 6 时记录, 即以非固定间隔采样记录.

根据图 1 所示信号轨迹, 即可定性和定量评估用户监测需求的满足性和满足程度. 以图 1 为例, 本文所提出的

聚合型信号时序逻辑, 将解决传统信号时序逻辑及其最新拓展研究工作所遇到的以下问题.

- 信号不可辨性: 由于传统信号时序逻辑的鲁棒性度量仅依赖于极值点 (例如, 时间点 0 时占用空间大小为 15 KB), 因此对于图 1 所示的蓝色和橙色两条信号轨迹线的监测结果相同, 即传统信号时序逻辑存在信号值迥异、满足度可能无差异的不可分辨性问题.

- 方法不鲁棒性: 最新拓展方法已注意到该缺陷, 并有效解决传统信号时序逻辑中的信号不可分辨性. 然而, 这些工作没有区分 S.P. 和 V.P., 可能存在不满足约束的信号值被满足约束的信号值覆盖, 导致其监测方法不鲁棒. 例如, 蓝线信号轨迹的平均满意度值为 0.4 (正值, 即代表满足; 负值, 代表不满足; 具体语义解释请详见第 4.2 节), 然而这并不能代表用户监测需求被满足, 如时刻 0 和 1 时其数据占用空间 (即 15 KB 和 12 KB) 超过阈值 10 KB.

- 场景不适用性: 尽管目前已有一些方法设计了相关方法解决上述不鲁棒性问题, 但是当前大多数研究工作均假设信号以周期性时间间隔记录 (例如, 1.5, 如图 1 中橙色线所示). 然而, 这种假设在很多实际场景中可能不成立, 如边缘物联网场景中存在着潜在的不确定和动态变化. 图 1 中蓝色轨迹线即是违反此假设的示例, 可看出其信号是非周期性记录的.

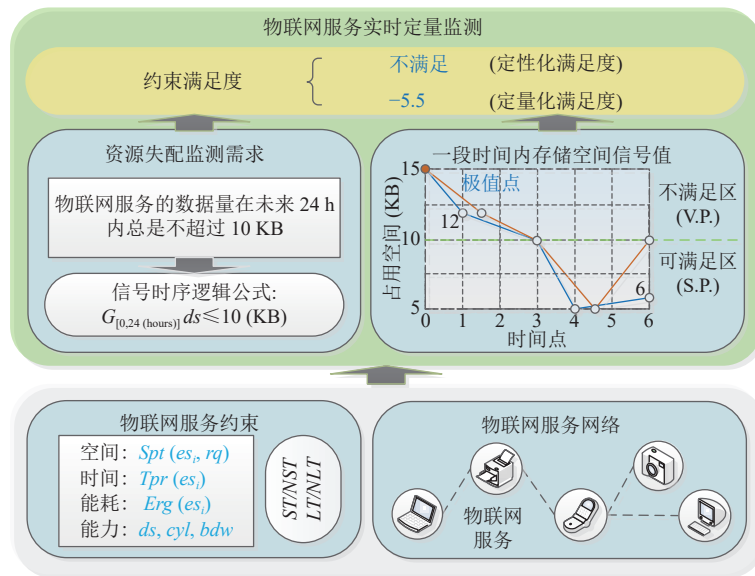


图 1 物联网服务实时定量监测示例

为了解决这些问题, 本文对信号时序逻辑进行拓展, 提出了一种新的鲁棒性度量方法, 如第 4 节所述, 其鲁棒性度量遵循如下原则: ① 不同于传统信号时序逻辑仅依赖于极值点, 该方法在整个时域上进行鲁棒性度量, 从而有效解决信号不可辨问题; ② 通过将 S.P. 和 V.P. 的正负鲁棒性值分开, 并分别构建正负偏置的黎曼和累加算子, 从而实现时序鲁棒的实时监测方法; ③ 允许信号任意采样间隔, 从而实现在可能出现不确定性的场景中的适用性, 例如物联网服务的实时监测. 因此, 本文提出了一种用于定性和定量解释的新型鲁棒性度量方式, 以支持物联网服务实时定量监测. 在此应用示例中, 物联网服务  $es_i$  无法满足  $\phi_{es_i}^{ds}$  所表示的监测需求 (如时间点 0 和 1 时违反监测需求), 即监测结果返回 Failure, 如图 1 所示; 并给出其违反度值, 即 -5.5 (由本文所提出的新型鲁棒性度量方式计算), 表示该物联网服务失配程度.

另一方面, 物联网服务通常通过不同组合结构组合形成复合服务, 以应答用户的服务请求. 在此, 以一个购买支付流程<sup>[43]</sup>为应用示例, 图 2 所示 9 个物联网服务根据该应用需求组合形成复合服务. 用户可针对这些物联网服务给定其运行期间的的时间约束和 QoS 约束, 以实现单个服务自身监测、服务之间联动监测、复合服务整体监测, 以及考虑到约束及其参数动态调整的自适应监测. 下面将分别给出具体监测示例.

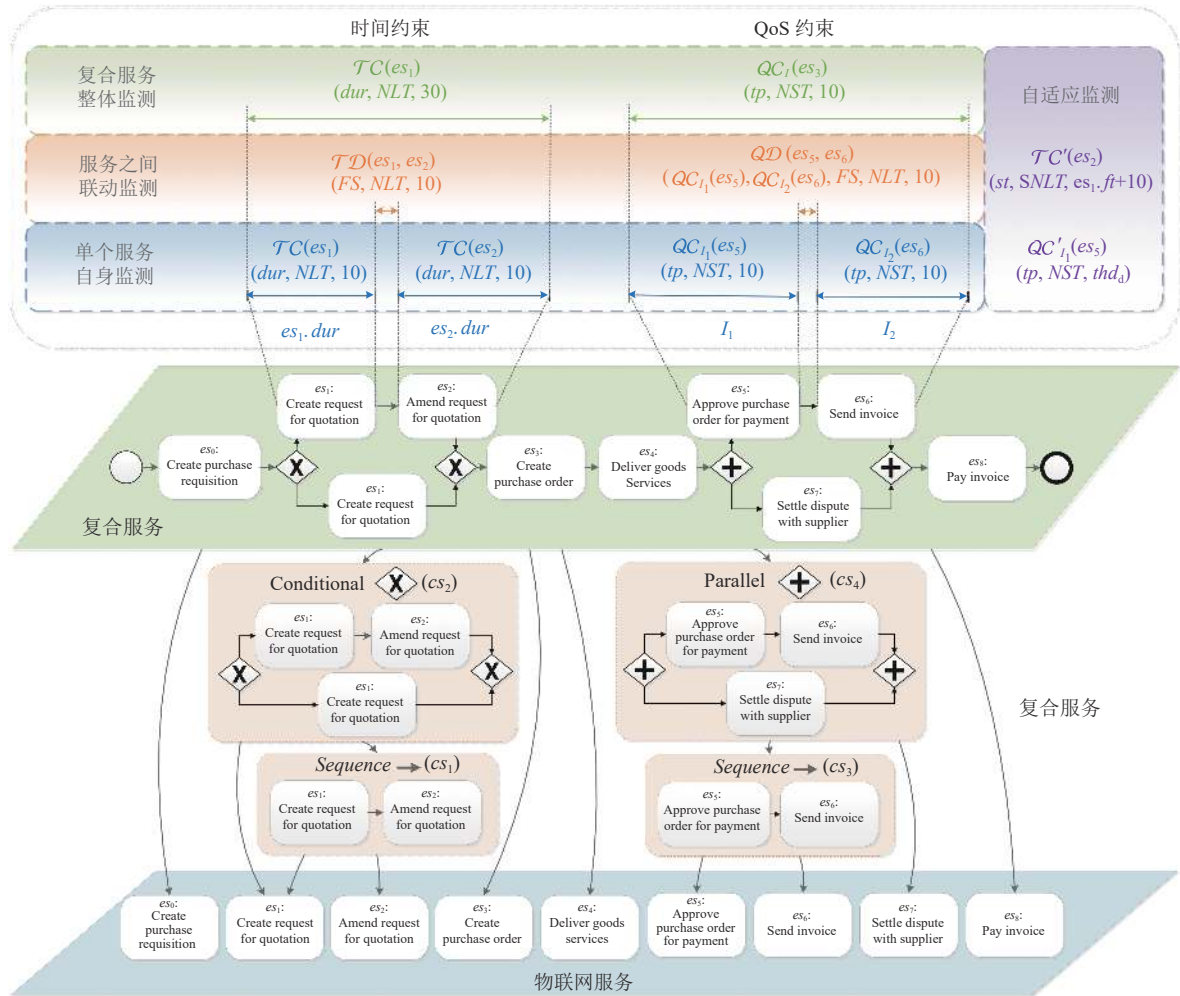


图 2 复合服务联动监测示例

例 1: 可对物联网服务指定时间约束, 以监测其运行期间是否发生时间异常。

• 首先, 以服务自身时间约束为例, 可对单个或多个物联网服务进行时间约束。在图 2 示例中,  $TC(es_1)$  (或  $TC(es_2)$ ) 即代表“物联网服务  $es_1$  (或  $es_2$ ) 须在 10 min 内完成其执行”。

• 此外, 可针对服务之间指定时间约束, 也称为服务之间时间依赖。如图 2 所示示例  $TC'(es_1, es_2)$ , 即代表“物联网服务  $es_2$  须在物联网服务  $es_1$  执行完后的 10 min 开始执行”。

• 同时, 在物联网服务组合形成复合服务后, 可针对复合服务的执行指定其时间约束。例如, 图 2 所示  $TC(cs_1)$ , 即代表“物联网服务  $es_1$  和  $es_2$  在以顺序结构组合形成复合服务  $cs_1$  (即  $es_1 \rightarrow es_2$ ) 后, 该复合服务  $cs_1$  总耗时须在 30 min 内”。

例 2: 此外, 可对物联网服务自身、服务之间, 以及复合服务指定 QoS 约束, 以监测物联网服务运行期间是否发生 QoS 质量下降。

• 首先, 如图 2 所示,  $QC_{I_1}(es_5)$  和  $QC_{I_2}(es_6)$  即为服务自身 QoS 约束示例, 分别代表“物联网服务  $es_5$  (或  $es_6$ ) 在其执行期间, 即  $I_1$  (或  $I_2$ ), 其所需吞吐量总是不小于 10 Kb/s”。

• 其次, 可针对物联网服务之间指定 QoS 约束, 即服务之间 QoS 依赖。例如, 图 2 所示  $QD(es_5, es_6)$ , 即代表“物联网服务 QoS 约束  $QC_{I_2}(es_6)$  须在  $QC_{I_1}(es_5)$  约束后 10 min 内满足”。

• 最后, 可针对复合服务的执行指定其 QoS 约束, 如  $QC_i(cs_3)$ , 即代表“复合服务  $cs_3$  (即  $es_5 \rightarrow es_6$ ) 的总吞吐量需在其执行期间  $I$  内总是不小于 10 Kb/s”。

例 3: 另外, 上述所述不同时间约束和 QoS 约束可能在物联网服务执行过程中发生动态变化, 其相关参数需根据实际情况进行调整, 以实现该约束的自适应监测。

• 在实际监测中, 物联网服务上的约束可能受其相关联服务的影响. 如例 1 中所示  $\mathcal{TD}(es_1, es_2)$ , 物联网服务  $es_2$  的开始时间依赖于物联网服务  $es_1$  的结束时间. 因此, 该物联网服务  $es_2$  在实际监测中, 其时间约束可被表示为  $\mathcal{TC}'(es_2)$ , 如图 2 所示.

• 另一方面, 物联网服务在实际动态边缘网络中, 其约束中相关变量可能受网络中可用资源或算力影响. 例如, 图 2 中所示  $QC'_i(es_5)$ , 即代表“物联网服务  $es_5$  的吞吐量由其所承载物联网设备可用动态资源限制, 即不大于其可用最大吞吐量”。

因此, 为了实现上述物联网服务上的这些约束监测, 本文在传统信号时序逻辑基础上进行扩展 (如第 4 节所述), 以实现该复合服务联动定量监测, 解决传统信号时序逻辑及其最新拓展工作所面临的以下问题.

• 复合服务整体性判别不足: 传统信号时序逻辑及其相关拓展方法仅针对单个服务进行监测, 而复合服务在以整体进行监测时, 其监测结果不是其子服务监测结果的简单逻辑并运算. 如例 1 中复合服务  $cs_1$  的时间约束  $\mathcal{TC}(cs_1)$  满足时, 即复合服务  $cs_1$  总耗时在 30 min 内, 可能存在单个服务耗时超过其预设阈值. 如服务  $es_1$  耗时 20 min (约束在 10 min 内, 违反用户约束), 服务  $es_2$  耗时 5 min (满足用户约束),  $cs_1$  可满足用户约束, 但其子服务未能都满足其相应时间约束. 由此, 对服务  $es_1$  和  $es_2$  监测结果进行逻辑并运算结果为该复合服务不满足用户约束, 但实际复合服务可满足  $\mathcal{TC}(cs_1)$ .

• 服务之间联动性支持不足: 考虑到服务执行过程中, 后执行的服务可能依赖之前已执行服务的时间/QoS 约束, 而现有监测方法无法在执行过程中自动根据依赖服务进行实时监测. 例如, 现有方法无法监测“物联网服务  $es_2$  在物联网服务  $es_1$  执行完后的 10 min 开始执行”这样的时间约束.

• 边缘场景动态性考虑不足: 物联网服务的监测受边缘场景动态影响, 监测参数、阈值等依赖于实际网络资源. 如例 3 中  $QC'_i(es_5)$  所示监测需求.

为了解决上述问题, 本文对信号时序逻辑进行拓展, 提出了一种面向复合服务动态监测场景的新型信号时序逻辑, 其监测过程遵循如下原则: ① 不同于传统信号时序逻辑仅考虑单个物联网服务约束满足度, 该方法依据多个物联网服务之间的组合结构以及所监测的时间/QoS 属性, 构建并拓展聚合算子, 从而有效解决复合服务整体上不同约束的监测问题; ② 通过拓展一种可随动态环境变化的动态变量, 以提高信号时序逻辑的表达能力, 解决物联网服务在边缘场景中的自适应监测问题, 并实现服务之间联动关系下的实时监测. 综上所述, 本文针对现有信号时序逻辑存在的鲁棒性度量缺陷以及应对复合服务监测不足的问题, 提出了一种聚合型信号时序逻辑, 对传统信号时序逻辑进行语法拓展和语义重定义, 以支持物联网复合服务的实时联动定量监测.

### 3 复合服务联动定量监测问题定义

本节首先回顾相关概念, 再对物联网服务上的时间约束和 QoS 约束进行详细介绍, 并对复合服务联动定量监测问题进行定义与说明. 首先, 可对复合服务定义如下.

**定义 1 (复合服务).** 复合服务  $cs$  即为一种新的增值服务, 由一组原子边缘物联网服务通过特定结构编排组成<sup>[44]</sup>. 形式上, 它可以使用 Backus-Naur (BNF) 语法定义如下:

$$cs := es \mid cs_1 \rightarrow cs_2 \mid cs_1 \oplus cs_2 \mid cs_1 \otimes cs_2,$$

其中,  $es$  代表边缘网络中的物联网服务;  $cs$ 、 $cs_1$  和  $cs_2$  为复合服务;  $\rightarrow$  代表顺序结构, 表示  $cs_2$  在  $cs_1$  执行完后开始执行;  $\oplus$  代表并行结构, 表示  $cs_1$  和  $cs_2$  同时执行;  $\otimes$  代表选择结构, 表示选择执行  $cs_1$  或  $cs_2$ .

图 2 中展示了在购买支付流程中的多种边缘物联网服务, 即  $\{es_i\}_{i=0}^8$ . 示例复合服务为  $\{cs_i\}_{i=1}^4$ , 例如  $cs_1 = es_1 \rightarrow es_2$  和  $cs_4 = (es_5 \rightarrow es_6) \oplus es_7$ . 在本文中, 一个物联网服务可记作一个原子服务, 即  $|es| = 1$ , 例如图 2 中所示  $es_1$ ; 一



个复合服务即由多个原子服务组成, 即  $|cs| > 1$ , 例如上述示例中  $cs_1$ ; 在本文中, 可将一个边缘网络中的物联网服务及其复合服务记作  $S = \{s_1, s_2, \dots, s_i, \dots\}$ , 其中  $s_i := es | cs$ .

### 3.1 时间约束

本节对物联网服务及其复合服务上指定的各种时间约束进行规约<sup>[24]</sup>, 包括: ① 单个服务自身时间约束: 用于确保该服务可在适当时间执行, 并在一定时间内完成; ② 服务之间时间依赖: 用于约束服务之间的依赖及相互影响; 以及③ 对复合服务的时间约束: 用于确保该复合服务的执行没有时间违规.

#### 3.1.1 服务自身时间约束

给定物联网服务  $es$  的时间约束, 可以指定对开始时间、结束时间和持续时间的要求 (如图 2 示例说明). 具体说明如下.

- 开始时间 (记作  $es.st$ ), 可以约束为不早于某一时间点开始 (start no earlier than,  $SNET$ )、不晚于某一时间点开始 (start no later than,  $SNLT$ ), 或必须开始于某一时间点 (must start on,  $MSO$ ), 记作  $CNT_{st} = \{SNET, SNLT, MSO\}$ ;

- 结束时间 (记作  $es.ft$ ), 可以限制为不早于某一时间点结束 (finish no earlier than,  $FNET$ )、不晚于某一时间点结束 (finish no later than,  $FNLT$ ),  $FNLT$  又名截止日期 (deadline,  $DDL$ ), 或必须于某一时间点结束 (must finish on,  $MFO$ ), 记作  $CNT_{ft} = \{FNET, FNLT \text{ (or } DDL), MFO\}$ ;

- 持续时间 (记作  $es.dur$ , 即  $es.ft - es.st$ ), 可以被约束为小于某一数值 (smaller than,  $ST$ , 即  $<$ )、大于某一数值 (larger than,  $LT$ , 即  $>$ )、不小于某一数值 (no smaller than,  $NST$ , 即  $\geq$ ), 和不大于某一数值 (no larger than,  $NLT$ , 即  $\leq$ ), 记作  $CNT = \{ST, LT, NST, NLT\}$ .

由此, 可定义物联网服务自身的时间约束如下.

**定义 2 (服务自身时间约束).** 对于某一物联网服务  $es$ , 可对其自身给定时间约束如下元组:  $\mathcal{TC}(es) = (ita, cnt, thd)$ , 其中:

- $ita := es.st | es.ft | es.dur$  表示该时间约束类型, 即开始时间、结束时间或持续时间.
- $cnt$  代表该时间约束的约束形式, 即当  $ita := es.st, cnt \in CNT_{st}$ ; 当  $ita := es.ft, cnt \in CNT_{ft}$ ; 当  $ita := es.dur, cnt \in CNT$ .
- $thd$  代表  $cnt$  上相应的约束阈值或变量, 例如, 当  $ita := es.st, cnt := SNET, thd$  即为具体时间点, 如 9 点, 表示该物联网服务开始时间不早于 9 点.

第 2 节在图 2 中即展示了示例时间约束  $\mathcal{TC}(es_1) = (es_1.dur, NLT, 10)$ .

#### 3.1.2 服务之间时间依赖

此外, 可对两个直接或间接连接的物联网服务之间指定时间关系 (temporal relation). 一个物联网服务  $es_i$  可能: ① 发生在另一个物联网服务  $es_j$  之前 (before) 或正好相接 (meet), 当  $es_i \rightarrow es_j$ ; 或 ② 与另一个物联网服务  $es_j$  重叠 (overlap)、触发开始 (start)、处于执行期间 (during)、触发结束 (finish), 或相等 (equal), 当  $es_i \oplus es_j$ .

基于两个物联网服务之间的时间关系, 可对服务之间的 4 种时间依赖关系 (temporal dependency) 进行规约, 即包括: ① 开始到结束 (start-to-finish,  $SF$ ): 对于物联网服务之间的上述所有时间关系, 均可对其进行规约, 即考虑两个服务的执行总时间; ② 结束到开始 (finish-to-start,  $FS$ ): 通常在  $es_i \rightarrow es_j$  时可对服务之间的时间间隔进行约束; ③ 开始到开始 (start-to-start,  $SS$ ) 和 ④ 结束到结束 (finish-to-finish,  $FF$ ): 当  $es_i \oplus es_j$  时,  $SS$  和  $FF$  可能会受到时间约束以监测两个服务的执行情况. 基于该时间关系和时间依赖, 可定义物联网服务之间时间依赖如下.

