

doi:10.11835/j.issn.1000-582X.2020.233

# 分布式时空相关服务网络构建及其 在 IOT 服务发现中的应用

杨冬菊<sup>a,b</sup>, 赵卓峰<sup>a,b</sup>

(北方工业大学 a.大规模流数据集成与分析技术北京市重点实验室;b.数据工程研究院,北京 100144)

**摘要:**随着 IOT(Internet of Things)服务的规模不断扩大,IOT 服务的分布式管理是必然趋势。基于时空相关属性快速发现一组满足要求的 IOT 服务是 IOT 服务管理与使用中要解决的首要问题。传统的根据语法或语义进行服务发现的方法通常面对的是海量、离散的服务集合,服务发现时间与服务规模关系密切,并且在多次迭代递进的服务发现请求中结果无法重用。针对上述问题,提出了一种分布式时空相关服务网络的构建方法,利用服务之间的时空聚合关系与时空连接关系在离散、独立的服务之间建立关联,从而通过构建时空相关服务网络对服务进行组织与管理,重点探讨分布式环境下时空相关服务网络的构建、演化方法,以高速公路紧急事件处置场景作为案例,讨论基于服务网络的服务快速发现。

**关键词:**IOT 服务;服务发现;时空关联关系;服务网络;时空相关服务网络

**中图分类号:**TP311

**文献标志码:**A

**文章编号:**1000-582X(2020)07-006-13

## Construction of distributed spatial-temporal correlation service network and its application to IOT service discovery

YANG Dongju<sup>a,b</sup>, ZHAO Zhuofeng<sup>a,b</sup>

(a. Beijing Key Laboratory on Integration and Analysis of Large-scale Stream Data;

b. Institute of Date Engineering, North China University of Technology, Beijing 100144, P. R. China)

**Abstract:** With the rapid increase in the number of IOT (Internet of Things) services, distributed management of IOT services is an inevitable trend. Quick discovery of a set of IOT services based on spatial-temporal properties is the primary issue in the management and use of IOT services. The traditional method of service discovery based on syntactic or semantics usually faces a mass and isolated set of services, and the service discovery time is closely related to the service number. Besides, the result cannot be reused in the service discovery request with multiple iterations. To solve the above problems, this paper proposed a method for distributed spatial-temporal properties related service network building, using the paradigmatic relations of spatial-temporal properties to build service connection relationship between the isolated services. The services were organized and managed by constructing spatial-temporal related service network. The focus of this research was on the construction and evolution of spatial-temporal related service network in distributed environment. Finally, the rapid discovery of services based on service

**收稿日期:**2019-12-13

**基金项目:**国家自然科学基金面上资助项目(61672042)。

Supported by the Projects of the National Natural Science Foundation of China(61672042).

**作者简介:**杨冬菊(1975—),女,副研究员,主要从事服务计算、云计算、大数据研究,(E-mail)yangdongju@ncut.edu.cn。

network was discussed in the case of highway emergency handling scenario.

**Keywords:** IOT service; service discovery; spatial-temporal properties; service network; spatial-temporal correlation service network

利用 SOA(service-oriented architecture)技术实现物联网的物端与云端松耦合协同是近年来的研究热点<sup>[1]</sup>。由于智能设备数量不断增加,提供的物联网服务的规模也在不断扩大。有报道显示,2020 年物联网设备数量将达到 204 亿。如此规模巨大的物联网服务,集中式的服务注册与管理显然无法满足要求,分布式架构下构建服务库实现 IOT 服务发现是必然趋势。

目前常用的服务发现方法主要分为 3 种,一是先通过概率方法或过滤条件缩小服务搜索的范围,然后对结果集中进行服务匹配<sup>[2]</sup>;二是使用机器学习或特征提取技术提取出服务的一些典型特征,根据特征进行相似性度量,筛选出服务列表,再对列表进行融合和汇总<sup>[3-5]</sup>;三是首先根据部分服务特征对服务进行分类、聚类或建立关联网络,然后以此为基础进行服务发现<sup>[6-7]</sup>。上述工作主要聚焦在集中管理环境中的服务发现方法。在分布式环境中,应用 Chord 协议构建服务存储与管理的分布式架构是目前比较常用的解决方案。文献[8]提出了基于 Chord 的带有 QoS 的语义 Web 服务发现系统,将服务质量(QoS, quality of service)信息加入到 OWL-S(web ontology language for service)中来描述服务,并称之为 OWL-QoS,采用基于 Chord 的分布式方式存储元数据,通过基于 OWL-QoS 的服务匹配算法来查询服务。文献[9]提出一种面向语义 Web 服务发现的混合式 P2P(peer-to-peer)网络 JXord,该网络基于 JXTA 和 Chord 协议,采用分层结构,上层利用 Chord 协议进行管理,下层采用 JXTA 分组思想,基于 JXord 进一步提出了语义 Web 服务发布与发现的实现过程。这些分布式环境下的相关工作主要集中在架构设计与服务路由方面,并未考虑物联网背景下服务规模巨大带来的服务发现效率问题。