**定义 3 (服务之间时间依赖).** 对于两个物联网服务  $es_i$  与  $es_j$ , 可定义服务之间时间依赖如下元组:  $\mathcal{TD}(es_i, es_j) = (itr, cnt, thd)$ , 其中,

- $itr := SS | SF | FS | FF$  表示该时间依赖类型.
- $cnt \in CNT$  代表该时间依赖的约束形式.
- $thd$  代表  $cnt$  上相应的约束阈值.

图 2 中即展示了示例时间依赖  $\mathcal{TD}(es_1, es_2) = (FS, NLT, 10)$ .

### 3.1.3 复合服务时间约束

另一方面,可对复合服务指定时间约束,以保证复合服务在其执行期间没有时间违规的情况.在此,可首先对复合服务  $cs \in \mathcal{S}$  内所包含的原子服务集表示如下:

$$L(cs) = \begin{cases} \{es \mid cs := es, es \in \mathcal{S}\} \\ \{\mathcal{L}(cs_1) \cup \mathcal{L}(cs_2) \mid cs := cs_1 (\rightarrow / \oplus / \otimes) cs_2\} \end{cases} \quad (1)$$

该复合服务  $cs$  上的时间约束与物联网服务  $es$  自身时间约束类似,即可表示为  $\mathcal{TC}(cs) = (ita, cnt, thd)$ , 其中,  $ita := cs.st \mid cs.ft \mid cs.dur$ , 具体参数为:

$$cs.st = \min \{es_i.st \mid \forall es_i \in \mathcal{L}(cs)\} \quad (2)$$

$$cs.ft = \max \{es_i.ft \mid \forall es_i \in \mathcal{L}(cs)\} \quad (3)$$

$$cs.dur = cs.ft - cs.st \quad (4)$$

示例约束如  $\mathcal{TC}(cs_1)$ , 可详见图 2 中应用示例. 由此,针对原子服务以及复合服务的自身时间约束,可表示为  $\mathcal{TC}(s)$ , 其中  $s := es \mid cs, s \in \mathcal{S}$ . 类似地,可将原子服务或复合服务之间的时间依赖表示为  $\mathcal{TD}(s_i, s_j) = (itr, cnt, thd)$ , 其中  $s_i := es \mid cs, s_j := es \mid cs$ , 且  $s_i \in \mathcal{S}, s_j \in \mathcal{S}$ .

## 3.2 QoS 约束

在本文中,物联网服务 QoS 属性值随时间变化. 首先,可对时间区间 (interval) 定义如下<sup>[42]</sup>.

**定义 4 (时间区间).** 给定两个时间点  $t_0$  与  $t_n$ ,  $t_0$  与  $t_n$  之间的时间区间可被定义  $I = [t_0, t_n]$ , 即为  $I = \{t_i \mid t_0 \leq t_i \leq t_n\}$ , 其中  $t_i$  为该时间区间中的时间点,  $i \in \mathbb{Z}^{\geq 0}, t_0 \leq t_n$ , 且  $I \subseteq \mathbb{R}^{\geq 0}$ . 由此,  $t + I$  可用来表示时间区间  $[t + t_0, t + t_n]$ .

具体而言,本节对物联网服务及其复合服务上指定的各种 QoS 约束进行规约,包括: ① 单个服务自身 QoS 约束: 用于确保该服务可满足用户服务请求的非功能约束; ② 服务之间 QoS 依赖: 用于约束服务之间的 QoS 依赖及相互影响; 以及 ③ 对复合服务的 QoS 约束: 用于确保该复合服务在其执行期间可持续维持高 QoS. 在实际的物联网服务 QoS 约束监测过程中,不同用户可能需针对不同 QoS 约束进行监测,如本文及实验章节中所提到的吞吐量、响应时间等. 对于其他不同 QoS 约束,如边缘设备能耗、CPU 资源、通讯带宽、数据大小等 QoS 约束,本文所提监测方法也均可适用. 当考虑到 QoS 约束可能存在重要性差异,可对其约束满足度设置合适的权重进行加权求和,得到有针对性的监测结果. 本文由于重点在于服务监测策略,不在此对其进行详细探讨.

### 3.2.1 服务自身 QoS 约束

与时间约束关联的服务自身 QoS 约束可定义如下.

**定义 5 (服务自身 QoS 约束).** 对于某一物联网服务  $es$ , 可对其自身给定 QoS 约束为  $\mathcal{QC}_I(es)$ , 即在其执行期间  $I$  内 (即在每一时间点  $\{t_i\}_{i=0}^n$ ) 的一组 QoS 约束. 可表示为  $\mathcal{QC}_I(es) = \{\mathcal{QC}_{t_i}(es)\}_{i=0}^n$ , 其中,  $\mathcal{QC}_{t_i}(es) = (atr, cnt, thd)$  代表时间点  $t_i$  的 QoS 约束, 其中:

- $atr$  代表该物联网服务上的特定 QoS 属性, 例如响应时间、吞吐量等.
- $cnt \in CNT$  代表该 QoS 约束的约束形式.
- $thd$  代表  $cnt$  上相应的约束阈值.

图 2 右上角部分展示了示例 QoS 约束  $\mathcal{QC}_{t_i}(es_5)$ . 如图 3(a) 所示, 蓝色线表示 QoS 值, 绿色线表示阈值, QoS 约束为“物联网服务所需吞吐量总是不小于 ( $NST$ ) 阈值”. 其中, 蓝色点为可满足用户 QoS 约束点, 红色点为不满足点.

物联网服务上的 QoS 约束  $\mathcal{QC}_I(es)$  即表示在时间区间  $I$  内每一时间点下的 QoS 约束集合, 即  $\mathcal{QC}_I(es) = \{\mathcal{QC}_{t_i}(es) = (atr, cnt_{t_i}, thd_{t_i}) \mid \forall t_i \in I\}_{i=0}^n$ , 其中,  $cnt_{t_i}$  和  $thd_{t_i}$  可随时间变化, 即  $\mathcal{QC}_I(es) = \{\mathcal{QC}_{t_i}(es)\}_{i=0}^5$ , 其中,  $I = [t_0, t_5], \forall t_i \in I$ : (a)  $\mathcal{QC}_{t_i}(es) = (tp, NST, 10)$ ; (b)  $\mathcal{QC}_{t_i}(es) = (tp, NST, thd_{t_i})$ ; (c)  $\mathcal{QC}_{t_i}(es) = (tp, cnt_{t_i}, 10)$ ; (d)  $\mathcal{QC}_{t_i}(es) = (tp, cnt_{t_i}, thd_{t_i})$ . 其中,  $thd_{t_i}$  和  $cnt_{t_i}$  随时间  $t_i$  变化, 即  $thd_{t_i} = \{10, 5, 10, 0, 5, 5\}$ ,  $cnt_{t_i} = \{NLT, NLT, NST, NST, NST, NST\}$ .

### 3.2.2 服务之间 QoS 依赖

考虑到物联网服务的 QoS 约束可能会受到另一个服务的影响, 可对服务之间 QoS 依赖关系规约如下.

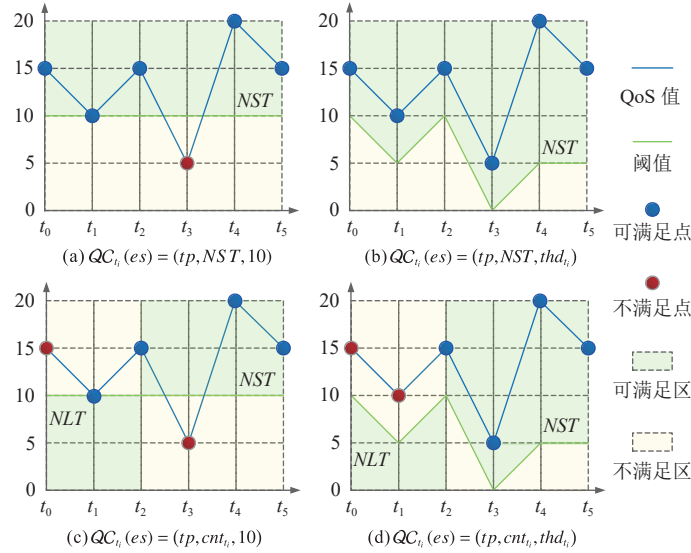


图3 物联网服务自身 QoS 约束示例

**定义 6 (服务之间 QoS 依赖).** 对于物联网服务  $es_i$  与  $es_j$  之间的 QoS 依赖, 可将其定义为如下元组:  $QD(es_i, es_j) = (QC_{t_1}(es_i), QC_{t_2}(es_j), itr, cnt, thd)$ , 其中,  $QC_{t_1}(es_i)$  和  $QC_{t_2}(es_j)$  分别代表物联网服务  $es_i$  与  $es_j$  上的 QoS 约束,  $itr := SS | SF | FS | FF$ ,  $cnt \in CNT$  代表一个约束形式,  $thd$  则代表相应约束阈值.

图 2 右上角部分即展示了示例 QoS 依赖  $QD(es_5, es_6)$ , 代表物联网服务  $es_5$  与  $es_6$  之间的 QoS 依赖.

### 3.2.3 复合服务 QoS 约束

与复合服务上的时间约束类似, 复合服务上也可指定相关 QoS 约束, 即  $QC_I(cs)$ , 代表时间区间  $I$  内每一时刻点下的 QoS 约束, 即  $QC_I(cs) = \{QC_{t_i}(cs)\}_{i=0}^n$ , 其中  $QC_{t_i}(cs)$  代表复合服务  $cs$  在时间点  $t_i$  下的 QoS 约束, 即  $QC_{t_i}(cs) = (atr, cnt, thd)$ , 如定义 5 所示.

不同于单个物联网服务上的 QoS 约束, 复合服务  $cs$  上某一属性  $atr$  的 QoS 属性值由其所包含物联网服务的 QoS 属性值进行聚合计算. 具体而言, 复合服务  $cs$  的 QoS 属性值可以通过最大值 (maximum)、最小值 (minimum)、加法 (additive)、平均值 (average) 或乘法 (multiplicative) 运算求得, 记作  $\mathcal{F} = \{\max, \min, \sum, \text{avg}, \prod\}$ . 在此, 可记作  $f_{atr}^{str} \in \mathcal{F}$  为特定聚合运算符, 即其取决于 ① 复合服务  $cs$  的组合结构, 记作  $str \in STR = \{\rightarrow, \oplus, \otimes\}$ ; 和 ② 该 QoS 属性  $atr$  类型. 表 1 展示了示例聚合运算符, 即不同组合结构下的复合服务求其不同 QoS 属性值的聚合运算符.

表 1 聚合运算符示例表

QoS属性	$\rightarrow$	$\oplus$	$\otimes$
响应时间	$\sum_{i=1}^{ \mathcal{L}(cs) }$	$\max_{i=1}^{ \mathcal{L}(cs) }$	$\max_{i=1}^{ \mathcal{L}(cs) }$
吞吐量	$\min_{i=1}^{ \mathcal{L}(cs) }$	$\min_{i=1}^{ \mathcal{L}(cs) }$	$\min_{i=1}^{ \mathcal{L}(cs) }$
可靠性	$\prod_{i=1}^{ \mathcal{L}(cs) }$	$\prod_{i=1}^{ \mathcal{L}(cs) }$	$\prod_{i=1}^{ \mathcal{L}(cs) }$
可用度	$\prod_{i=1}^{ \mathcal{L}(cs) }$	$\prod_{i=1}^{ \mathcal{L}(cs) }$	$\min_{i=1}^{ \mathcal{L}(cs) }$
能耗	$\sum_{i=1}^{ \mathcal{L}(cs) }$	$\sum_{i=1}^{ \mathcal{L}(cs) }$	$\max_{i=1}^{ \mathcal{L}(cs) }$

对于图 2 中间部分所示复合服务  $cs_4$ , 即  $cs_4 = (es_5 \rightarrow es_6) \oplus es_7$ , 其总吞吐量为  $cs_4.tp = f_{tp}^{\oplus}(f_{tp}^{\rightarrow}(es_5.tp, es_6.tp), es_7.tp) = \min(es_5.tp, es_6.tp, es_7.tp)$ . 由此, 可针对复合服务指定其 QoS 约束, 即  $QC_I(cs)$ , 如定义 5 所示. 图 2 右上角

部分即给出示例约束  $QC_I(cs_3)$ .

综上所述, 可针对单个物联网服务  $es$ , 或复合服务  $cs$ , 指定 QoS 约束为  $QC_I(s)$ , 其中,  $s := es | cs, s \in \mathcal{S}$ . 类似地, 可将物联网服务或复合服务之间的 QoS 依赖表示为  $QD(s_i, s_j) = (QC_{I_1}(s_i), QC_{I_2}(s_j), itr, cnt, thd)$ , 其中  $s_i := es | cs, s_j := es | cs$ , 如定义 6 所示.

### 3.3 复合服务联动定量监测问题定义

基于上述定义, 我们将物联网服务运行态监测问题表述为时序任务, 可针对服务自身、服务之间, 以及复合服务上指定不同时间约束和 QoS 约束, 并对其进行持续监测, 以实时发现资源失配、约束违规等情况. 本文假设用户所指定约束已满足约束一致性要求, 不再做一致性验证, 如若需要可通过适当的技术进行验证. 具体而言, 可从以下角度对本文所提出的复合服务联动定量监测问题进行表示.

- 服务自身实时监测: 可对物联网服务自身所指定的各项时间约束和 QoS 约束进行实时监测, 如第 3.1.1 节和第 3.2.1 节所示.
- 服务之间联动监测: 可对物联网服务之间的时间依赖和 QoS 依赖进行规约, 实现服务之间的联动监测, 如第 3.1.2 节和第 3.2.2 节所示.
- 复合服务整体监测: 此外, 可对物联网服务组合形成的复合服务进行整体监测, 以实时判定其是否满足其上时间约束和 QoS 约束, 如第 3.1.3 节和第 3.2.3 节所示.

为实现上述服务自身、服务之间以及复合服务的多项约束监测, 本文提出了自适应监测机制, 即在物联网服务监测期间其监测参数可随时间发生改变. 由此, 为满足用户对于物联网服务及其复合服务运行态的监测需求, 及时有效发现 QoS 降低或时间违反情况, 复合服务联动定量监测问题可形式化表述如下.

**定义 7 (复合服务联动定量监测).** 复合服务联动定量监测问题, 即为在监测时间区间  $I$  内解决物联网服务及其复合服务上各项时间约束和 QoS 约束的定性化及定量化满足问题:

$$\left\{ \begin{array}{l} \text{Qualitative satisfaction } \forall t_i \in I : \forall s_j \in \mathcal{S}, \Psi_{t_i}(s_j) = \{\top, \perp\} \\ \text{Quantitative satisfaction } \forall t_i \in I : \forall s_j \in \mathcal{S}, \eta_{t_i}(s_j) \in \mathbb{R} \\ \text{s.t. } \forall s_j \in \mathcal{S}, \mathcal{TC}(s_j), \mathcal{QC}_I(s_j) \text{ are satisfied, and } \forall s_j, s_k \in \mathcal{S}, \mathcal{TD}(s_j, s_k), \mathcal{QD}(s_j, s_k) \text{ are satisfied.} \end{array} \right.$$

给定在一组物联网服务及其复合服务上指定的时间约束或/和 QoS 约束, 可实现服务运行态下定性和定量解释, 以在每一个运行时刻  $t_i \in I$  下指定: ① 该服务  $s_j \in \mathcal{S}$  是否满足其上指定约束, 即满足 (即  $\Psi_{t_i}(s_j) = \top$ ) 或不满足 (即  $\Psi_{t_i}(s_j) = \perp$ ), 以及 ② 满足度或不满足度大小, 即  $\eta_{t_i}(s_j) \in \mathbb{R}$ . 其中, 用户所指定服务上的时间约束和 QoS 约束包括  $\mathcal{TC}(s_j)$  和  $\mathcal{QC}_I(s_j)$ , 并包括时间依赖  $\mathcal{TD}(s_j, s_k)$  和 QoS 依赖  $\mathcal{QD}(s_j, s_k)$ . 在以下章节中, 将对该复合服务联动定量监测机制进行详细说明, 并提供具体应用案例进行说明, 具体运行实例可参见第 6 节.

## 4 聚合型信号时序逻辑构建

本节将对所拓展的聚合型信号时序逻辑 (compositional signal temporal logic, CSTL) 进行详细介绍, 即从其语法与语义分别叙述, 以支持上述第 3.3 节所述复合服务联动定量监测.

### 4.1 聚合型信号时序逻辑语法

基于定义 4 中时间区间, 可定义多维信号变量来表示一段时间内物联网服务的执行情况, 即  $\omega : I \times \mathcal{S} \rightarrow \mathbb{B} \times \mathbb{R}$ , 其中,  $I \subseteq \mathbb{R}^{\geq 0} := [0, \infty)$  代表一个时间区间;  $\mathcal{S}$  代表边缘网络内被监测的物联网服务及其复合服务集合, 如图 1 和图 2 所示;  $\mathbb{B} = \{\perp, \top\}$ , 其中,  $\perp := \text{False}, \top = \text{True}$ . 针对物联网服务上的时间约束和 QoS 约束, 如第 3.1 节和第 3.2 节所述, 可定义  $\mathcal{X} = \{x_1, x_2, \dots, x_n\} \subseteq \mathbb{R}^n$ , 即物联网服务及其复合服务上的多约束集合. 对每一个约束  $x \in \mathcal{X}$ , 定义其在多维信号变量上的投影, 即  $\pi_x(\omega)$ , 代表时间区间  $I$  内一组服务  $\mathcal{S}$  的变量值集合.

给定边缘网络中一个监测服务  $s \in \mathcal{S}$ , 即单个物联网服务或复合服务,  $s := es | cs$ , 其上某个约束  $x \in \mathcal{X}$  在特定时间点  $t \in I$  上的信号值可规约为:

- $\pi_x(\omega)[t, es]$ : 对于一个物联网服务  $es \in \mathcal{S}$ , 即  $s := es$ , 该  $es$  在时间点  $t \in I$  下变量  $x$  的数值可表示为  $\pi_x(\omega)[t, es]$ .

•  $\pi_x(\omega)[t, cs]$ : 对于一个复合服务  $cs \in \mathcal{S}$ , 即  $s := cs$ , 该  $cs$  在时间点  $t \in I$  下变量  $x$  的数值可表示其所包含物联网服务的变量值的聚合结果:

$$\pi_x(\omega)[t, cs] = C_S^{\mathcal{F}} \pi_x(\omega)[t, \mathcal{L}(cs)] = \begin{cases} \pi_x(\omega)[t, es], & \text{if } cs := es \\ f_x^{str} \left( C_S^{\mathcal{F}} \pi_x(\omega)[t, \mathcal{L}(cs_1)], C_S^{\mathcal{F}} \pi_x(\omega)[t, \mathcal{L}(cs_2)] \right), & \end{cases}$$

$$\text{where } f_x^{str}(\cdot) = \begin{cases} f_x^{\rightarrow}(\cdot), & \text{if } cs := cs_1 \rightarrow cs_2 \\ f_x^{\oplus}(\cdot), & \text{if } cs := cs_1 \oplus cs_2 \\ f_x^{\otimes}(\cdot), & \text{if } cs := cs_1 \otimes cs_2 \end{cases} \quad (5)$$

在此, 将  $\pi_x(\omega)[t, es]$  简写为  $v$ ,  $\pi_x(\omega)[t, cs]$  简写为  $C_S^{\mathcal{F}} \mathcal{V}$ . 可给定所构建的聚合型信号时序逻辑 CSTL 语法表示如下.

**定义 8 (CSTL 语法).** 给定公式  $\varphi \in \mathcal{F}$ , 其句法 (syntax) 可由如下语法 (grammar) 定义:

$$\varphi := v \sim c \mid v \sim \xi \mid C_S^{\mathcal{F}} \mathcal{V} \sim \xi \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 U_I \varphi_2,$$

其中,

- $v \sim c$  表示断言,  $\sim = \{<, \leq\}$ ,  $c \in \mathbb{R}$  是用户指定的常数阈值.
- $\neg$  是逻辑非算子, 可用来表示某一监测公式的反例  $\neg \varphi$ .
- $\wedge$  是逻辑与算子, 可用来表示多约束的共同监测, 例如  $\varphi_1 \wedge \varphi_2$ .
- $U_I$  是时序“直到”算子, 表示  $\varphi_2$  能在时间区间  $I$  内至少某一时间点  $t \in I$  满足, 且  $\varphi_1$  在该时间点  $t$  之前总是被满足.

逻辑或算子  $\vee$ 、时序“最终”算子  $F_I$  与“总是”算子  $G_I$ , 与传统信号时序逻辑本身语法定义相同. 具体而言,  $\vee$  是逻辑或算子, 满足多个约束中的至少一个约束即可表示为  $\varphi_1 \vee \varphi_2$ ;  $F_I$  是时序“最终”算子, 表示在时间区间内至少存在某一时间点  $t \in I$ ,  $\varphi$  能被满足;  $G_I$  是时序“总是”算子, 表示在时间区间  $I$  内所有时间点,  $\forall t \in I$ ,  $\varphi$  均能被满足. 逻辑或算子  $\vee$  可由逻辑与算子  $\wedge$  进行推导, 时序“最终”算子  $F_I$  和时序“总是”算子  $G_I$  可由时序“直到”算子  $U_I$  进行推导, 推导过程如下:

$$\varphi_1 \vee \varphi_2 = \neg(\neg \varphi_1 \wedge \neg \varphi_2) \quad (6)$$

$$F_I \varphi = \top U_I \varphi \quad (7)$$

$$G_I = \neg(F_I \neg \varphi) \quad (8)$$

在该聚合型信号时序逻辑 CSTL 中, 本文扩展了动态变量和聚合算子如下.

•  $\xi$  代表复合服务监测过程中的动态变量, 可用来作为物联网服务约束中的不同动态变化阈值. 由此, CSTL 公式  $v \sim \xi$  可用来表示在不同时间点下变量值  $v$  小于或小于等于一个实时捕捉或根据边缘环境自适应变化的阈值. 例如,  $v \sim \xi$  可为: ①  $\pi_{tp}(\omega)[t_0, es] < 10$ , 代表物联网服务  $es$  在时间点  $t_0$  下的吞吐量 (记作  $tp$ ) 需小于 10 Kb/s; ②  $\pi_{tp}(\omega)[t_1, es] < 5$ , 代表物联网服务  $es$  在时间点  $t_1$  下的吞吐量需小于 5 Kb/s. 该变量  $\xi$  的动态特性对于物联网服务自适应监测是必不可少的, 如第 6.5 节中实验所评估与分析结果所示.

• 此外, CSTL 扩展了一个聚合算子  $C_S^{\mathcal{F}} \mathcal{V}$  来对复合服务  $cs$  中所包含的物联网服务 (即  $\mathcal{L}(cs)$ ) 上的不同变量值  $\mathcal{V}$  (即  $\pi_x(\omega)[t, \mathcal{L}(cs)]$ ) 进行聚合运算和约束. 由此, CSTL 公式  $C_S^{\mathcal{F}} \mathcal{V} \sim \xi$  可实现对于复合服务的整体监测. 例如, 对于图 2 所示复合服务  $cs_4$ , 即  $cs_4 = cs_5 \oplus es_7$ ,  $cs_5 = es_5 \rightarrow es_6$ , 其吞吐量可表示为  $C_S^{\mathcal{F}} \mathcal{V} = f_{tp}^{\oplus}(C_S^{\mathcal{F}} \pi_{tp}(\omega)[t, \mathcal{L}(cs_5)], \pi_{tp}(\omega)[t, es_7])$ , 其中,  $C_S^{\mathcal{F}} \pi_x(\omega)[t, \mathcal{L}(cs_5)] = f_{tp}^{\rightarrow}(\pi_{tp}(\omega)[t, es_5], \pi_{tp}(\omega)[t, es_6])$ . 再结合变量  $\xi$  可对该复合服务的吞吐量进行动态约束与实时监测.

由上可知, 根据 CSTL 语法可对第 3.1 节和第 3.2 节中所述物联网服务上的各项时间约束和 QoS 约束进行形式化表达, 即将用户监测需求转换为 CSTL 公式. 以图 2 中所示例, 即第 2 节中例 1–例 3 中所示例监测需求, 可将其转换为相应的 CSTL 公式如下:

• 例 1 中所示时间约束:  $\mathcal{TC}(es_1/es_2)=G_I(es.dur \leq 10)$ ,  $\mathcal{TD}(es_1, es_2)=G_I(es_2.st \leq es_1.ft+10)$ 和  $\mathcal{TC}(cs_1)=G_I(cs_1.ft \leq cs_1.st+30)$ .

• 例 2 中所示 QoS 约束:  $\mathcal{QC}_I(es_5/es_6)=G_I \neg(es.tp < 10)$ ,  $\mathcal{QD}(es_5, es_6)=G_I(FS \leq 10)$ 和  $\mathcal{QC}_I(cs_3)=G_I \neg(C_S^f\{es_5.tp, es_6.tp\} < 10)$ .

• 例 3 中所示约束:  $\mathcal{TC}'(es_2)=G_I(es_2.st \leq es_1.ft+10)$ 和  $\mathcal{QC}'_{I_i}(es_5)=G_I \neg(es_5.tp < thd_d)$ .

#### 4.2 聚合型信号时序逻辑语义

在将用户监测需求或不同约束转换为 CSTL 公式后, 可定义定性化语义, 即定义满足关系  $(\omega, t, s) \models \varphi$ , 表示信号  $\omega$  在时间点  $t$  时可满足物联网服务  $s$  上公式  $\varphi$  所代表约束. 对于判定满足关系  $(\omega, t, s) \models \varphi$  是否成立, 可构建如下定性化语义.

**定义 9 (CSTL 定性化语义).** 基于该信号时序逻辑语法, 可对复合服务定性监测进行规约, 即给定函数  $\Psi: \mathcal{F} \times \mathcal{W} \times I \times SEV \rightarrow \mathbb{B}$ , 将某一给定信号时序逻辑公式  $\varphi \in \mathcal{F}$ , 某一给定的多维信号  $\omega \in \mathcal{W}$ , 某一时间点  $t \in I$ , 和某一服务  $s := es \mid cs, s \in \mathcal{S}$ , 映射到布尔型值域  $\mathbb{B} = \{\perp, \top\}$ :

$$\begin{aligned} \Psi(v \sim c, \omega, t, es) &= \pi_x(\omega)[t, es] \sim c \\ \Psi(x \sim \xi, \omega, t, es) &= \pi_x(\omega)[t, es] \sim \xi \\ \Psi(C_S^f \mathcal{V} \sim \xi, \omega, t, cs) &= \pi_x(\omega)[t, cs] \sim \xi \\ \Psi(\neg \varphi, \omega, t, s) &= \neg \Psi(\varphi, \omega, t, s) \\ \Psi(\varphi_1 \wedge \varphi_2, \omega, t, s) &= \Psi(\varphi_1, \omega, t, s) \wedge \Psi(\varphi_2, \omega, t, s) \\ \Psi(\varphi_1 U \varphi_2, \omega, t, s) &= \exists t' \in t+I: (\Psi(\varphi_2, \omega, t', s) \wedge \forall t'' \in [t, t']: \Psi(\varphi_1, \omega, t'', s)) \end{aligned}$$

该定性化语义可对复合服务运行过程中, 是否存在服务质量变化且超出约束范围, 或者其他约束不被满足等问题, 给出形式化语义结果, 即可满足 ( $\top$ ), 或不满足 ( $\perp$ ). 对于示例 2 中的示例约束, 可对其进行定性化语义解释: 给定物联网服务  $es$  在时间区间  $I = [t_0, t_5] = \{t_0, t_1, \dots, t_5\}$  内吞吐量的信号值, 如图 3 所示, 即  $\pi_{tp}(\omega) = \{(t_0, 15), (t_1, 10), (t_2, 15), (t_3, 5), (t_4, 20), (t_5, 15)\}$ . 对于示例 2 中约束  $\mathcal{QC}_I(es)=G_I \neg(es.tp < 10)$  的监测需求, 可发现该约束不能被满足, 即存在时间点  $\exists t' \in I$  (如  $t'=t_3$ ) 使得  $\Psi(\neg(es.tp < 10), \omega, t', es) = \perp$ . 相反, 对于示例 3 中另一约束  $\mathcal{QC}'_I(es)=G_I \neg(es.tp < thd_d)$  的监测需求, 其中,  $thd_d = thd_{I_i}$ , 如图 3 所示, 可发现该约束在时间区间  $I$  内总是被满足, 即  $\forall t' \in I, \Psi(\neg(es.tp < thd_d), \omega, t', es) = \top$ .

上述定性化语义可判定多维信号对于用户指定约束范式的满足关系, 然而, 这种定性判断可能在实际监测中信息量不够, 更可取的是对满足或违反的程度做定量判断. 因此, 信号时序逻辑及其拓展工作通过配备定量语义来解决这个问题. 如第 2 节中所讨论的, 目前的研究工作在监测复合服务及其带有自适应参数的约束监测方面仍存缺陷, 并且缺乏形式化的解释与验证. 为了解决这个问题, 本文提出了一种新型的信号时序逻辑, 即 CSTL, 通过使用组合模式和动态参数扩展现有 STL, 对服务自身约束、服务之间依赖, 以及复合服务整体约束的定性和定量满意度进行形式化度量.

具体而言, CSTL 改进了 STL 的传统鲁棒性度量方式, 通过正负偏置的黎曼和算子分别累积满足和违反部分的鲁棒度值. 其中, 正负偏置黎曼和算子可定义如下.

**定义 10 (正负偏置的黎曼和算子).** 一个正偏置的黎曼和算子可由变量  $\tau$  中正数部分进行偏置运算:

$$R_I^{\geq 0}(\tau) = \begin{cases} \sum_{i=1}^k \frac{\Delta I_i}{2} \times (\sigma^{\geq 0}(\tau_{t_i}) + \sigma^{\geq 0}(\tau_{t_{i+1}})), & \text{if } \exists \tau_{t_i} > 0 \\ \sum_{i=1}^k \frac{\Delta I_i}{2} \times (\tau_{t_i} + \tau_{t_{i+1}}), & \text{otherwise} \end{cases}$$

其中,  $I = \{I_1, I_2, \dots, I_k\}$  是时间区间  $I$  的划分; 一个划分后的时间区间  $I_i = [t_i, t_{i+1}]$ , 其中  $(i = 1, 2, \dots, k \text{ 且 } k \geq 1)$ , 且  $I_i \cap I_{i+1} = \{t_{i+1}\}$ ; 带有采样时间信息的变量  $\tau = \{\tau_{t_1}, \tau_{t_2}, \dots, \tau_{t_{k+1}}\}$ , 其中  $\tau_{t_i}$  代表时间点  $t_i$  下的变量值;  $\sigma^{\geq 0}(\tau_{t_i}) = \max(0, \tau_{t_i}) \geq 0$ ; 在此, 通过梯形法计算  $\tau$  的积分近似值, 求得黎曼和结果.

类似地, 负偏置黎曼和算子可定义如下, 或可由正偏置黎曼和算子进行推导 (即  $R_I^{\leq 0}(\tau) = -R_I^{\geq 0}(-\tau)$ ):

$$R_i^{\leq 0}(\tau) = \begin{cases} \sum_{i=1}^k \frac{\Delta I_i}{2} \times (\sigma^{\leq 0}(\tau_i) + \sigma^{\leq 0}(\tau_{i+1})), & \text{if } \exists \tau_i < 0 \\ \sum_{i=1}^k \frac{\Delta I_i}{2} \times (\tau_i + \tau_{i+1}), & \text{otherwise} \end{cases}$$

对于没有采样时间信息的变量  $\tau$ , 即  $\tau = \{\tau_1, \tau_2, \dots, \tau_{k+1}\}$ , 可定义正偏置黎曼和算子为:

$$R^{\geq 0}(\tau) = \begin{cases} \sum_{i=1}^{k+1} \sigma^{\geq 0}(\tau_i), & \text{if } \exists \tau_i > 0 \\ \sum_{i=1}^k \tau_i, & \text{otherwise} \end{cases} \quad (9)$$

类似地, 可对负黎曼和算子进行推导, 即  $R^{\leq 0}(\tau) = -R^{\geq 0}(-\tau)$ .

在实际的监测过程中, 执行黎曼求和运算的时间复杂度呈线性, 即  $O(|\tau|)$ . 其中,  $|\tau|$  是  $\tau$  中变量的总数量, 即  $|\tau| = k+1$ . 在实际执行过程中, 该黎曼和算子的时间复杂度在每个时间点均为  $O(1)$ , 这是因为在实际运行时, 只需将新到达采样数据或变量值进行求积运算, 再累加到上一时刻的累加和上. 因此, 黎曼和算子对于像物联网这种能量有限的网络是节能高效的, 不会耗费过多能量或资源. 与黎曼和算子时间复杂度类似, 执行正负偏置的黎曼和算子仅需常数量级, 即  $O(1)$ , 即可实现其累加型计算处理. 因此, 本文所提出的黎曼和算子, 及其正负偏置类型算子, 耗能均在可接受范围内, 可适用于物联网服务监测场景.

基于上述构建的正负偏置黎曼和算子, 可提出定量化语义函数, 即  $\eta(\varphi, \omega, t, s)$ , 也称为聚合型鲁棒性定量度量函数, 其定义如下.

**定义 11 (聚合型定量化鲁棒值).** 一个鲁棒性定量度量函数  $\eta(\varphi, \omega, t, s): \Phi \times \mathcal{W} \times I \times \mathcal{S} \rightarrow \mathbb{R}$  可将给定的一个信号时序逻辑公式  $\varphi \in \mathcal{F}$ , 一个信号值  $\omega \in \mathcal{W}$ , 在一个时间点  $t \in I$ , 对于一个物联网服务或复合服务  $s \in \mathcal{S}$ , 映射到  $\mathbb{R}$  内的一个实数值:

$$\begin{aligned} \eta(v \sim c, \omega, t, es) &= c - \pi_x(\omega)[t, es] \\ \eta(v \sim \xi, \omega, t, es) &= \xi - \pi_x(\omega)[t, es] \\ \eta(\mathcal{C}_S^{\mathcal{F}} \mathcal{V} \sim \xi, \omega, t, cs) &= \xi - \pi_x(\omega)[t, cs] \\ \eta(\neg \varphi, \omega, t, s) &= -\eta(\varphi, \omega, t, s) \\ \eta(\varphi_1 \wedge \varphi_2, \omega, t, s) &= R^{\leq 0}\{\eta(\varphi_1, \omega, t, s), \eta(\varphi_2, \omega, t, s)\} \\ \eta(\varphi_1 U_I \varphi_2, \omega, t, s) &= R_{t+I}^{\geq 0}\{R^{\leq 0}\{\eta(\varphi_2, \omega, t', s), R_{[t,t']}^{\leq 0}\eta(\varphi_1, \omega, t', s)\}\} \end{aligned}$$

基于该鲁棒值函数, 可定量化复合服务运行态满足度, 即给出具体的约束满足 ( $\top$ ) 或不满足 ( $\perp$ ) 度数值, 实现复合服务的定量监测, 如定义 7 所示  $\eta_i(s)$ . 当一个服务  $s$  上所指定的 CSTL 公式  $\varphi$  可由一个信号值  $\omega$  在时间点  $t$  满足 (即  $\Psi(\varphi, \omega, t, s) = \top$ ), 当且仅当  $\eta(\varphi, \omega, t, s) > 0$ . 例如, 给定图 3 中所示信号值, 即物联网服务  $es$  在时间区间  $I$  内吞吐量的信号值  $\pi_p(\omega)$ , 可定量化计算示例约束  $QC_I(es) = G_I \neg(es.tp < 10)$  的违反度 (由上文可知该约束未被满足), 即  $\eta(G_I \neg(es.tp < 10), \omega, t, es) = -2.5$ . 对于另一约束  $QC'_I(es) = G_I \neg(es.tp < thd_d)$ , 该约束的定量满足度为  $\eta(G_I \neg(es.tp < thd_d), \omega, t, es) = 37.5$  (由上文可知该约束可被满足).

与传统信号时序逻辑 STL<sup>[36]</sup> 的鲁棒性度量  $\rho(\varphi, \omega, t, s)$  相比较, 本文所提出的聚合型鲁棒性度量方式  $\eta(\varphi, \omega, t, s)$  由于在 STL 上对复合服务的监测需求进行适应性扩展, 并针对动态约束进行参数自适应调整, 将更加的适用于本文所提的应用场景. 在此, 可对其可靠性 (soundness) 和正确性 (correctness) 定理进行证明.

**定理 1 (可靠性 (soundness)).** 所提出的聚合型鲁棒性度量方法是可靠的 (sound), 当传统信号时序逻辑中常量值满足特定约束, 即具有严格正鲁棒度值的信号满足规范, 而具有严格负鲁棒度值的信号则违反规范:

$$\begin{cases} \eta(\varphi, \omega, t, s) > 0 \text{ and } c \geq \max(\xi) \Rightarrow \rho(\varphi, \omega, t, s) > 0 \\ \eta(\varphi, \omega, t, s) < 0 \text{ and } c \leq \min(\xi) \Rightarrow \rho(\varphi, \omega, t, s) < 0. \end{cases}$$

证明: 可根据结构化的归纳法推导证明该可靠性定理. 首先, 可证明基本范例是可靠的, 即对于  $\varphi := v \sim c$ , 可直观上推导出  $\eta(v \sim c, \omega, t, s) = \rho(v \sim c, \omega, t, s)$ , 即  $\eta(\varphi, \omega, t, s) > 0 \Leftrightarrow \rho(\varphi, \omega, t, s) > 0$ . 其次, 对于该聚合型信号时序逻辑

辑中所扩展的带有变量  $\xi$  的范例, 即  $\varphi := v \sim \xi$ , 由于 STL 中没有该范例, 可将 CSTL 中  $\varphi := v \sim \xi$  与 STL 中  $\varphi := v \sim c$  进行比对证明其可靠性如下: ① 当  $c \geq \max(\xi)$ , 即  $c \geq \xi$ , 可推导  $\rho(\varphi, \omega, t, s) \geq \eta(\varphi, \omega, t, s)$ , 因此,  $\eta(\varphi, \omega, t, s) > 0 \Rightarrow \rho(\varphi, \omega, t, s) > 0$ ; ② 类似地, 当  $c \leq \min(\xi) \leq \xi$ , 此时  $\eta(\varphi, \omega, t, s) \geq \rho(\varphi, \omega, t, s)$ , 由此可得  $\eta(\varphi, \omega, t, s) < 0 \Rightarrow \rho(\varphi, \omega, t, s) < 0$ . 然后, 可类似地推导该聚合型信号时序逻辑中所扩展范例  $\varphi := C_S^f \mathcal{V} \sim \xi$  的可靠性. 最后, 对于不同逻辑算子和时序算子下公式范例的可靠性, 可仅对其算子所修饰的子公式的可靠性进行证明即可. 在此, 以时序算子总是满足为例, 当  $\eta(G_I \varphi, \omega, t, s) > 0$ , 即  $\forall t' \in I, \eta(\varphi, \omega, t', s) > 0$ , 由此,  $\rho(G_I \varphi, \omega, t, s) = \min\{\rho(\varphi, \omega, t, s)\} > \min\{\eta(\varphi, \omega, t, s)\} > 0$ , 当  $c \geq \max(\xi)$ ; 相反地, 当  $\eta(G_I \varphi, \omega, t, s) < 0$ , 即存在  $t' \in I$  使得  $\eta(\varphi, \omega, t', s) < 0$ , 即可推导出  $\rho(\varphi, \omega, t, s) < 0, \rho(G_I \varphi, \omega, t, s) = \min\{\rho(\varphi, \omega, t, s)\} < 0$ , 因此可证明  $\eta(G_I \varphi, \omega, t, s) < 0 \Rightarrow \rho(G_I \varphi, \omega, t, s)$ .

**定理 2 (正确性 (correctness)).** 由鲁棒性定量化度量值可指示定性化满足性如下:

$$\begin{cases} \eta(\varphi, \omega, t, s) > 0 \Rightarrow \Psi(\varphi, \omega, t, s) = \top \\ \eta(\varphi, \omega, t, s) < 0 \Rightarrow \Psi(\varphi, \omega, t, s) = \perp \end{cases}$$

证明: 首先, 可证明基本范例是正确的, 即对于  $\varphi := v \sim c$ , 可直观上推导出  $\eta(v \sim c, \omega, t, s) > 0 \Rightarrow \Psi(v \sim c, \omega, t, s) = \top$ . 对于迭代组合的公式范例, 可在其子公式满足该正确性定理时, 证明该公式范例同样满足. 因此, 可分别证明不同逻辑算子和时序算子下公式范例的正确性. 以时序算子总是满足为例, 当  $\eta(G_I \varphi, \omega, t, s) > 0$ , 即  $\forall t' \in I, \eta(\varphi, \omega, t', s) > 0$ , 则  $\Psi(\varphi, \omega, t', s) = \top$ , 由此  $\Psi(G_I \varphi, \omega, t, s) = \top$ , 即  $\eta(G_I \varphi, \omega, t, s) > 0 \Rightarrow \Psi(G_I \varphi, \omega, t, s) = \top$ .