时空相关性是物联网服务区别于传统 SOA 服务的一大典型特征<sup>[10]</sup>,它反映的是物联网服务及其背后的智能设备在可用时间及空间位置方面的属性。基于时空相关属性快速发现一组满足要求的 IOT 服务是 IOT 服务管理与使用中要解决的首要问题。传统的根据语法或语义进行服务发现的方法通常面对的是海量、离散的服务集合,服务发现时间与服务规模关系密切,并且在多次迭代递进的服务发现请求中结果无法重用。

针对上述问题,提出了一种分布式时空相关服务网络的构建方法,首先利用服务之间的时空聚合关系与时空连接关系在离散、独立的服务之间建立关联,从而通过构建时空相关服务网络对服务进行组织与管理,其次重点探讨分布式环境下时空相关服务网络的构建、演化方法,以高速公路紧急事件处置场景为案例讨论时空相关服务网络的构建及基于服务网络的服务发现方法,分析文中方法的有效性。

## 1 问题概述

物联网服务提供的数据和操作在时间和空间上具有一定的相关性,比如高速公路上,某个收费站与相邻地点的摄像头、收费站等监测数据具有一定的空间相关性,并且某个时间点的监测数据与过去或未来近几个时间点的监测数据也具有一定的相关性,从而使用户在发现服务时可以通过一定的相关推断来获得近似的一组服务。在某些突发事件情况下,服务的时间和位置往往成为服务发现的关键属性,需要基于服务的时空属性能够快速筛选出一组服务。

当高速公路上发生紧急事故时,需要在短时间内快速筛选出距离事故点一定范围内的摄像头、收费站以及该范围内在事故发生时间点之前一定时间的所有路段或收费站的车流量或车速,根据事故紧急情况对事故作出相应的处置方案。场景如图 1 所示。流程如下:首先对事故数据进行采集,包括事故发生的时间、地点、当前路段的流量和车速、最近几个收费站的车流量、当前的天气情况、事故地点附近的收费站或路段监控摄像头,根据这些数据对事故级别进行判定,包括车辆损伤程度、有无人员伤亡、有无恶劣天气、有无道路损毁等,然后对事故进行相应的处置,调度一定范围内的应急车辆、管控收费站及路段车辆。如果事故升级,则需进一步扩大摄像头、收费站、应急车辆的调度范围,调整收费站和路段的管控范围。