## 5 基于 CSTL 的复合服务联动定量监测

本节对基于 CSTL 的复合服务联动定量监测进行详细叙述, 即将用户指定的监测约束转换为 CSTL 公式, 并在运行时使用定性和定量语义解释其实时监测结果.

### 5.1 基于 CSTL 语法的复合服务监测规约

如第 3 节所介绍, 物联网服务及其复合服务在其执行期间可能存在时间约束和 QoS 约束, 这些约束指定于其服务自身、服务之间, 以及复合服务上, 并在物联网服务运行过程中持续监测, 以确保单个物联网服务或复合服务整体满足用户需求. 具体而言, 可通过以下 CSTL 公式来制定规定约束的监测.

服务自身实时监测: 可对物联网服务自身所指定的各项时间约束和 QoS 约束进行实时监测, 如第 3.1.1 节和第 3.2.1 节所示. 具体包括: ① 开始/结束时间和持续时间的限制, 如定义 2 中所述, 以及 ② 特定时间区间的 QoS 约束, 如定义 5 所述.

- 服务自身时间约束  $\mathcal{TC}(es) = (ita, cnt, thd)$  可被表示为:

$$\varphi_{ic} := es.ita \sim thd \mid \neg \varphi_{ic} \mid \varphi_{ic_1} \wedge \varphi_{ic_2} \quad (10)$$

其中,  $thd := thd_f \mid thd_d$  代表固定或动态可变阈值, 例如 CSTL 语法中的  $c$  或  $\xi$ .

- 服务自身 QoS 约束  $\mathcal{QC}_I(es) = (ita, cnt, thd)$  可被转换为如下公式:

$$\varphi_{qc} := es.atr \sim thd \mid \neg \varphi_{qc} \mid \varphi_{qc_1} \wedge \varphi_{qc_2} \mid \varphi_{qc_1} U_I \varphi_{qc_2} \quad (11)$$

服务之间联动监测: 可对物联网服务之间的时间依赖和 QoS 依赖进行规约, 实现服务之间的联动监测, 如第 3.1.2 节和第 3.2.2 节所示. 具体包括: ① 服务之间时间依赖, 如定义 3 所述, 以及 ② QoS 依赖, 如定义 6 所述.

- 给定一时间依赖约束  $\mathcal{TD}(es_i, es_j) = (itr, cnt, thd)$ , 可将其转换为如下 CSTL 公式:

$$\varphi_{id} := itr \sim thd \mid \neg \varphi_{id} \mid \varphi_{id_1} \wedge \varphi_{id_2} \quad (12)$$

- 对于一个 QoS 依赖约束  $\mathcal{QD}(es_i, es_j) = (\mathcal{QC}_{I_1}(es_i), \mathcal{QC}_I(es_j), itr, cnt, thd)$ , 可指定如下 CSTL 公式来表示该约束:

$$\varphi_{qd} := \varphi_{qc}^{es_i} \wedge \varphi_{qc}^{es_j} \wedge \varphi_{qd} \quad (13)$$

其中,  $\varphi_{qc}^{es_i}$  和  $\varphi_{qc}^{es_j}$  分别代表物联网服务  $es_i$  和  $es_j$  上的 QoS 约束, 如公式 (11) 所示.  $\varphi_{qd} := itr \sim thd \mid \neg \varphi_{qd} \mid \varphi_{qd_1} \wedge \varphi_{qd_2}$ .

复合服务整体监测: 此外, 可对物联网服务组合形成的复合服务进行整体监测, 以实时判定其是否满足其上时间约束和 QoS 约束, 如第 3.1.3 节和第 3.2.3 节所示. 具体而言, 复合服务的监测涉及复合服务的时间和 QoS 约束,



可用 CSTL 公式表示这些约束如下:

$$\varphi^{cs} := \varphi_{ic}^{cs} \mid \varphi_{qc}^{cs} \mid \neg\varphi^{cs} \mid \varphi^{cs_1} \wedge \varphi^{cs_2} \mid \varphi^{cs_1} U_I \varphi^{cs_2} \quad (14)$$

其中,  $\varphi_{ic}^{cs} := +cs.ita \sim thd$ ,  $\varphi_{qc}^{cs} := C_S^f \mathcal{V} \sim thd$ .

为实现上述服务自身、服务之间以及复合服务的多项约束监测, 本文提出了自适应监测机制, 即在物联网服务监测期间其监测参数可随时间发生改变. 如上述公式 (10)–公式 (14) 中阈值参数  $thd$  可根据实际动态边缘资源, 或相关依赖服务, 进行自适应调整. 在第 6 节实验中, 动态阈值被记作  $thd_d$ .

由此, 为满足用户对于物联网服务及其复合服务运行态的自适应监测需求, 及时有效发现 QoS 降低或时间违反情况, 可实现基于 CSTL 的复合服务联动定量监测. 服务监测约束可首先利用 CSTL 语法表示为 CSTL 公式, 具体运行实例详见第 6 节. 对于这些 CSTL 公式, 可再利用 CSTL 定性及定量化语义进行实时监测与形式化解释.

## 5.2 基于 CSTL 语义的联动定量监测

在将各种服务约束转换为 CSTL 公式后, 可在运行时利用 CSTL 语义来定性和定量地评估约束满意度. 如第 3.3 节中所述,  $\Psi_{t_i}(s_j)$  和  $\eta_{t_i}(s_j)$  被形式化解释, 即利用定义 9 中的定性语义和定义 11 中的定量语义进行复合服务联动定量监测. 算法 1 即给出了具体的基于 CSTL 的复合服务联动定量监测过程, 其中多种不同服务请求监测以一种迭代方式进行解析. 首先, 给定可满足用户需求的一个复合服务  $cs$ , 以及为保证该复合服务运行的各项时间约束和 QoS 约束 (行 1), 记作  $\mathcal{R}$ . 根据第 5.1 节所述监测规约过程, 可将各项约束转换为相应的 CSTL 公式, 记作  $\Phi_{\mathcal{R}}$  (行 2). 由此, 可对这些约束进行实时定量监测, 即在服务运行期间的每一时间点,  $\forall t_i \in I$ , 每一个 CSTL 公式 (即  $\forall \varphi_r \in \Phi_{\mathcal{R}}$ ) 可对其进行监测定性及定量化满足度 (行 3, 4), 即  $\Psi_{t_i}(cs, \varphi_r)$  和  $\eta_{t_i}(cs, \varphi_r)$ .

---

**算法 1.** 基于聚合型信号时序逻辑的复合服务联动定量监测算法.

---

输入:  $cs$ : 可满足物联网应用的复合服务;

输出:  $\Psi_{t_i}(cs)$ : 区间  $I$  中每一时刻  $t_i$  下复合服务  $cs$  的定量化满足度.

---

$\eta_{t_i}(cs)$ : 区间  $I$  中每一时刻  $t_i$  下复合服务  $cs$  的定量化满足度.

1.  $\mathcal{R} \leftarrow \{TC(s_j) \cup QC_I(s_j) \cup TD(s_j, s_k) \cup QD(s_j, s_k) \mid s_j \in \mathcal{S}, s_k \in \mathcal{S}\}$
  2.  $\Phi_{\mathcal{R}} \leftarrow$  一组由公式 (10)–公式 (14) 指定的用来表示约束集  $\mathcal{R}$  的 CSTL 公式
  3. **while**  $t_i \in I$  and  $\Phi_{\mathcal{R}} \neq \emptyset$  **do**
  4.   **switch**  $\forall \varphi_r \in \Phi_{\mathcal{R}}$  :
  5.     **case**  $\varphi_1 \wedge \varphi_2$  **do**
  6.        $\Psi_{t_i}(cs, \varphi_r) \leftarrow \Psi_{t_i}(cs, \varphi_1) \wedge \Psi_{t_i}(cs, \varphi_2)$
  7.        $\eta_{t_i}(cs, \varphi_r) \leftarrow R^{\leq 0} \{\eta_{t_i}(cs, \varphi_1), \eta_{t_i}(cs, \varphi_2)\}$
  8.     **case**  $\varphi_1 U_{I_i} \varphi_2$  **do**
  9.        $\Psi_{t_i}(cs, \varphi_r) \leftarrow \exists t' \in t + I_i : (\Psi(\varphi_2, \omega, t', s) \wedge \forall t'' \in [t, t'] : \Psi(\varphi_1, \omega, t'', s))$
  10.        $\eta_{t_i}(cs, \varphi_r) \leftarrow R_{t+I_i}^{\geq 0} \left( R^{\leq 0} \left\{ \eta(\varphi_2, \omega, t', s), R_{[t, t']}^{\leq 0} \eta(\varphi_1, \omega, t'', s) \right\} \right)$
  11.     **case**  $\neg\varphi$  **do**
  12.        $\Psi_{t_i}(cs, \varphi_r) \leftarrow \neg\Psi_{t_i}(cs, \varphi_r)$ ,  $\eta_{t_i}(cs, \varphi_r) \leftarrow -\eta_{t_i}(cs, \varphi_r)$
  13.     **case**  $v \sim c$  **do**
  14.        $\Psi_{t_i}(cs, \varphi_r) \leftarrow \pi_x(\omega)[t, es] \sim c$ ,  $\eta_{t_i}(cs, \varphi_r) \leftarrow c - \pi_x(\omega)[t, es]$
  15.     **case**  $v \sim \xi$  **do**
  16.        $\Psi_{t_i}(cs, \varphi_r) \leftarrow \pi_x(\omega)[t, es] \sim \xi$ ,  $\eta_{t_i}(cs, \varphi_r) \leftarrow \xi - \pi_x(\omega)[t, es]$
  17.     **case**  $C_S^f \mathcal{V} \sim \xi$  **do**
  18.        $\Psi_{t_i}(cs, \varphi_r) \leftarrow \pi_x(\omega)[t, cs] \sim \xi$ ,  $\eta_{t_i}(cs, \varphi_r) \leftarrow \xi - \pi_x(\omega)[t, cs]$
-

19. **end switch**

20. **return**  $\Psi_{t_i}(cs) \leftarrow \bigwedge \{\forall \Psi_{t_i}(s, \varphi_r)\}, \eta_{t_i}(cs) \leftarrow R^{\leq 0}\{\forall \eta_{t_i}(s, \varphi_r)\}$

21. **end while**

一个 CSTL 公式可由多个子公式通过逻辑并算子结合, 即  $\varphi_r := \varphi_1 \wedge \varphi_2$  (行 5), 如第 5.1 节所述多个时间约束或 QoS 约束 (如公式 (10) 中  $\varphi_{rc} := \varphi_{rc_1} \wedge \varphi_{rc_2}$ ). 针对该复合约束, 其定性化满足性由其子约束满足性决定, 即判定是否每一个子公式均可满足 (即  $\Psi_{t_i}(cs, \varphi_1)$  和  $\Psi_{t_i}(cs, \varphi_2)$ ), 并取其逻辑并的结果 (行 6); 量化满足度则由相应的量化语义决定, 即负偏置下的黎曼和运算 (行 7).

考虑到一段时间段内 (即  $I_{t_i}$ ) 的约束, 即  $\varphi_r := \varphi_1 U_{I_{t_i}} \varphi_2$  (行 8), 其中,  $I_{t_i}$  表示与当前时间点  $t_i$  相关的时间区间, 可根据 CSTL 定性化及量化语义分别求得  $\Psi_{t_i}(cs, \varphi_r)$  和  $\eta_{t_i}(cs, \varphi_r)$  (行 9, 10). 对于逻辑非规约下的 CSTL 公式, 即  $\varphi_r := \neg \phi$  (行 11), 其定性化满足性即为  $\Psi_{t_i}(cs, \varphi_r)$  的逻辑非结果, 量化满足度即为  $\eta_{t_i}(cs, \varphi_r)$  取反的结果 (行 12). 对于基础规约形式  $\varphi_r := v \sim c$  (行 13), 其  $\Psi_{t_i}(cs, \varphi_r)$  结果为变量值  $\pi_x(\omega)[t, es]$  与  $c$  的逻辑比较结果,  $\eta_{t_i}(cs, \varphi_r)$  为相应变量值与  $c$  的差值 (行 14).

此外, 对于扩展的逻辑范式  $v \sim \xi$ , 其定性化与量化满足度与  $\varphi_r := v \sim c$  类似 (行 15, 16); 对于复合服务下的约束形式  $\varphi_r := C_S^f \mathcal{V} \sim \xi$  (行 17), 其  $\Psi_{t_i}(cs, \varphi_r)$  结果为该聚合算子下的复合服务聚合变量值  $\pi_x(\omega)[t, cs]$  (如公式 (5) 所示) 与动态变量  $\xi$  之间的逻辑比较结果,  $\eta_{t_i}(cs, \varphi_r)$  为相应变量值与  $\xi$  的差值 (行 18). 由此, 针对  $\forall \phi_r \in \varphi_R$ , 该复合服务的定性化满足性由所有  $\Psi_{t_i}(cs, \varphi_r)$  决定, 即  $\Psi_{t_i}(cs) \leftarrow \bigwedge \{\forall \Psi_{t_i}(s, \varphi_r)\}$ , 取  $\{\forall \Psi_{t_i}(s, \varphi_r)\}$  的逻辑并结果; 量化满足度则由负偏置的黎曼和算子求得, 即  $\eta_{t_i}(cs) \leftarrow R^{\leq 0}\{\forall \eta_{t_i}(s, \varphi_r)\}$ .

基于信号时序逻辑的监测算法时间复杂度与所需监测约束规约后生成的 CSTL 公式和监测时长相关. 其中, 包含逻辑算子  $\wedge$ 、 $\neg$ 、 $v \sim c$  与  $v \sim \xi$  的 CSTL 公式在实际监测中的复杂度, 与原本信号时序逻辑所述复杂度相同, 即正比于监测过程  $I$  中所采样的信号值个数 (记作  $|I|$ ), 即  $O(|I|)$ ; 类似地, 包含时序算子  $U_I$  的 CSTL 公式在实际监测中的复杂度为  $O(|I|^2)$ , 其中基于正负偏置黎曼和的实时运算复杂度为  $O(1)$ ; 包含本文所提出聚合算子  $C_S^f \mathcal{V} \sim \xi$  的 CSTL 公式在实际监测中的复杂度与复合服务所包含的物联网服务个数 (记作  $|\mathcal{L}(cs)|$ ) 和  $|I|$  相关, 即  $O(|\mathcal{L}(cs)| \times |I|)$ . 具体的运行实例及其监测结果可参见第 6 节.

## 6 实验分析

本文使用 Python 语言开发实验, 来模拟实现复合服务联动定量监测. 为保证实验结果的可比较性, 均在同一台机器上运行, 机器为 64 位 Ubuntu 系统, 配置的处理为 AMD Ryzen 3800X 8-Core, CPU 为 3.9 GHz, 内存为 16 GB. 具体实验数据 (第 6.1 节)、对比方法 (第 6.2 节)、评价指标 (第 6.3 节)、实验设置 (第 6.4 节), 以及实验结果 (第 6.5 节) 说明如下.

### 6.1 实验数据

在此采用真实服务日志和服务 QoS 数据集两个公开可用的数据集进行复合服务联动定量监测的性能评估. 其中, 真实服务日志数据集包括如下 3 个服务执行日志, 具体如表 2 所示.

① 真实服务日志: 在此可基于真实日志数据进行实验分析, 即 4TU Centre for Research Data (<http://data.4tu.nl>) 公开数据网站中真实服务过程产生的服务执行日志. 该日志均包含服务执行的开始时间 (即日志中 Start 标签所表示时间) 和结束时间 (即日志中 Complete 标签所表示时间), 可用于时间约束的模拟监测. 具体包括以下内容.

- 采购流程日志 (purchasing process event logs, PUC): 是由一个购买到支付过程生成的服务执行日志<sup>[43]</sup>, 该日志由信息系统实时记录生成, 被用作本文的应用示例, 如第 2 节所示. 采购流程日志 PUC 可详见网址 <http://fluxicon.com/>

- 生产分析日志 (production analysis process data, PRD): 是由一个生产制造过程生成的服务执行日志<sup>[45]</sup>, 由 ERP 或 MES 系统记录并导出. 生产分析日志 PRD 可详见网址 <https://doi.org/10.4121/uuid:68726926-5ac5-4fab->

b873-ee76ea412399

• 日常活动日志 (activities of daily living of several individuals logs): 具体包括 8 个服务执行日志 (简称 AUL, AUW, ..., E110W 等), 代表不同参与者在工作日和休息日通过物联网设备 (如不同传感器) 记录到的不同生活场景中的日常活动, 例如睡眠、做饭、洗漱等. 日常活动日志可详见网址 <https://doi.org/10.4121/uuid:01eaba9f-d3ed-4e04-9945-b8b302764176>

表 2 服务日志数据集统计详情表

服务执行日志 名称 (及简称)	总案例 个数	服务调用 次数	服务类 个数	案例长度			
				最小值	平均值	最大值	
采购流程日志 (PUC)	608	9119	21	2	15	44	
生产分析日志 (PRD)	225	4543	45	1	20	175	
日常 活动 日志	AUL	25	1392	32	34	56	92
	AUW	10	488	30	28	49	70
	E102L	18	1152	36	46	64	82
	E102W	7	420	36	46	60	76
	E104L	43	4200	38	58	98	134
	E103W	18	1728	38	56	96	124
	E110L	21	1390	34	20	66	90
	E110W	6	368	28	54	61	66

② 服务 QoS 数据集: 该数据集提供 Web 服务的分布式可靠性评估机制 (distributed reliability assessment mechanism for Web services, 由此该数据集被简称为 WS-DREAM), 允许用户进行多种服务质量评估<sup>[46]</sup>. 具体数据集是由 142 个用户在 64 个时间片中以 15 min 的时间间隔对 4532 个分布式 Web 服务进行 30287611 次服务调用产生, 包括了该 4532 个服务及其 30287611 次执行过程中的 QoS 数值变化, 即响应时间 (response-time (s)) 和吞吐量 (throughput (Kb/s)). WS-DREAM 数据集可详见网址 <http://www.inpluslab.com/wsdream/>

实际数据收集所占用的资源与需要的时间可能受多方面因素的约束, 该问题在传统的传感器网络和物联网领域已有较多工作开展了数据采集过程及性能优化研究, 包括本人前期相关的研究成果<sup>[47, 48]</sup>, 为本论文服务运行时定量监测策略研究提供了高效可靠的数据支撑. 在本论文实验评估中, 所采用的数据集可视为实时数据, 即以实时到达的方式验证物联网服务的实时监测效果. 其中, 真实服务日志被用于物联网服务时间约束的模拟监测, 服务 QoS 数据集 WS-DREAM 被用于服务 QoS 约束的监测. 具体运行实例可参见表 3, 即物联网服务自身、服务之间以及复合服务上的时间约束和 QoS 约束监测.

表 3 复合服务联动定量监测运行实例表

约束	不同约束描述	CSTL公式
服务 自身 时间 约束	服务执行日志PUC中物联网服务 $es_0$ , 即“Create Purchase Requisition”, 如图2所示, 总是开始时间不早于 $thd_f^{st}$ (或 $thd_d^{st}$ )	$\varphi_1 = G_I \neg (es_0.st < thd_f^{st})$ $\varphi'_1 = G_I \neg (es_0.st < thd_d^{st})$
	$es_0$ 总是结束时间不晚于 $thd_f^{ft}$ (或 $thd_d^{ft}$ )	$\varphi_2 = G_I (es_0.ft \leq thd_f^{ft})$ $\varphi'_2 = G_I (es_0.ft \leq thd_d^{ft})$
	$es_0$ 总是耗时不超过 $thd_f^{dur}$ (或 $thd_d^{dur}$ )	$\varphi_3 = G_I (es_0.dur \leq thd_f^{dur})$ $\varphi'_3 = G_I (es_0.dur \leq thd_d^{dur})$
	$es_0$ 总是开始时间不早于 $thd_f^{st}$ (或 $thd_d^{st}$ ), 并且结束时间不晚于 $thd_f^{ft}$ (或 $thd_d^{ft}$ )	$\varphi_4 = \varphi_1 \wedge \varphi_2$ $\varphi'_4 = \varphi'_1 \wedge \varphi'_2$
	$es_0$ 总是开始时间不早于 $thd_f^{st}$ (或 $thd_d^{st}$ ), 并且耗时不超过 $thd_f^{dur}$ (或 $thd_d^{dur}$ )	$\varphi_5 = \varphi_1 \wedge \varphi_4$ $\varphi'_5 = \varphi'_1 \wedge \varphi'_4$
	$es_0$ 总是耗时不超过 $thd_f^{dur}$ (或 $thd_d^{dur}$ ), 并且结束时间不晚于 $thd_f^{ft}$ (或 $thd_d^{ft}$ )	$\varphi_6 = \varphi_2 \wedge \varphi_4$ $\varphi'_6 = \varphi'_2 \wedge \varphi'_4$

表3 复合服务联动定量监测运行实例表(续)

约束	不同约束描述	CSTL公式	
服务自身 QoS 约束	WS-DREAM数据集中以第1个物联网服务为例, 记作 $es_0$ , 其响应时间总是不大于 $thd_f^{rt}$ (或 $thd_d^{rt}$ )	$\varphi_7 = G_I(es_0.rt \leq thd_f^{rt})$ $\varphi_7' = G_I(es_0.rt \leq thd_d^{rt})$	
	$es_0$ 总是吞吐量不小于 $thd_f^{tp}$ (或 $thd_d^{tp}$ )	$\varphi_8 = G_I \neg (es_0.tp < thd_f^{tp})$ $\varphi_8' = G_I \neg (es_0.tp < thd_d^{tp})$	
	WS-DREAM数据集中前50个物联网服务, 记作 $\{es_i\}_{i=1}^{50}$ , 其响应时间各不大于 $thd_f^{rt}$ (或 $thd_d^{rt}$ )	$\varphi_9 = G_I(es_i.rt \leq thd_f^{rt})$ $\varphi_9' = G_I(es_i.rt \leq thd_d^{rt})$	
	物联网服务 $es_i$ ( $i = \{1, 2, \dots, 50\}$ ) 吞吐量各不小于 $thd_f^{tp}$ (或 $thd_d^{tp}$ )	$\varphi_{10} = G_I \neg (es_i.tp < thd_f^{tp})$ $\varphi_{10}' = G_I \neg (es_i.tp < thd_d^{tp})$	
服务之间 时间 依赖	服务执行日志PRD中顺序执行的物联网服务“Final Inspection”与“Packing”, 记作 $es_3$ 与 $es_4$ , 服务之间时间间隔FS不大于12 h, 或不大于24 h, 当 $es_4$ 开始执行于 $es_3$ 结束之后	$\varphi_{11} = es_4.st - es_3.ft \leq 12$ $\varphi_{12} = es_4.st - es_3.ft \leq 24$ $\varphi_{13} = \varphi_{11} \vee ((\neg\varphi_{11}) \wedge \varphi_{12})$	
	$es_3$ 与 $es_4$ 之间FS值满足 $\varphi_{11}$ (或 $\varphi_{13}$ ), 即FS值不大于12 h (或不大于24 h), 并且SF值不大于24 h	$\varphi_{14} = es_4.ft - es_3.st \leq 24$ $\varphi_{15} = \varphi_{11} \wedge \varphi_{14}$ $\varphi_{15}' = \varphi_{13} \wedge \varphi_{14}$	
	并行执行的物联网服务“Flat Grinding”与“Lapping”, 记作 $es_6$ 与 $es_7$ , 服务之间FS值不大于5 h, 且SF值不超过15 h	$\varphi_{16} =  es_6.ft - es_7.st  \leq 5$ $\varphi_{17} = es_7.ft - es_6.st \leq 15$ $\varphi_{18} = \varphi_{16} \wedge \varphi_{17}$	
	$es_6$ 与 $es_7$ 之间FS值不大于5 h, 或不超过10 h; 且其SF值不大于15 h, 或不超过25 h	$\varphi_{19} =  es_6.ft - es_7.st  \leq 10$ $\varphi_{20} = \varphi_{16} \vee ((\neg\varphi_{16}) \wedge \varphi_{19})$ $\varphi_{21} = es_7.ft - es_6.st \leq 25$ $\varphi_{22} = \varphi_{17} \vee ((\neg\varphi_{17}) \wedge \varphi_{20})$ $\varphi_{23} = \varphi_{20} \wedge \varphi_{22}$	
	服务之间 QoS 依赖	WS-DREAM数据集中物联网服务 $es_0$ 在时间区间 $I_1 = t + [0, 5]$ 内存在时间点下的响应时间超过 $thd_f^{rt} = 4$ , 并在 $\Delta t \in [10, \Delta t_{\max}]$ 时间片后 (其中 $10 \leq \Delta t_{\max} \leq 30$ ), 在时间区间 $I_2 = t + 5 + \Delta t + [0, 5]$ 内总是不超过 $thd_f^{rt}$ (或 $thd_d^{rt} = 6$ )	$\varphi_{24} = F_{I_1} \neg (es_0.rt \leq thd_f^{rt})$ $\varphi_{25} = G_{I_2} (es_0.rt \leq thd_f^{rt})$ $\varphi_{25}' = G_{I_2} (es_0.rt \leq thd_d^{rt})$ $\varphi_{26} = \varphi_{24} U_{[10, \Delta t_{\max}]} \varphi_{25}$ $\varphi_{26}' = \varphi_{24} U_{[10, \Delta t_{\max}]} \varphi_{25}'$
		物联网服务 $es_0$ 在时间区间 $I_1$ 内吞吐量最终不超过 $thd_f^{tp} = 0.4$ , 并在 $\Delta t$ 时间片后, 在时间区间 $I_2$ 内能够总是超过 $thd_f^{tp}$ (或 $thd_d^{tp} = 0.3$ )	$\varphi_{27} = F_{I_1} (es_0.tp \leq thd_f^{tp})$ $\varphi_{28} = G_{I_2} \neg (es_0.tp \leq thd_f^{tp})$ $\varphi_{28}' = G_{I_2} \neg (es_0.tp \leq thd_d^{tp})$ $\varphi_{29} = \varphi_{27} U_{[10, \Delta t_{\max}]} \varphi_{28}$ $\varphi_{29}' = \varphi_{27} U_{[10, \Delta t_{\max}]} \varphi_{28}'$
WS-DREAM数据集中物联网服务 $es_0$ 和 $es_1$ 响应时间各不超过3 s		$\varphi_{30} = es_0.rt \leq 3$ $\varphi_{31} = es_1.rt \leq 3$	
由 $es_0$ 和 $es_1$ 以顺序结构组成的复合服务 $es_0 \rightarrow es_1$ , 其响应时间不超过6 s		$\varphi_{32} = C_S^F \{es_0.rt, es_1.rt\} \leq 6$	
由 $es_0$ 和 $es_1$ 以并行结构组成的复合服务 $es_0 \oplus es_1$ , 其响应时间不超过6 s		$\varphi_{33} = C_S^F \{es_0.rt, es_1.rt\} \leq 6$	
复合 服务 不同 约束	$es_0$ 响应时间不超过3 s, 且 $es_1$ 响应时间不超过3 s, 或不超过 $6 - es_0.rt$	$\varphi_{34} = es_1.rt \leq 6 - es_0.rt$ $\varphi_{35} = \varphi_{30} \wedge (\varphi_{31} \vee ((\neg\varphi_{31}) \wedge \varphi_{34}))$	
	$es_0$ 响应时间不超过6 s, 且 $es_1$ 响应时间不超过 $6 - es_0.rt$	$\varphi_{36} = es_0.rt \leq 6$ $\varphi_{37} = \varphi_{36} \wedge \varphi_{34}$	

## 6.2 对比方法

为了验证本文所提出技术方案的有效性和高效性, 拟与以下已有工作进行对比分析。

- 传统信号时序逻辑中鲁棒性 (traditional robustness in signal temporal logic, STL)<sup>[14-16, 36]</sup>: 为了展示相对于传统 STL<sup>[36]</sup>鲁棒性的改进, 在此可与其进行对比。STL 已在最近工作中被广泛运用 (如第1节所述), 例如, 采用 STL 进行 CPS 系统安全实时监测<sup>[14]</sup>、空间扩展下的城市监测<sup>[15]</sup>和物联网服务部署<sup>[16]</sup>等。

- 算术几何平均下的鲁棒性 (arithmetic-geometric mean robustness, AGM)<sup>[20]</sup>: 此外, 可与基于平均算子的鲁棒性进行比较, 例如最近提出一项基于算术几何平均的鲁棒性度量。

- 平滑累积下的鲁棒性 (smooth cumulative robustness, SC)<sup>[21]</sup>: 另外, 可与另一种广泛采用的改进鲁棒性的算子, 即平滑算子, 进行对比分析, 例如, 一种平滑累积语义下的鲁棒性。

- 加权鲁棒性 (weighted robustness of STL, wSTL)<sup>[49]</sup>: 目前被研究学者所提出的加权鲁棒性, 也是一种改进 STL 原有鲁棒性方式<sup>[50]</sup>, 并在不同场景中可设置不同权重值进行加权运算。

- 累加型鲁棒性 (accumulative STL, ASTL): 本文为克服现有信号时序逻辑存在的信号不可辨、方法不鲁棒及场景不适用等问题, 拓展构建了累加型鲁棒性度量方式。另一方面, 本文在其基础上扩展了聚合算子和动态变量以适应于复合服务的联动定量监测。为验证该两部分拓展工作的有效性, 在此可与仅拓展累加型鲁棒性度量的研究工作 (记作 ASTL) 进行对比。

这些对比方法均是具有代表性且与本文所构建 CSTL 策略相似的方法。具体来说, 针对基于平均算子和平滑算子的拓展型信号时序逻辑, AGM 为近年来最新运用平均算子改进信号时序逻辑的方法, 它对离散时间下采样到的信号值进行鲁棒性运算, 并定义了正满意度和负违规定下鲁棒性的不同处理算子, 与本文所提出的 CSTL 策略相似; SC 是近年来最贴近本文构建新型信号时序逻辑时采用不同算子实现满足部分和不满足部分鲁棒性度量方式的方法。此外, 考虑与最新基于加权和算子的信号时序逻辑 (即 wSTL) 进行对比。最后, 并与所提出的累加型鲁棒性度量方式进行对比, 验证本文在复合服务监测方面的适用性。在后续章节中, 将与这些对比方法进行对比实验及结果分析。从 CSTL 对于复合服务监测以及动态场景下自适应参数调整两个角度需求的有效满足, 证明本文在实际监测中有效性、适用性和鲁棒性。

### 6.3 评价指标

基于第 6.1 节所述的服务数据集, 从以下 3 个方面对提出的方法进行评估。

- (1) 提出的方法在时间约束和 QoS 约束监测方面的效果如何?
- (2) 提出的方法对服务自身、服务之间以及复合服务监测的效果如何?
- (3) 与现有的监测方法相比, 提出的方法表现如何?

为了有效评估上述问题, 本文选用物联网服务实时监测过程中的鲁棒性值作为性能指标进行评估。其中, 正值代表可满足用户需求, 负值代表违反用户需求, 更高的鲁棒性值表示监测任务获得了更高的满意度, 正如现有工作中所验证的结果<sup>[20,21]</sup>。在此, 基于真实服务日志的时间约束监测, 和基于服务 QoS 数据集 WS-DREAM 的 QoS 约束监测, 可评估所提出方法在时间约束和 QoS 约束监测方面的效果; 并考虑服务自身、服务之间和复合服务上的时间约束和 QoS 约束, 评估所提出方法的监测效果。与第 6.2 节所述相关方法的对比结果, 可验证本文方法的监测有效性、场景适用性和结果鲁棒性。

### 6.4 实验设置

实验设置参数详情如表 4 所示。具体设置说明如下。基于第 6.1 节所述公开数据集 WS-DREAM, 可模拟部署 4532 个物联网服务, 并模拟其 QoS 约束的实时监测; 或从服务执行日志中挖掘服务实际执行情况, 并模拟时间约束的不同监测。其中, 所模拟监测的时间约束和 QoS 约束的固定阈值或动态变量可设置为表 4 中所示不同参数; 在此设置了固定阈值和随机阈值, 对时间约束和 QoS 约束监测情况分别进行对比实验, 进而说明本文所提方法对于动态阈值或非固定阈值的监测有效性; 时间约束阈值, 例如执行时长不超过一定阈值, 可设置为固定值 (如 40 min, 如有需要也可设置为其他合适的数值) 或随机模拟; QoS 约束阈值, 例如吞吐量约束阈值, 可模拟为全部服务满足约束、存在部分服务不满足约束, 并反映阈值大小对于服务监测结果的影响; 考虑到 wSTL 方法对于不同信号值的权重设置, 在此考虑衰减系数权重参数设置方式, 即  $w = \{1, \alpha^1, \alpha^2, \dots, \alpha^{|\tau|-1}\}$ , 其中,  $\alpha$  是 0-1 之间的衰减系数 (本实验中取值 0.9, 如有需要可设置为其他数值;  $\alpha$  越小, 表示衰减速率越大),  $|\tau|$  表示变量  $\tau$  中信号个数。更多其他合适权重可根据用户需求进行设置。其他对比方法权重可以假定为  $\{1\}^{|\tau|}$ , 并在实验中通过缩放权重使得不同方法的总权重相同, 即  $w = w \cdot \text{sum}(\{1\}^{|\tau|}) / \text{sum}(w)$ 。

表 4 实验设置参数表

参数名	参数值	参数名	参数值
物联网服务个数	4532	物联网服务调用次数	30287611
用户个数	142	服务执行时间片个数	64
单个时间片时长	15 min	开始时间不早于特定阈值	4:00, 9:00
结束时间不晚于特定阈值	17:00, 22:00	执行时长不超过固定阈值	40 min
执行时长不超过随机阈值	30-50 min	响应时间约束阈值	3.3, 4, 6.7
吞吐量约束阈值	0.3, 0.5, 0.6	wSTL权重中衰减系数	0.9

## 6.5 实验结果与分析

本节与第 6.2 节介绍的对比方法进行了不同数据集下的时间约束和 QoS 约束监测, 并对对比分析了不同影响因子下的实验结果.

### 6.5.1 服务自身时间约束监测

首先, 可对服务自身的多项时间约束进行监测, 以验证 CSTL 的适用性和有效性. 对于一个物联网服务的起始时间和结束时间, 如第 3.1.1 节所述, 可由  $CNT_{st}$  (或  $CNT_{ft}$ ) 进行约束, 例如开始时间不早于某一时间点. 在此, 以 PUC 日志数据为例, 可对该日志数据中所记录的服务执行开始时间和结束时间进行挖掘, 以此来模拟物联网服务的实时执行. 图 4(a) 即展示了实例物联网服务 (即“create purchase requisition”, 图 2 中物联网服务  $es_0$ ) 的执行开始时间和结束时间详情, 其中, 横坐标代表物联网服务标识, 纵坐标代表时间点, 橙色方块代表物联网服务从开始执行到结束执行的时长竖道图.

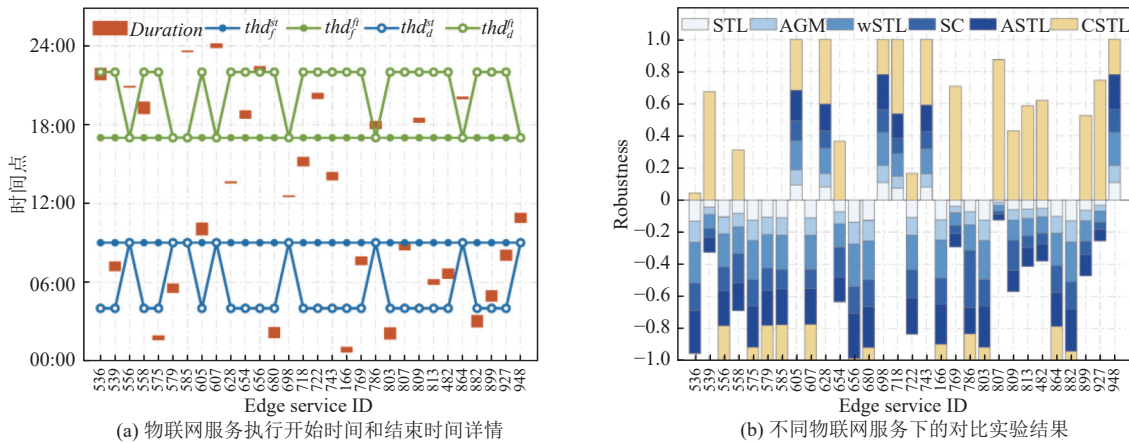


图 4 不同物联网服务执行时间详情及其对比实验结果

#### (1) 不同物联网服务下的实验结果与分析

由图 4(a) 可知, 单个物联网服务可能会执行多次, 例如, 创建多次采购申请, 并且其开始和结束时间各不相同. 本实验假设物联网服务开始时间不早于某个时间点 (如图 4(a) 中带实心点蓝色线所示, 即  $thd_f^{st} = 9:00$ ), 如表 3 中的  $\varphi_1$  所表示; 或开始时间不早于不同的时间点 (即  $thd_d^{st} = 4:00 | 9:00$ , 如图 4(a) 中带空心点蓝色线所示), 如表 3 中的  $\varphi'_1$  所表示. 对于约束  $\varphi_1$  和  $\varphi'_1$  的监测结果, 即其鲁棒性度量值, 如图 4(b) 所示 (即表 5 所对应数据). 实验结果表明某些物联网服务 (例如标识为 536、539 和 558 的物联网服务) 在对比方法下违反其约束  $\varphi_1$ , 但可在 CSTL 方法下满足  $\varphi'_1$ . 这表明 CSTL 在具备表达动态阈值能力的情况下, 可实现更加灵活的监测效果, 例如这里由于  $thd_d^{st} \leq thd_f^{st}$ , CSTL 对于约束违反情况具有更高的故障容忍度. 或者, 对于违反情况具有绝对值更小的违反度值, 例如对于标识为 575 和 656 的物联网服务.

表 5 不同物联网服务下的对比实验结果

服务标识	STL	AGM	wSTL	SC	ASTL	CSTL
536	-0.12796	-0.12785	-0.24787	-0.16605	-0.26066	<b>0.042654</b>
539	-0.05687	-0.0592	-0.11232	-0.0732	-0.11848	<b>0.867299</b>
556	-0.109	-0.10925	-0.21185	-0.15303	<b>-0.22275</b>	<b>-0.22275</b>
558	-0.05213	-0.05343	-0.10379	-0.11397	-0.109	<b>0.194313</b>
575	-0.20853	-0.21089	-0.40047	-0.27237	<b>-0.4218</b>	-0.13744
579	-0.10427	-0.10661	-0.20237	-0.13544	<b>-0.21327</b>	<b>-0.21327</b>
585	-0.18957	-0.18831	-0.36493	-0.20836	<b>-0.38389</b>	<b>-0.38389</b>
605	0.21327	0.211227	0.409479	0.278595	0.43128	<b>0.71564</b>
607	-0.19905	-0.19761	-0.38294	-0.21487	<b>-0.40284</b>	<b>-0.40284</b>
628	0.099526	0.09979	0.193365	0.129219	0.203791	<b>0.488152</b>
654	-0.04265	-0.04412	-0.08578	-0.10746	-0.09005	<b>0.21327</b>
656	-0.14692	-0.14645	-0.28389	-0.17907	<b>-0.29858</b>	-0.01422
680	-0.20379	-0.20615	-0.39147	-0.26615	<b>-0.41232</b>	-0.12796
698	0.132701	0.132295	0.256398	0.172787	<b>0.270142</b>	<b>0.270142</b>
718	0.066351	0.06728	0.130332	0.085651	0.137441	<b>0.421801</b>
722	-0.08531	-0.08599	-0.16682	-0.13675	-0.17536	<b>0.127962</b>
743	0.094787	0.095147	0.18436	0.122995	0.194313	<b>0.478673</b>
766	-0.23697	-0.23932	-0.4545	-0.30971	<b>-0.47867</b>	-0.19431
769	-0.04739	-0.04971	-0.09431	-0.06076	-0.09953	<b>0.848341</b>
786	-0.01896	-0.02082	-0.04076	<b>-0.09126</b>	-0.04265	-0.04265
803	-0.20853	-0.21089	-0.40047	-0.27237	<b>-0.4218</b>	-0.13744
807	-0.01422	-0.01646	-0.03128	-0.01778	-0.03318	<b>0.781991</b>
809	-0.03318	-0.03481	-0.06777	-0.10096	-0.07109	<b>0.232227</b>
813	-0.09005	-0.09239	-0.17536	-0.11677	-0.18483	<b>0.933649</b>
842	-0.07583	-0.07817	-0.14834	-0.0981	-0.1564	<b>0.905213</b>
864	-0.08531	-0.08599	-0.16682	-0.13675	<b>-0.17536</b>	<b>-0.17536</b>
882	-0.18009	-0.18245	-0.34645	-0.23503	<b>-0.36493</b>	-0.08057
899	-0.12322	-0.12557	-0.23839	-0.16034	-0.25118	<b>1</b>
927	-0.03791	-0.04022	-0.0763	-0.04831	-0.08057	<b>0.829384</b>
948	0.184834	0.183369	0.35545	0.241251	<b>0.374408</b>	<b>0.374408</b>

当  $thd_d^{st} = thd_f^{st}$ , 例如对于标识为 556 和 579 的物联网服务, 约束  $\varphi_1$  和  $\varphi'_1$  均代表其开始时间不早于 9:00, 此时 CSTL 对于  $\varphi'_1$  的约束满足度与 ASTL 对于  $\varphi_1$  的约束满足度相同, 且均高于其他对比方法所获鲁棒度值. 这表明 ASTL 在物联网服务监测方面具有较好的结果鲁棒性, CSTL 在其上拓展可表达边缘场景下动态阈值的变量, 具有更高的监测表达能力和场景适用性.