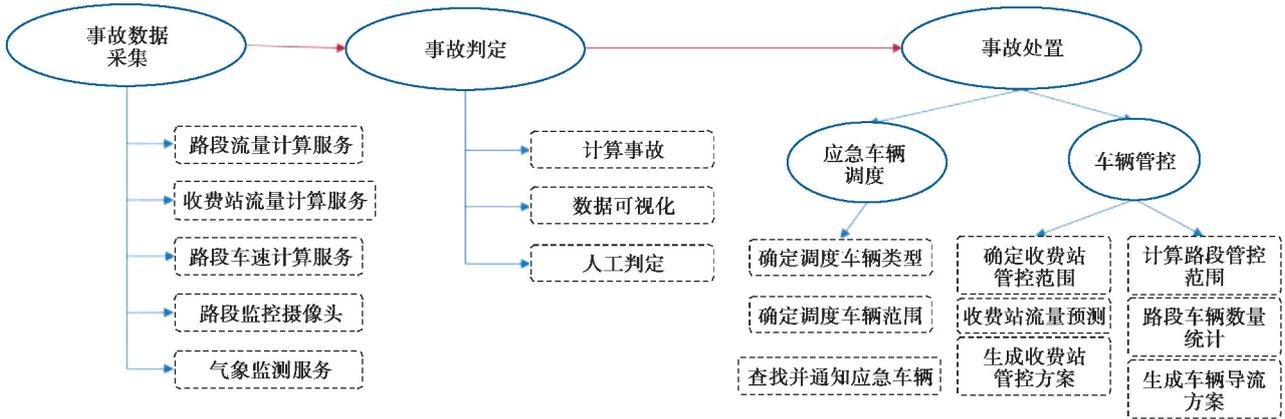


图 1 高速公路紧急事件处置场景

Fig. 1 Scenario of highway emergency

在此过程中,要解决的核心问题是根据时空需求快速获取相关服务。其中涉及的服务包括两类,一类是由终端设备封装的实体服务,如收费站服务、摄像头服务、气象监测服务、警车服务、清障车服务、救护车服务等,通过这些服务能够不间断地获取数据;另一类是功能服务,如计算收费站或路段内的车流量、计算路段内的平均车速、数据可视化展示等。接下来分析要实现基于时空属性的服务的快速发现,需要解决的几个挑战性问题。

1)在此场景下,服务的首要特征是具有时间属性和空间位置属性,需要根据服务的时空相关属性过滤掉无关服务,缩小服务发现范围。所以不管针对实体服务还是功能服务,如何刻画、描述服务的时空相关属性、设计一套具有时空相关属性的 IOT 服务模型是要解决的首要问题。

2)传统的根据语法或语义进行服务发现的方法通常面对的是海量、离散的服务集合,服务发现时间与服务规模关系密切,并且在多次迭代递进的服务发现请求中结果无法重用。比如在图 1 的场景中,事故处置人员可能会根据事故进展不断扩大服务的查找范围,每次进行服务的全局检索显然无法满足事故处理的时效性要求。对场景进一步分析发现,在服务之间存在一定的时空相关关系,比如查找距离事故地点最近的收费站,既需要考虑服务之间的空间距离,又需要考虑服务的可用性,即围绕突发事件的收费站服务、摄像头服务、警车服务等多个服务之间存在时空相关关系,如何发现服务之间的时空关联关系并进行刻画和建模,使得海量 IOT 服务从无序变为有序,进而构建一套时空相关服务网络是提升服务发现效率要解决的关键问题。

3)服务的分布式管理环境使得服务网络的构建、演化更加复杂。

## 2 时空相关服务网络模型

在物联网环境中,服务的时间和空间位置属性往往成为服务发现的关键属性,需要基于服务的时空属性能够快速筛选出一组服务。为了解决海量服务发现带来的时间开销问题,首先根据时空关联关系对服务进行组织,构建网状结构的时空相关服务网络,然后将服务发现问题转化为图的遍历与查找问题,从而减少服务规模对服务发现效率的影响。服务模型是服务发现、选择和组合的基础,是整个服务生态系统的基点。本章首先定义服务的时空相关属性,在已有服务模型基础上设计了一套带有时空相关属性的服务模型,并在此基础上进一步设计了时空相关服务网络模型。

### 2.1 带有时空属性的 IOT 服务模型

目前在 IOT 服务模型方面的研究大多考虑物联网设备、环境与服务的作用关系,比较典型的是 DPWS<sup>[11]</sup>(devices profile for web services),它定义了设备和服务之间的规范,但是它以实体设备为中心,将物联网服务作为实体设备的一部分进行描述,无法很好地描述多个设备提供同一服务的情况。目前比较常

用的方法是将服务与实体设备分开描述,在两者之间建立关联关系,实体设备提供了服务,服务作用于实体设备<sup>[12-13]</sup>。时空相关属性大多通过实体设备的位置和可用时间属性进行描述<sup>[14]</sup>。

OWL-S<sup>[15]</sup>是一种用于 Web 服务描述的本体语言,目前在 IOT 服务描述方面仍然比较活跃,它包括 3 个部分:①ServiceProfile,用于说明服务的功能,定义服务的名字、描述、输入/输出语义;②ServiceModel,描述服务的具体实现细节,定义输入和输出的参数;③ServiceGrounding,定义用户与服务交互时的细节,包括入口点和参数等。在物联网环境中,服务主要以资源受限的实体服务为主,传统 Web 服务为次,因此在选择服务描述模型时重点考虑其轻量性,减小资源消耗,同时仍需要保证它能够描述传统的 Web 服务。所以文中重点考虑在 OWL-S 模型基础上扩展服务的时空相关属性,如图 2 所示。