而对于其他的一些物联网服务, 例如标识为 605 和 628 的物联网服务, 可发现 CSTL 监测的鲁棒性度量值高于对比方法. 通常, 较高的鲁棒性分数表示监测方法对于所需监测约束的满意度较高, 这也正是目前监测方法性能评价指标之一. 综上所述, CSTL 在不同物联网服务上的约束表达能力和鲁棒性监测方面表现均优于对比方法.

(2) 不同时间约束下的实验结果与分析

此外, 还可以针对物联网服务完成时间进行时间约束及其实时监测, 如表 3 中的  $\varphi_2$  (或  $\varphi'_2$ ) 所示. 其监测的鲁棒性得分如图 5(a) 所示, 其中所示物联网服务被约束为结束时间不晚于某些时间点 (即  $thd_f^{ft} = 17:00$  和  $thd_d^{ft} = 17:00 \mid 22:00$ ), 如图 4(a) 中所示带有实心点或空心点的绿色线所示. 比较结果表明, CSTL 对于结束时间的约束监测同样可获得更加鲁棒的结果, 并可有效表达不同的阈值.

另一方面, 物联网服务在执行过程中还可对其执行时长进行约束, 如图 6(a) 中所示物联网服务的执行时长示例. 这些物联网服务可被约束为执行时长不超过特定阈值, 如表 3 中的  $\varphi_3$  (或  $\varphi'_3$ ) 所示, 以满足用户对于执行延迟的要求. 如图 6(a) 所示, 横坐标代表物联网服务标识, 纵坐标代表实际的执行时长数值. 如图 6(a) 所示固定或动态

阈值, 物联网服务执行时长阈值可设为固定值 (即  $thd_f^{dur} = 40 \text{ min}$ ), 或非统一值 (即  $thd_f^{dur} \in [30, 50]$  代表处于 30–50 min 的随机值), 如图 6(a) 所示带有实心点和空心点的绿色线. 图 6(b) 展示了不同物联网服务在执行时长约束下的鲁棒性值, 结果表明 CSTL 可有效完成对于不同时间约束 (即这里的执行时长约束) 的监测, 并获得更高的鲁棒性值或更符合非固定阈值场景下的监测需求.

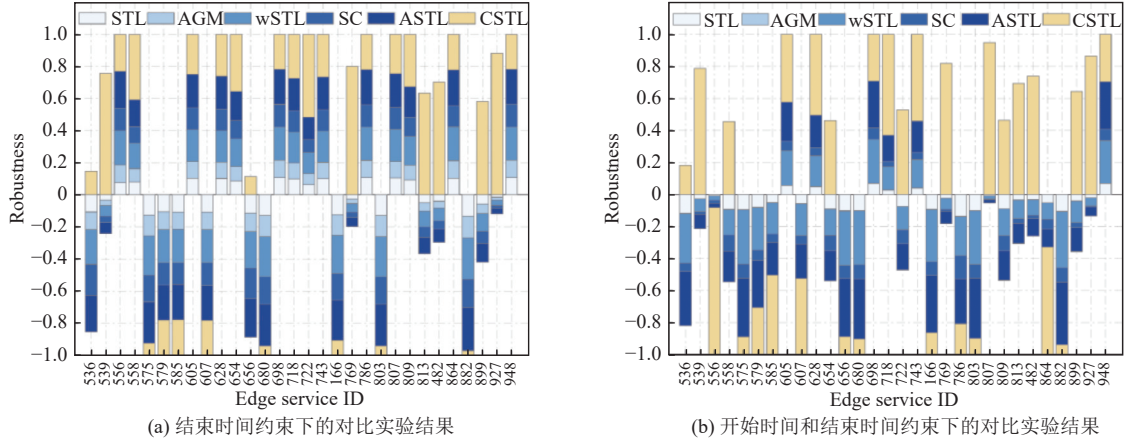


图 5 不同时间约束下的对比实验结果

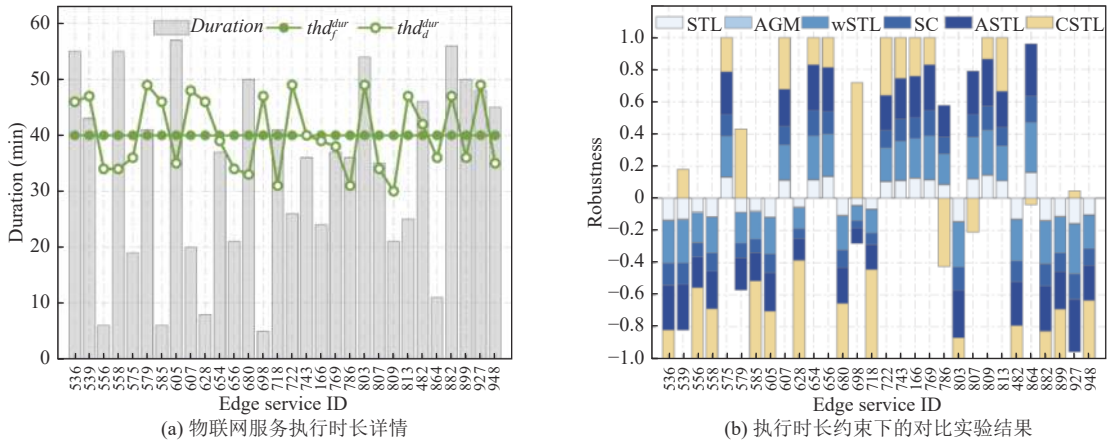


图 6 物联网服务在执行时长约束下的对比实验结果

(3) 自适应监测的实验结果与分析

由上述实验可看出, 在物联网服务的实际监测中, 不同物联网可能无法只用固定的相同阈值来表示其监测需求, 具有表达非固定阈值的监测方法具有更强的监测适用性. 本节实验以实际场景中动态监测阈值来说明自适应监测的适用性, 并对比说明仅在固定阈值下监测的不足. 例如, 不同物联网服务开始时间可被约束为不早于固定阈值 (即公式  $\varphi_1$  中所示  $thd_f^{st}$ ), 或非固定阈值 (即公式  $\varphi_1'$  中所示  $thd_f^{st}$ ). 如图 4(a) 所示, 物联网服务开始时间不早于 9:00 (或 4:00); 执行时长时间受固定阈值 (即 40 min) 或动态阈值 (30–50 min 的随机值) 的约束, 如图 6(a) 所示. 更多示例可参见表 3 所示“服务自身时间约束”中  $\varphi_i$  和  $\varphi_i'$ . 与不太适用于动态场景的静态阈值相比, 动态阈值可以更好地代表物联网服务监测的实际约束. 因此, 配备了可以在线调整的动态参数, CSTL 在具有高适应性的物联网服务监测方面具有更强的监测表达能力和场景适用性, 可有效开展物联网服务在动态场景下的自适应监测.

(4) 多个时间约束下的实验结果与分析

同时, 考虑到用户可能对物联网服务具有多个时间约束, 本实验针对物联网服务的开始时间、结束时间和执行时长进行多约束监测. 具体而言, 可对物联网服务开始时间和结束时间约束为表 3 中的  $\varphi_4$  (或  $\varphi_4'$ ), 对物联网服



务开始时间和执行时长约束为  $\varphi_5$  (或  $\varphi'_5$ ), 对物联网服务结束时间和执行时长约束为  $\varphi_6$  (或  $\varphi'_6$ ), 监测结果分别如图 5(b)、图 7(a) 和图 7(b) 所示. 在对物联网服务实现多时间约束下的监测时, 其物联网服务总的约束满足度应依赖于其各个约束的满足度. 例如, 对于标识为 605 的物联网服务, 它可分别满足用户对于开始时间和结束时间的约束, 如图 4(b) 和图 5(a) 中所示鲁棒性值, 即均为正值. 由此, 该物联网服务可同时满足该两项约束, 如图 7(a) 中所示鲁棒性值. 相反地, 标识为 556 的物联网服务由于违反其开始时间约束, 如图 4(b) 所示, 以至于该物联网服务的监测结果显示其无法同时满足该两项约束 (如图 7(b) 所示). 同理, 对于物联网服务执行时长与开始/结束时间的约束, 其结果依赖于图 6(b) 与图 4(b)(或图 6(b) 与图 5(a)) 所示监测结果, 最终结果如图 7 所示.

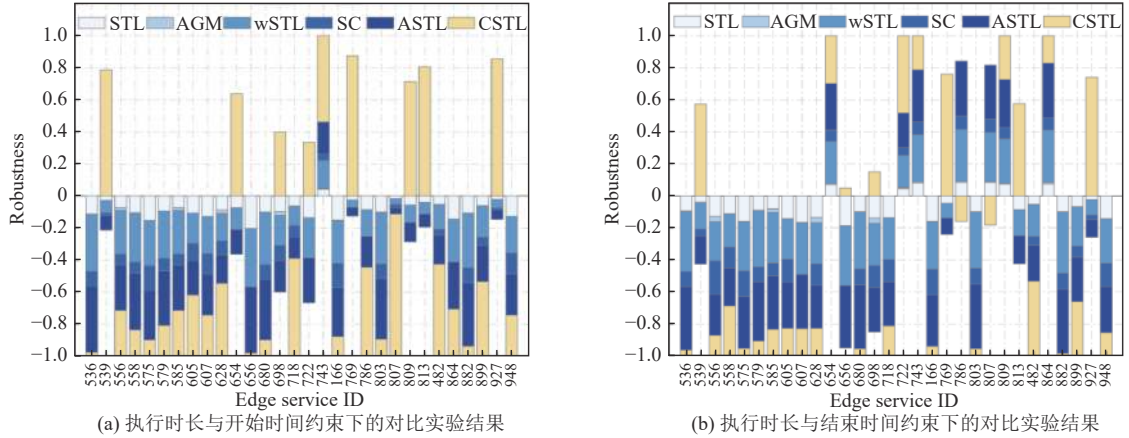


图 7 执行时长与开始/结束时间约束下的对比实验结果

### 6.5.2 服务自身 QoS 约束监测

此外, 可对物联网服务自身 QoS 约束进行模拟监测实验, 即基于 WS-DREAM 数据集中服务执行的真实 QoS 数值进行不同 QoS 约束监测. 具体而言, WS-DREAM 数据集记录了 4532 个物联网服务在 64 个时间点下的 QoS 值 (即响应时间和吞吐量). 图 8 展示了某一物联网服务 (记作  $es_0$ ) 在不同时间点下的响应时间和吞吐量, 图 9 则针对不同物联网服务, 展示了其在某一时间点 (即  $t_0$ ) 下的响应时间和吞吐量. 其中, 带有实心点的蓝色线和绿色线分别代表了 QoS 约束中的动态阈值 (即  $thd_d^r$  和  $thd_d^p$ ) 和固定阈值 (即  $thd_f^r$  和  $thd_f^p$ ). 由图可发现单一物联网服务在不同时间点下的 QoS 数值会随时间变化, 并且不同物联网服务在同一时间点下的 QoS 数值均不同. 下面将针对该 QoS 数值情况进行实时监测, 和不同影响因子下的对比实验.

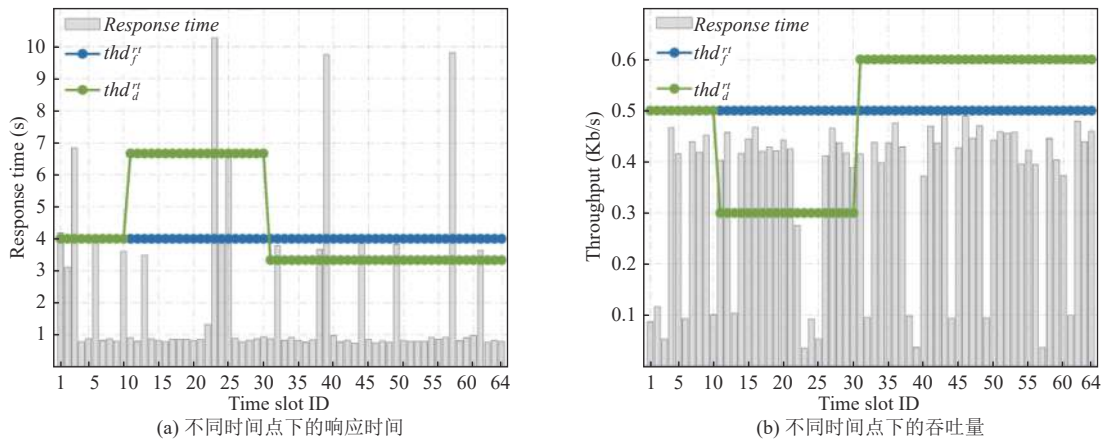


图 8 物联网服务  $es_0$  在不同时间点下的响应时间与吞吐量

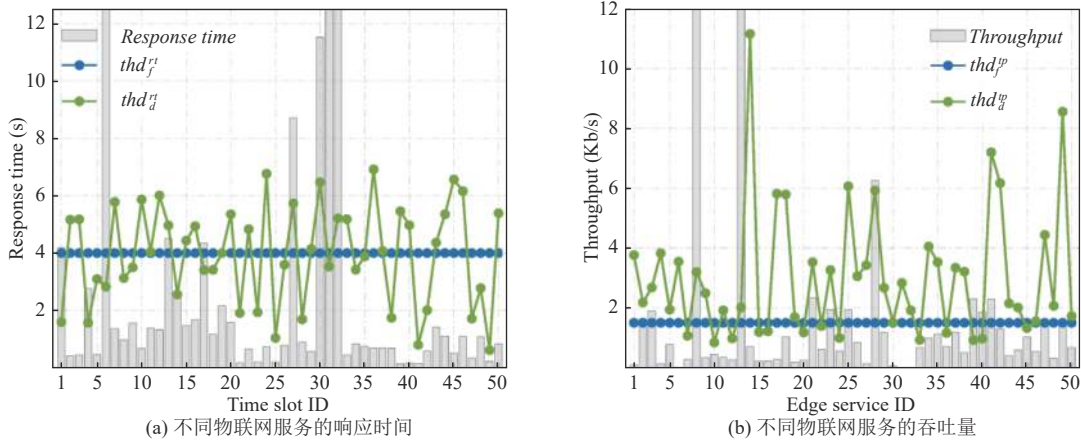


图9 不同物联网服务在时间点  $t_0$  下的响应时间与吞吐量

(1) 实时监测下的实验结果与分析

图 10 显示了物联网服务  $e_{s_0}$  在一段时间段内 (即 64 个时间点) 对于响应时间和吞吐量两个 QoS 约束上的鲁棒性值, 即由  $\varphi_7$  (或  $\varphi_7'$ ) 和  $\varphi_8$  (或  $\varphi_8'$ ) 所指定的全局约束的监测结果. 在本实验场景中, 物联网服务 QoS 值随时间变化, 如图 8 所示不同时间点下的响应时间和吞吐量. 这说明一些全局约束, 即物联网服务可能无法满足某些 QoS 值总是满足特定约束, 例如图 10 中所示负数鲁棒性部分. 并且, 图示结果表明 CSTL 在约束可满足时可获得更高的满意度值, 例如图 10(a) 中  $t_4$  至  $t_{22}$  期间. 由于 CSTL 可表达在运行时自适应调整阈值为合适值的能力, 它可更鲁棒地识别 QoS 违规, 例如, 在  $t_{23}$  和  $t_{32}$ . 具体而言, 对比方法在  $t_{32}$  处产生正鲁棒性值, 即误判 QoS 违规现象为正常执行状态, 而 CSTL 可成功识别出具有负鲁棒性值的违规行为. 由此, 可说明 CSTL 方法具有更强的监测表达能力和鲁棒性.

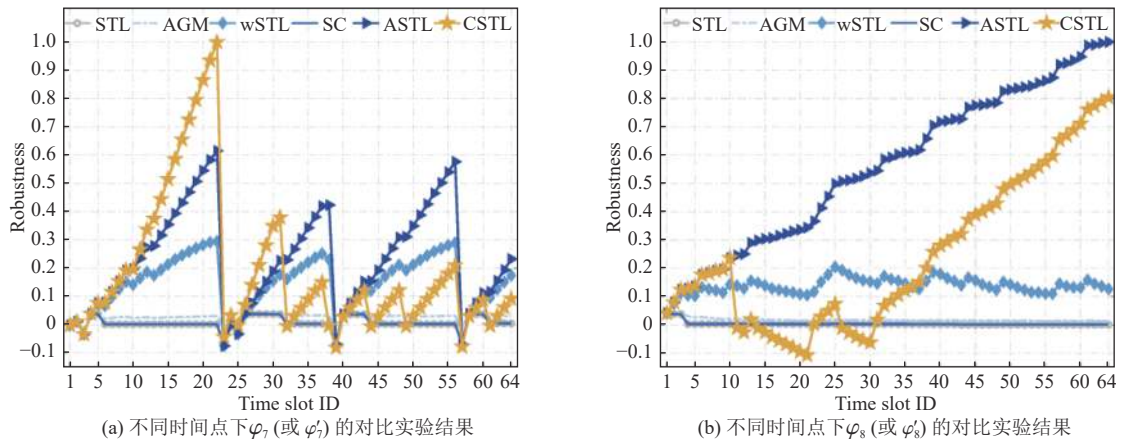


图 10 物联网服务  $e_{s_0}$  在不同时间点下的对比实验结果

以类似的方式, 图 10(b) 显示了针对物联网服务吞吐量所指定的 QoS 约束的鲁棒性值, 即  $\varphi_8$  (或  $\varphi_8'$ ) 的监测结果. 由图 10 可看出, CSTL 所获得的鲁棒性值并非总是最高, 这是由于它可以根据实际动态阈值进行实时监测, 并捕获所有的 QoS 违规情况, 所以也将更符合实际的监测结果. 因此, 配备了表达变化参数的语法和语义, CSTL 在表达能力、适用性和鲁棒性方面对比方法表现更好.

(2) 不同物联网服务下的实验结果与分析

此外, 可对不同物联网服务进行 QoS 约束监测, 即图 9 所示总共 50 个服务. 这 50 个服务在某一时间点  $t_0$  下

响应时间和吞吐量的 QoS 值分别如图 8(a) 和图 8(b) 所示. 可监测这些不同物联网服务是否均可满足响应时间或吞吐量上特定约束, 即表 3 中  $\varphi_9$  (或  $\varphi'_9$ ) 和  $\varphi_{10}$  (或  $\varphi'_{10}$ ) 所示约束. 监测结果分别如图 10(a) 和图 10(b) 所示. 结果表明, CSTL 方法可有效监测不同物联网服务上的 QoS 约束, 并获得更加鲁棒的结果.

(3) 不同 QoS 约束下的实验结果与分析

此外, 可采用 WS-DREAM 数据集中的 QoS 值来评估对不同 QoS 约束的监测. 如图 10 和图 11 所示比较结果, 物联网服务运行态监测可针对不同的 QoS 属性 (即响应时间和吞吐量) 实现不同 QoS 约束的监测. 比较结果表明, CSTL 方法在对物联网服务指定动态 QoS 约束时表达能力更强, 并且在各种服务自身 QoS 约束 (如表 3 所示) 下实现更鲁棒的满意度值.

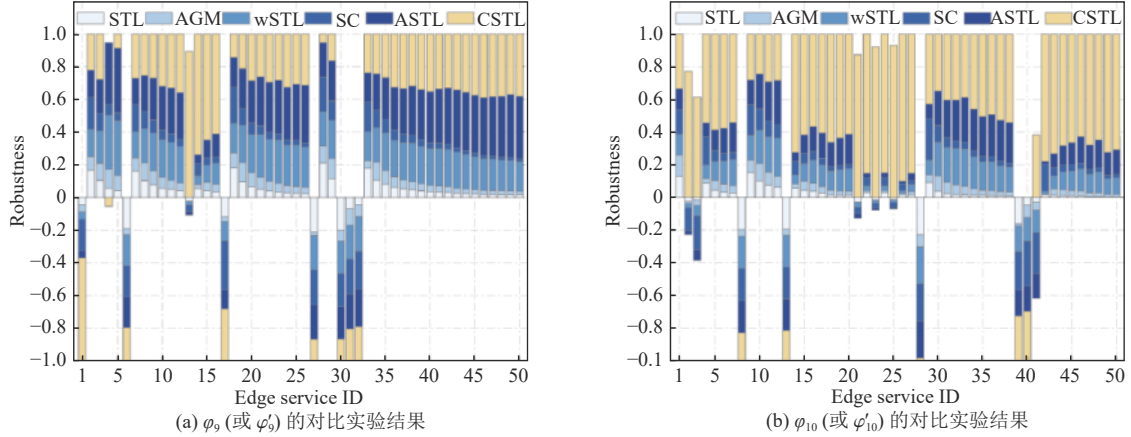


图 11 不同物联网服务在时间点  $t_0$  下的对比实验结果

(4) 动态与固定阈值下的实验结果与分析

如图 8 和图 9 所示, “服务自身 QoS 约束”的阈值可为固定不变 (即  $thd_q^r$  和  $thd_q^p$ ), 或动态变化 (即  $thd_f^r$  和  $thd_f^p$ ), 这与服务自身时间约束监测类似 (如图 4(a) 和图 6(a) 所示). 考虑到实际场景中的动态特性, CSTL 方法较对比方法更适应实际监测场景, QoS 约束中的某些参数可进行在线自适应调整. 例如, QoS 约束可能涉及根据执行时实时网络进行调整的不同阈值, 如根据实时网络中的动态资源或容量以确定物联网服务吞吐量极值. 具体运行实例可参见表 3 所示  $\varphi_7$  (或  $\varphi'_7$ ) 至  $\varphi_{10}$  (或  $\varphi'_{10}$ ). 图 10 和图 11 所示监测结果表明 CSTL 较对比方法具有更高的鲁棒性值, 并由此说明 CSTL 更适用于动态边缘场景下的自适应性监测场景.

6.5.3 服务之间时间依赖监测

(1) 不同日志下的实验结果与分析

在本节实验中, 将根据真实服务执行日志分析各种服务间时间关系和依赖性. 表 2 所示日志的每一日志中服务之间的 7 种时间关系如图 12(a) 所示, 结果表明不同的服务执行日志具有不同的服务间时间关系. 其中, “before”占比最多, 为 74%–97.5% 不等, 这表明一个物联网服务通常在另一物联网服务之前 (或之后) 执行. 其中, 服务执行日志 PRD 具有最多多样化的时间关系, 其中“overlap”“during”“finish”和“equal”的百分比均大于其他日志中的相应百分比.

(2) 不同时间依赖下的实验结果与分析

以服务执行日志 PRD 为例, 可分析其时间依赖 SF 和 FS 值, 如图 13(a) 所示. 当一个物联网服务在另一个物联网服务之前执行时, 用户可能会关心 SF 和 FS 这样两个时间依赖, 以研究: ① 这两个顺序执行服务的总执行时间, 以及 ② 它们之间的执行时间间隔. 在 PRD 日志中, “Final Inspection”服务在“Packing”之前执行, 记作  $es_3$  和  $es_4$ . 其 SF 值的直方图 (范围从 8–874 h) 和 FS 值 (范围从 6–872 h) 如图 12(b) 所示. 该直方统计图表明 SF 值为 8 h 的情况共在 PRD 日志中记录过 2 次, FS 值为 6 h 的服务之间时间间隔出现过 7 次. 并且, 可发现 SF 值大多在 9–19 h 之间, FS 值多数处于 6–17 h 之间, 即出现次数超过 5 次的情况. 两个服务自身时长 (记作  $Dur_1$  和  $Dur_2$ ), 及其 SF 与 FS 时间依赖时长, 如图 13(a) 展示, 可有如下关系:  $SF = Dur_1 + Dur_2 + FS$ .

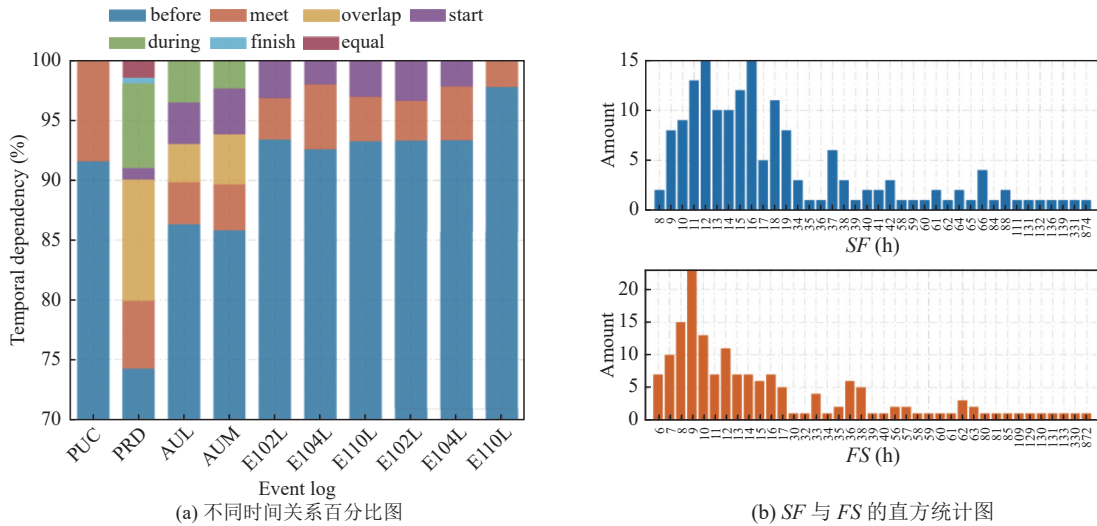


图 12 不同日志下的时间关系和时间依赖统计图

在这个实验中, 服务之间时间依赖  $FS$  值可模拟设置约束为不超过 12 h (即表 3 中所示  $\varphi_{11}$ ), 且  $SF$  值不超过 24 h (即  $\varphi_{14}$  所示约束). 考虑到用户可能允许服务之间的时间间隔超出一定阈值, 只需两个服务的总执行时长满足其要求即可, 因此该服务之间时间依赖可表示为  $FS$  数值不超过 12 h, 或不超过 24 h (即  $\varphi_{12}$  所示约束) 即可, 即由  $\varphi_{13}$  进行规约;  $SF$  数值将约束为总是不超过 24 h. 对于该约束, 即由  $\varphi_{15}$  (或  $\varphi'_{15}$ ) 所表示, 其监测结果如图 13(b) 所示. 结果表明 CSTL 方法可针对不同时间依赖进行有效约束与监测, 并由于其可表达动态可变参数的语法使其具有更高的鲁棒性和动态场景监测适用性.

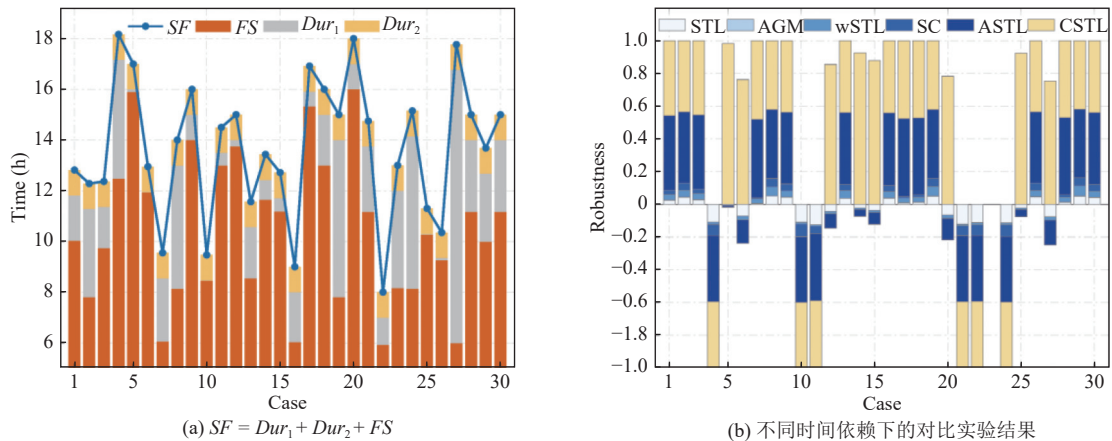


图 13 不同时间依赖的数值统计详情

(3) 不同时间关系下的实验结果与分析

除了考虑服务之间时间关系“before”, 本实验针对服务之间另一常见时间关系“overlap”进行实验分析. 具体而言, 顺序执行的物联网服务之间时间关系通常表现为 before, 当两个物联网服务呈时间重叠的执行, 该两个物联网服务并行执行. 在此同样考虑时间依赖  $SF$  与  $FS$  数值约束, 以物联网服务“Flat Grinding”和“Lapping” (记作  $es_6$  和  $es_7$ ) 为例, 该两个物联网服务自身执行时长 (即  $Dur_1$  和  $Dur_2$ ) 及其  $SF$  与  $FS$  数值关系如图 14(a) 所示. 结果表明其具有如下关系:  $SF = Dur_1 + Dur_2 - |FS|$ .

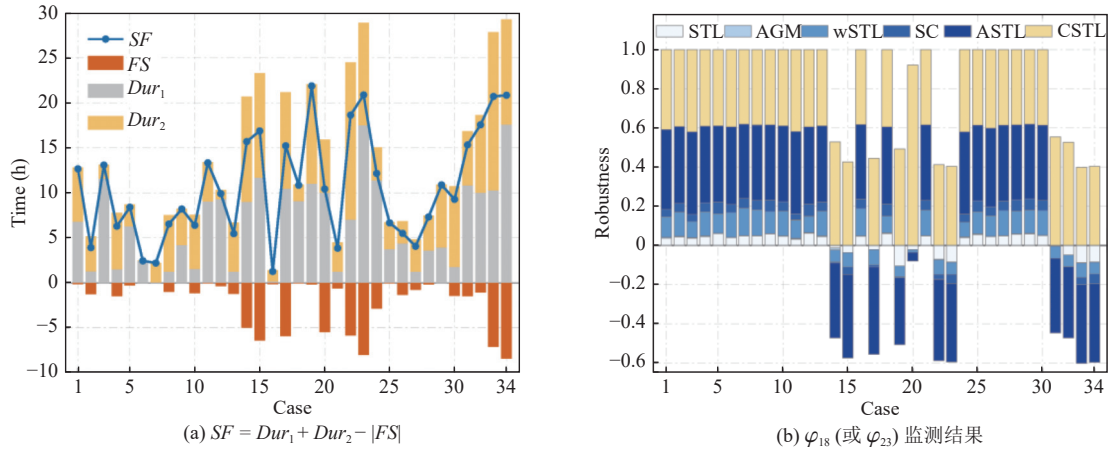


图 14 不同时间关系下的对比实验结果

在此实验中, 可假设用户需求为该两个物联网服务之间  $FS$  值不超过 5 h (即表 3 中所示  $\varphi_{16}$ ), 并且  $SF$  值不超过 15 h (即  $\varphi_{17}$  所示约束), 该时间依赖约束最终由  $\varphi_{18}$  指定. 当考虑用户对于  $FS$  和  $SF$  值具有一定偏差容忍度时, 可实现如下监测, 即  $FS$  值不超过 10 h (即  $\varphi_{19}$  所示约束), 且  $SF$  值不超过 25 h (即  $\varphi_{21}$  所示约束), 也可满足用户需求, 即使  $FS$  超过 5 h 或  $SF$  超过 15 h. 该时间依赖约束最终由表 3 中  $\varphi_{18}$  指定.  $\varphi_{18}$  (或  $\varphi_{23}$ ) 监测结果如图 14(b) 所示, 比较结果表明 CSTL 方法性能优于对比方法, 其可通过在线更新约束来实现自适应约束监测, 更适用于实际动态场景监测.

#### 6.5.4 服务之间 QoS 依赖监测

图 15 显示了根据响应时间和吞吐量指定的服务之间 QoS 依赖的监测结果. 如图 8(a) 和图 8(b) 所示, 物联网服务 (即  $es_0$ ) 的响应时间和吞吐量各不相同, 并随时间推移而发生变化. 由此, 表 3 中所示“服务自身 QoS 约束”可能被满足或违反, 如图 10 和图 11 所示. 考虑到 QoS 约束之间的时间间隔, 我们可以指定服务之间 QoS 依赖性, 如表 3 中 CSTL 公式  $\varphi_{26}$  (或  $\varphi'_{26}$ ) 和  $\varphi_{29}$  (或  $\varphi'_{29}$ ). 其中公式中显示的参数  $\Delta t_{max}$  代表时间间隔的最大值, 并且  $10 \leq \Delta t_{max} \leq 30$ . 该参数可影响实际监测结果, 如图 15 所示, 这是由于后一个 QoS 约束受前一 QoS 约束结束后时间间隔的影响. 例如, 图 15(a) 所示结果表明该 QoS 依赖可在  $10 \leq \Delta t_{max} \leq 13$  时满足, 但在  $14 \leq \Delta t_{max} \leq 20$  时由于后一 QoS 约束无法在这一时间间隔后总是满足其约束, 该 QoS 依赖呈负数鲁棒性值. 总的来说, 该结果表明 CSTL 方法可有效监测服务之间 QoS 依赖, 并在可满足用户需求时鲁棒性值更高, 在违背用户需求时由于其动态阈值使得其对于违约度具有更高的容忍度.

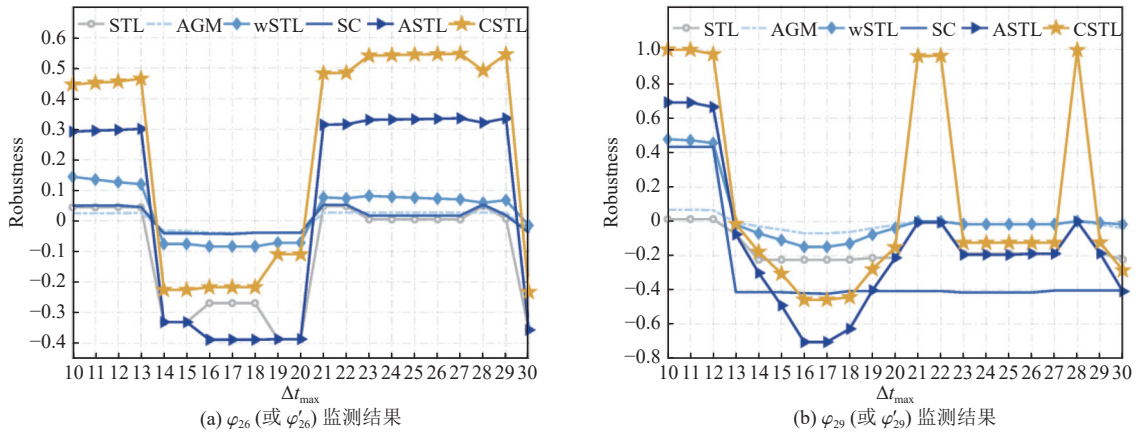


图 15 服务之间 QoS 依赖的对比实验结果

### 6.5.5 复合服务约束监测

此外,可对物联网服务组成的复合服务进行监测.在此,可考虑不同组合结构下的复合服务,以及考虑复合服务中前后服务之间的关联影响,实现更加鲁棒的复合服务监测.

#### (1) 复合服务与独立服务下的实验结果与分析

如第3节所述,可针对复合服务指定其时间约束和QoS约束,以保证多个物联网服务组成的复合服务在其执行过程中满足用户请求.与针对独立的物联网服务监测对比,复合服务在考虑多个服务的整体性下将更具有应用适用性.以WS-DREAM数据集中前两个物联网服务为例,记作 $es_0$ 和 $es_1$ ,该两个服务的响应时间和吞吐量监测可分别从复合服务及其独立服务两个角度进行对比分析.并且,在该实验中,考虑 $es_0$ 和 $es_1$ 不同的组合结构,即 $es_0 \rightarrow es_1$ 和 $es_0 \oplus es_1$ ,实现其复合服务监测结果的对比分析.

在该模拟实验中,可假设该两个服务分别响应时间小于3s,如表3中CSTL公式 $\varphi_{30}$ 和 $\varphi_{31}$ 所示.或者,该两个服务整体响应时间小于6s,即对于顺序执行时 $es_0 \rightarrow es_1$ ,其约束为 $\varphi_{32}$ ,对于并行执行时 $es_0 \oplus es_1$ ,其约束表示为 $\varphi_{33}$ .针对该 $\varphi_{30}$ 、 $\varphi_{31}$ 、 $\varphi_{32}$ 和 $\varphi_{33}$ 的实时监测结果如图16(a)所示.结果表明,①并行结构下的复合服务的鲁棒性值(图16(a)所示 $comp_{pal}$ ,即 $\varphi_{33}$ 的监测结果)高于顺序结构下复合服务的鲁棒性值(图16(a)所示 $comp_{seq}$ ,即 $\varphi_{32}$ 的监测结果),即 $comp_{pal} > comp_{seq}$ ;并且②复合服务的鲁棒性值高于独立服务的监测结果(图16(a)所示 $ncomp_{es_0}$ 和 $ncomp_{es_1}$ ),即 $comp_{pal} > ncomp_{es_0}$ ,  $comp_{pal} > ncomp_{es_1}$ ,  $comp_{seq} > ncomp_{es_0}$ ,  $comp_{seq} > ncomp_{es_1}$ .由此,可说明CSTL可有效监测复合服务上约束,并较独立的物联网服务可实现更优监测结果.

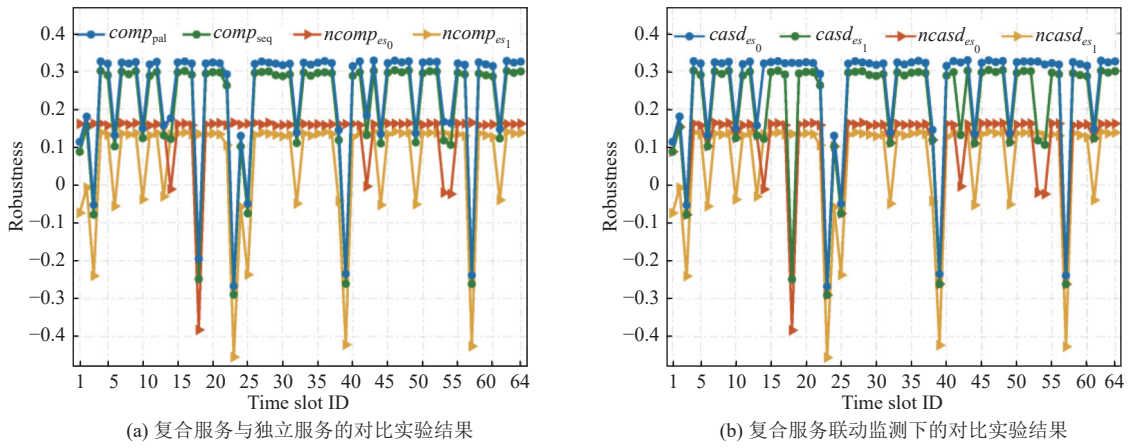


图16 复合服务约束监测的对比实验结果

#### (2) 不同复合服务联动监测下的实验结果与分析

复合服务的监测可能涉及其中前后服务之间的级联影响,即某一物联网服务的性能可能会影响之后物联网服务的约束满足结果.例如,不同于上一实验中考虑单个独立服务上的响应时间约束(如公式 $\varphi_{30}$ 和 $\varphi_{31}$ 所示),本实验可进行如下约束监测:物联网服务 $es_0$ 响应时间不超过3s,且物联网服务 $es_1$ 响应时间不超过3s或 $6 - es_0.rt$ (如公式 $\varphi_{34}$ 所示),即该约束可由表3中CSTL公式 $\varphi_{35}$ 表示;此外,考虑到复合服务监测的整体性,可设置约束如下:物联网服务 $es_0$ 响应时间不超过6s,且物联网服务 $es_1$ 响应时间不超过 $6 - es_0.rt$ ,即该约束可由表3中CSTL公式 $\varphi_{37}$ 表示.

针对上述 $\varphi_{35}$ 和 $\varphi_{37}$ 的实时监测结果见图16(b).结果表明,考虑复合服务之间的联动影响可实现更好的监测效果,即考虑联动影响下的物联网服务鲁棒性值(记作 $casd_{es_0}$ 和 $casd_{es_1}$ )高于未考虑的情况(记作 $ncasd_{es_0}$ 和 $ncasd_{es_1}$ ), $casd_{es_0} > ncasd_{es_0}$ ,  $casd_{es_1} > ncasd_{es_1}$ .并且,可发现物联网服务 $es_0$ 的鲁棒性值高于物联网服务 $es_1$ ,即 $casd_{es_0} > casd_{es_1}$ ,  $ncasd_{es_0} > ncasd_{es_1}$ .这是由于物联网服务 $es_1$ 的动态监测阈值(即 $6 - es_0.rt$ )较物联网服务 $es_0$ 的阈值(即6)更加严苛,当 $es_0.rt > 0$ .总的来说,CSTL方法可考虑复合服务之间的联动影响,实现复合服务的联动监测,其中监测约束中的参数可随实际监测结果进行动态调整.

综上所述,本节对物联网服务自身、服务之间以及复合服务的时间约束和QoS约束进行了多维多场景的监

测对比, 本文所提 CSTL 方法在上述场景和实验参数均取得比现有方法更好的性能, 表现出更强的服务监测适用性, 即更高的物联网服务约束鲁棒度值. 具体而言, 传统信号时序逻辑 STL 由于仅依赖于极值点, 存在信号不可辨问题; 其拓展工作 AGM、SC、wSTL 由于仅考虑固定采样率信号值, 未能适用于动态场景监测; 拓展了累加型鲁棒度量方式的 ASTL 方法可有效解决上述方法存在的缺陷, 而再拓展聚合算子和动态变量的 CSTL 方法则可面向复合服务监测场景, 实现复合服务的整体联动定量监测. 实验结果从不同维度论证了 CSTL 方法较现有方法具有更好的监测表达能力、场景适用性和结果鲁棒性.

## 7 总 结

本文针对现有物联网服务在实时定量监测方面存在的缺陷, 基于传统信号时序逻辑扩展了一种累加型鲁棒性度量方式, 通过强调所有子公式在其整个时域内的鲁棒性值, 实现了一种时序鲁棒、信号可辨、动态适用的物联网服务运行态实时定量监测机制; 并提出了一种具有扩展聚合算子和在线参数设置的新型聚合型信号时序逻辑, 即 CSTL, 实现了一种自适应监测机制. 具体而言, 物联网服务自身、服务之间以及复合服务的时间约束和 QoS 约束可以通过 CSTL 公式进行规约表示, 并且可在物联网服务运行时定性和定量解释约束满意度, 有效地监测物联网服务 QoS 变化和时间违规情况. 基于公开数据集的评估结果表明, 本文所提出的 CSTL 方法较现有方法具有更好的监测表达能力、场景适用性和结果鲁棒性.

## References:

- [1] Rafique W, Qi LY, Yaqoob I, Imran M, Rasool RU, Dou WC. Complementing IoT services through software defined networking and edge computing: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 2020, 22(3): 1761–1804. [doi: [10.1109/comst.2020.2997475](https://doi.org/10.1109/comst.2020.2997475)]
- [2] Neiat AG, Bouguettaya A, Bahutair M. A deep reinforcement learning approach for composing moving IoT services. *IEEE Trans. on Services Computing*, 2022, 15(5): 2538–2550. [doi: [10.1109/TSC.2021.3064329](https://doi.org/10.1109/TSC.2021.3064329)]
- [3] Wang SG, Zhou A, Bao R, Chou W, Yau SS. Towards green service composition approach in the cloud. *IEEE Trans. on Services Computing*, 2021, 14(4): 1238–1250. [doi: [10.1109/TSC.2018.2868356](https://doi.org/10.1109/TSC.2018.2868356)]
- [4] Hayyolalam V, Pourhaji Kazem AA. A systematic literature review on QoS-aware service composition and selection in cloud environment. *Journal of Network and Computer Applications*, 2018, 110: 52–74. [doi: [10.1016/j.jnca.2018.03.003](https://doi.org/10.1016/j.jnca.2018.03.003)]
- [5] Yang WQ, Liu GZ. Research on QOE aware service composition strategy in smart city. *Computer Technology and Development*, 2021, 31(4): 176–181 (in Chinese with English abstract). [doi: [10.3969/j.issn.1673-629X.2021.04.030](https://doi.org/10.3969/j.issn.1673-629X.2021.04.030)]
- [6] Zhao D, Zhou ZB, Xue X, Zhao ZF, Gaaloul W, Zhang WB. CTL-based dynamic IoT service composition. In: *Proc. of the 2021 IEEE Int'l Conf. on Web Services*. Chicago: IEEE, 2021. 381–388. [doi: [10.1109/ICWS53863.2021.00057](https://doi.org/10.1109/ICWS53863.2021.00057)]
- [7] Rama-Maneiro E, Vidal JC, Lama M. Deep learning for predictive business process monitoring: Review and benchmark. *IEEE Trans. on Services Computing*, 2023, 16(1): 739–756. [doi: [10.1109/TSC.2021.3139807](https://doi.org/10.1109/TSC.2021.3139807)]
- [8] Tundo A, Mobilio M, Riganelli O, Mariani L. Automated probe life-cycle management for monitoring-as-a-service. *IEEE Trans. on Services Computing*, 2023, 16(2): 969–982. [doi: [10.1109/TSC.2022.3180500](https://doi.org/10.1109/TSC.2022.3180500)]
- [9] Costa B, Bachiega J, Carvalho LR, Rosa M, Araujo A. Monitoring fog computing: A review, taxonomy and open challenges. *Computer Networks*, 2022, 215: 109189. [doi: [10.1016/j.comnet.2022.109189](https://doi.org/10.1016/j.comnet.2022.109189)]
- [10] Trihinas D, Pallis G, Dikaiakos MD. Low-cost adaptive monitoring techniques for the Internet of Things. *IEEE Trans. on Services Computing*, 2021, 14(2): 487–501. [doi: [10.1109/TSC.2018.2808956](https://doi.org/10.1109/TSC.2018.2808956)]
- [11] Zhang PC, Zhang YL, Dong H, Jin HY. Mobility and dependence-aware QoS monitoring in mobile edge computing. *IEEE Trans. on Cloud Computing*, 2021, 9(3): 1143–1157. [doi: [10.1109/TCC.2021.3063050](https://doi.org/10.1109/TCC.2021.3063050)]
- [12] Dev Gupta S, Genc B, O'Sullivan B. Explanation in constraint satisfaction: A survey. In: *Proc. of the 30th Int'l Joint Conf. on Artificial Intelligence*. Montreal, 2021. 4400–4407. [doi: [10.24963/ijcai.2021/601](https://doi.org/10.24963/ijcai.2021/601)]
- [13] Tian Y, Zhang YJ, Su YS, Zhang XY, Tan KC, Jin YC. Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization. *IEEE Trans. on Cybernetics*, 2022, 52(9): 9559–9572. [doi: [10.1109/TCYB.2020.3021138](https://doi.org/10.1109/TCYB.2020.3021138)]
- [14] Zhou XG, Ahmed B, Aylor JH, Asare P, Alemzadeh H. Hybrid knowledge and data driven synthesis of runtime monitors for cyber-physical systems. *IEEE Trans. on Dependable and Secure Computing*, 2023. [doi: [10.1109/TDSC.2023.3242653](https://doi.org/10.1109/TDSC.2023.3242653)]

- [15] Ma MY, Bartocci E, Lifland E, Stankovic JA, Feng L. A novel spatial-temporal specification-based monitoring system for smart cities. *IEEE Internet of Things Journal*, 2021, 8(15): 11793–11806. [doi: [10.1109/JIOT.2021.3069943](https://doi.org/10.1109/JIOT.2021.3069943)]
- [16] Haris I, Ferreira LL, Okic I, Dukkon A, Tucakovic Z, Grosu R. QoS for dynamic deployment of IoT services. In: *Proc. of the 22nd IEEE Int'l Conf. on Industrial Technology*. Valencia: IEEE, 2021. 1144–1151. [doi: [10.1109/ICIT46573.2021.9453670](https://doi.org/10.1109/ICIT46573.2021.9453670)]
- [17] Antonyshyn L, Silveira J, Givigi S, Marshall J. Multiple mobile robot task and motion planning: A survey. *ACM Computing Surveys*, 2023, 55(10): 213. [doi: [10.1145/3564696](https://doi.org/10.1145/3564696)]
- [18] Mehdipour N, Vasile CI, Belta C. Average-based robustness for continuous-time signal temporal logic. In: *Proc. of the 58th IEEE Conf. on Decision and Control*. Nice: IEEE, 2019. 5312–5317. [doi: [10.1109/CDC40024.2019.9029989](https://doi.org/10.1109/CDC40024.2019.9029989)]
- [19] Pant YV, Abbas H, Mangharam R. Smooth operator: Control using the smooth robustness of temporal logic. In: *Proc. of the 2017 IEEE Conf. on Control Technology and Applications (CCTA)*. Maui: IEEE, 2017. 1235–1240. [doi: [10.1109/CCTA.2017.8062628](https://doi.org/10.1109/CCTA.2017.8062628)]
- [20] Mehdipour N, Vasile CI, Belta C. Arithmetic-geometric mean robustness for control from signal temporal logic specifications. In: *Proc. of the 2019 American Control Conf. Philadelphia: IEEE*, 2019. 1690–1695. [doi: [10.23919/ACC.2019.8814487](https://doi.org/10.23919/ACC.2019.8814487)]
- [21] Haghighi I, Mehdipour N, Bartocci E, Belta C. Control from signal temporal logic specifications with smooth cumulative quantitative semantics. In: *Proc. of the 2019 IEEE Conf. on Decision and Control (CDC)*. Nice: IEEE, 2019. 4361–4366. [doi: [10.1109/CDC40024.2019.9029429](https://doi.org/10.1109/CDC40024.2019.9029429)]
- [22] Su GX, Liu L, Zhang MJ, Rosenblum DS. Quantitative verification for monitoring event-streaming systems. *IEEE Trans. on Software Engineering*, 2022, 48(2): 538–550. [doi: [10.1109/TSE.2020.2996033](https://doi.org/10.1109/TSE.2020.2996033)]
- [23] Zhang YW, Cui GM, Deng SG, Chen FF, Wang Y, He Q. Efficient query of quality correlation for service composition. *IEEE Trans. on Services Computing*, 2021, 14(3): 695–709. [doi: [10.1109/TSC.2018.2830773](https://doi.org/10.1109/TSC.2018.2830773)]
- [24] Guidara I, Guermouche N, Chaari T, Jmaiel M. Time-aware selection approach for service composition based on pruning and improvement techniques. *Software Quality Journal*, 2020, 28(3): 1245–1277. [doi: [10.1007/s11219-019-09473-2](https://doi.org/10.1007/s11219-019-09473-2)]
- [25] Luo HY, Liu X, Liu J, Yang Y, Grundy J. Runtime verification of business cloud workflow temporal conformance. *IEEE Trans. on Services Computing*, 2022, 15(2): 833–846. [doi: [10.1109/TSC.2019.2962666](https://doi.org/10.1109/TSC.2019.2962666)]
- [26] Li M, Dai HP, Wang XY, Xia R, Liu AX, Chen GH. Thresholded monitoring in distributed data streams. *IEEE/ACM Trans. on Networking*, 2020, 28(3): 1033–1046. [doi: [10.1109/TNET.2020.2979654](https://doi.org/10.1109/TNET.2020.2979654)]
- [27] Barkowsky M, Brand T, Giese H. Improving adaptive monitoring with incremental runtime model queries. In: *Proc. of the 2021 Int'l Symp. on Software Engineering for Adaptive and Self-managing Systems (SEAMS)*. Madrid: IEEE, 2021. 71–77. [doi: [10.1109/SEAMS51251.2021.00019](https://doi.org/10.1109/SEAMS51251.2021.00019)]
- [28] Khurana S, Deb N, Mistry S, Ghose A, Krishna A, Dam HK. Egalitarian transient service composition in crowdsourced IoT environment. *IEEE Trans. on Services Computing*, 2023, 16(5): 3305–3317. [doi: [10.1109/TSC.2023.3264581](https://doi.org/10.1109/TSC.2023.3264581)]
- [29] Sefati S, Navimipour NJ. A QoS-aware service composition mechanism in the Internet of Things using a hidden-Markov-model-based optimization algorithm. *IEEE Internet of Things Journal*, 2021, 8(20): 15620–15627. [doi: [10.1109/JIOT.2021.3074499](https://doi.org/10.1109/JIOT.2021.3074499)]
- [30] Chattopadhyay S, Banerjee A. QoS constrained large scale Web service composition using abstraction refinement. *IEEE Trans. on Services Computing*, 2020, 13(3): 529–544. [doi: [10.1109/TSC.2017.2707548](https://doi.org/10.1109/TSC.2017.2707548)]
- [31] Palade A, Clarke S. Collaborative agent communities for resilient service composition in mobile environments. *IEEE Trans. on Services Computing*, 2022, 15(2): 876–890. [doi: [10.1109/TSC.2020.2964753](https://doi.org/10.1109/TSC.2020.2964753)]
- [32] Mostafa B, Benslimane A, Saleh M, Kassem S, Molnar M. An energy-efficient multiobjective scheduling model for monitoring in Internet of Things. *IEEE Internet of Things Journal*, 2018, 5(3): 1727–1738. [doi: [10.1109/JIOT.2018.2792326](https://doi.org/10.1109/JIOT.2018.2792326)]
- [33] Cociglio M, Fioccola G, Marchetto G, Sapio A, Sisto R. Multipoint passive monitoring in packet networks. *IEEE/ACM Trans. on Networking*, 2019, 27(6): 2377–2390. [doi: [10.1109/TNET.2019.2950157](https://doi.org/10.1109/TNET.2019.2950157)]
- [34] Xu Z, Chen YX, Topcu U. Adaptive teaching of temporal logic formulas to preference-based learners. In: *Proc. of the 35th AAAI Conf. on Artificial Intelligence*. Palo Alto: AAAI Press, 2021. 5061–5068. [doi: [10.1609/aaai.v35i6.16640](https://doi.org/10.1609/aaai.v35i6.16640)]
- [35] Xu Z, Topcu U. Transfer of temporal logic formulas in reinforcement learning. In: *Proc. of the 28th Int'l Joint Conf. on Artificial Intelligence*. Macao: AAAI Press, 2019. 4010–4018.
- [36] Maler O, Nickovic D. Monitoring temporal properties of continuous signals. In: *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*. Grenoble: Springer, 2004. 152–166. [doi: [10.1007/978-3-540-30206-3\\_12](https://doi.org/10.1007/978-3-540-30206-3_12)]
- [37] Chen G, Liu M, Kong ZD. Temporal-logic-based semantic fault diagnosis with time-series data from industrial Internet of Things. *IEEE Trans. on Industrial Electronics*, 2021, 68(5): 4393–4403. [doi: [10.1109/TIE.2020.2984976](https://doi.org/10.1109/TIE.2020.2984976)]
- [38] Huang YC, Li J, Wang N, Sun H. A multi-robot cooperative planning method based on local LTL task specifications. In: *Proc. of the 40th Chinese Control Conf. Shanghai*, 2021. 645–650 (in Chinese). [doi: [10.26914/c.cnkihy.2021.029288](https://doi.org/10.26914/c.cnkihy.2021.029288)]



- [39] Ma MY, Stankovic J, Bartocci E, Feng L. Predictive monitoring with logic-calibrated uncertainty for cyber-physical systems. *ACM Trans. on Embedded Computing Systems*, 2021, 20(5S): 101. [doi: [10.1145/3477032](https://doi.org/10.1145/3477032)]
- [40] Bartocci E, Deshmukh J, Donzé A, Fainekos G, Maler O, Ničković D, Sankaranarayanan S. Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications. In: Bartocci E, Falcone Y, eds. *Lectures on Runtime Verification: Introductory and Advanced Topics*. Springer, 2018. 135–175. [doi: [10.1007/978-3-319-75632-5\\_5](https://doi.org/10.1007/978-3-319-75632-5_5)]
- [41] a MY, Gao J, Feng L, Stankovic J. STLnet: Signal temporal logic enforced multivariate recurrent neural networks. In: *Proc. of the 34th Int'l Conf. on Neural Information Processing Systems*. Vancouver: Curran Associates Inc., 2020. 14604–14614.
- [42] Jha S, Tiwari A, Seshia SA, Sahai T, Shankar N. TeLEx: Learning signal temporal logic from positive examples using tightness metric. *Formal Methods in System Design*, 2019, 54(3): 364–387. [doi: [10.1007/s10703-019-00332-1](https://doi.org/10.1007/s10703-019-00332-1)]
- [43] Camargo M, Dumas M, González-Rojas O. Automated discovery of business process simulation models from event logs. *Decision Support Systems*, 2020, 134: 113284. [doi: [10.1016/j.dss.2020.113284](https://doi.org/10.1016/j.dss.2020.113284)]
- [44] Wang ZN, Cheng B, Zhang WK, Chen JL. Many-objective automatic service composition based on temporal goal decomposition. *IEEE Trans. on Network and Service Management*, 2021, 18(3): 3813–3828. [doi: [10.1109/TNSM.2021.3056414](https://doi.org/10.1109/TNSM.2021.3056414)]
- [45] López-Pintado O, Dumas M. Business process simulation with differentiated resources: Does it make a difference? In: *Proc. of the 20th Int'l Conf. on Business Process Management*. Münster: Springer, 2022. 361–378. [doi: [10.1007/978-3-031-16103-2\\_24](https://doi.org/10.1007/978-3-031-16103-2_24)]
- [46] Zheng ZB, Zhang YL, Lyu MR. Investigating QoS of real-world Web services. *IEEE Trans. on Services Computing*, 2014, 7(1): 32–39. [doi: [10.1109/TSC.2012.34](https://doi.org/10.1109/TSC.2012.34)]
- [47] Zhou ZB, Zhao D, Hancke G, Shu L, Sun YC. Cache-aware query optimization in multiapplication sharing wireless sensor networks. *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, 2018, 48(3): 401–417. [doi: [10.1109/TSMC.2016.2598398](https://doi.org/10.1109/TSMC.2016.2598398)]
- [48] Zhou ZB, Fang W, Niu JW, Shu L, Mukherjee M. Energy-efficient event determination in underwater WSNs leveraging practical data prediction. *IEEE Trans. on Industrial Informatics*, 2017, 13(3): 1238–1248. [doi: [10.1109/TII.2017.2669967](https://doi.org/10.1109/TII.2017.2669967)]
- [49] Aasi E, Vasile CI, Bahreinian M, Belta C. Classification of time-series data using boosted decision trees. In: *Proc. of the 2022 IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*. Kyoto: IEEE, 2022. 1263–1268. [doi: [10.1109/IROS47612.2022.9982105](https://doi.org/10.1109/IROS47612.2022.9982105)]
- [50] Mehdipour N, Vasile CI, Belta C. Specifying user preferences using weighted signal temporal logic. *IEEE Control Systems Letters*, 2021, 5(6): 2006–2011. [doi: [10.1109/LCSYS.2020.3047362](https://doi.org/10.1109/LCSYS.2020.3047362)]

#### 附中文参考文献:

- [5] 杨文青, 刘广钟. 智慧城市中 QOE 感知服务组合策略的研究. *计算机技术与发展*, 2021, 31(4): 176–181. [doi: [10.3969/j.issn.1673-629X.2021.04.030](https://doi.org/10.3969/j.issn.1673-629X.2021.04.030)]
- [38] 黄钰翀, 李杰, 王楠, 等. 一种基于局部线性时序逻辑任务描述的多机器人协同规划方法. 见: 第 40 届中国控制会议. 上海, 2021. 645–650. [doi: [10.26914/c.cnkihy.2021.029288](https://doi.org/10.26914/c.cnkihy.2021.029288)]



周长兵(1974—), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为服务计算, 物联网, 边缘计算.



孙晓(1988—), 男, 博士, 助理研究员, 主要研究领域为群体智能, 智能群体行为建模与演化动力分析.



赵登(1991—), 女, 博士, 讲师, CCF 专业会员, 主要研究领域为服务计算, 物联网, 形式化方法.



薛霄(1979—), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为服务计算, 计算实验, 群体智能.



张文博(1976—), 男, 博士, 研究员, 博士生导师, CCF 专业会员, 主要研究领域为云计算, 服务计算.