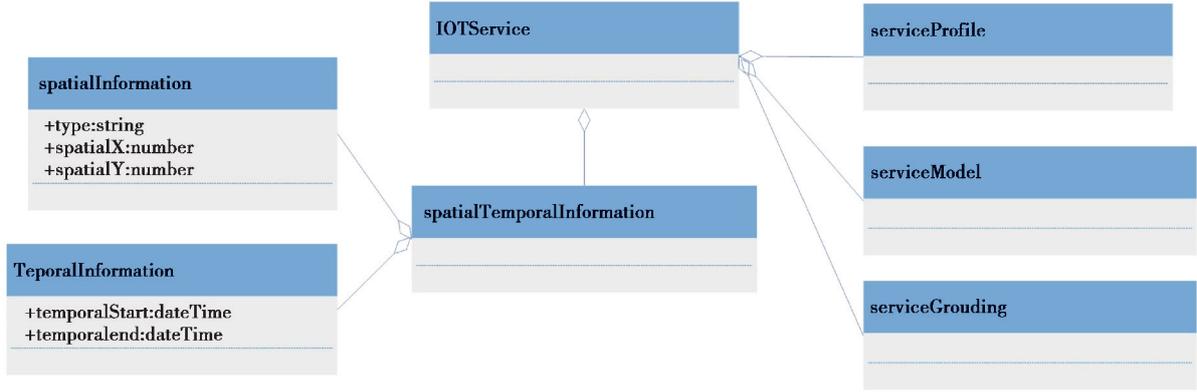


图 2 带有时空属性的 IOT 服务模型

Fig. 2 IOT service model with spatial-temporal properties

**定义 1** IOT 服务:一个 IOT 服务指在物联网环境下将智能物品或设备提供的功能和资源与现有互联网环境中的计算、存储和处理资源都统一抽象形成服务,包括物端资源提供的实体服务和云端资源提供的数据融合、数据分析、数据可视化等各种功能服务。一个 IOT 服务可以描述为一个四元组: $S_{\text{IOT}} = \langle P_{\text{rof}}, P_{\text{roc}}, G, P_{\text{ST}} \rangle$ 。其中, $P_{\text{rof}}$ 指服务的基本信息,包括服务名字、描述、功能、输入输出等; $P_{\text{roc}}$ 指服务流程,定义服务输入、输出的参数; $G$ 定义用户与服务交互时的细节,包括入口点和参数等; $P_{\text{ST}}$ 用于定义服务的时空相关属性。

$P_{\text{rof}}, P_{\text{roc}}, G$  的详细信息可以参照文献<sup>[15]</sup>中的详细描述,接下来重点探讨  $P_{\text{ST}}$  的定义。

$$P_{\text{ST}} = \langle S_*, T_* \rangle,$$

式中: $S_*$ 表示服务的空间位置属性, $T_*$ 表示服务的时间属性。

$$S_* = \langle P_{\text{type}}, P_{\text{XY}} \rangle,$$

式中: $P_{\text{type}}$ 指服务空间位置属性的类型,

$$P_{\text{type}} \in \{“D”, “L”, “A”\},$$

其中,“D”表示服务的位置属性为一个固定点,比如收费站服务、摄像头服务的位置;“L”表示服务的位置属性为一条线,比如高速上的警车、救护车、计算某路段的车流量等;“A”表示服务的位置属性为一个区域,比如在城市道路上移动的警车、救护车等。为了简化服务表示及后续的计算,将服务位置区域简化为一个矩形。

$$S_* = \begin{cases} \langle “D”, ((x_1, y_1), (x_2, y_2)) \rangle; x_1 = x_2, y_1 = y_2, (x_1, y_1) \text{ 表示位置点,} \\ \langle “L”, ((x_1, y_1), (x_2, y_2)) \rangle; ((x_1, y_1), (x_2, y_2)) \text{ 表示线段的 2 个顶点, } x_1 \leq x_2, y_1 \leq y_2, \\ \langle “A”, ((x_1, y_1), (x_2, y_2)) \rangle; ((x_1, y_1), (x_2, y_2)) \text{ 表示矩形的 2 个对角顶点, } x_1 \leq x_2, y_1 \leq y_2, \end{cases}$$

式中: $T_* = \langle T_1, T_2, \dots, T_m \rangle$ ,其中  $m$  表示有  $m$  个可用时间段或时间范围; $T_i = \langle S_i, E_i \rangle, 1 \leq i \leq m$ ,  $S_i$  和  $E_i$  分别表示开始时间和结束时间。

## 2.2 时空相关服务网络模型

在 2 个或多个时空属性相关联的服务之间建立关联关系,则构成时空相关服务网络。图 3 是一个时空

相关服务网络模型实例。考虑到服务的空间位置通常比较固定,而服务随时间变化比较频繁,所以为了减少服务网络的生长与演化开销、在构造时空相关服务网络时将服务的空间属性和时间属性分开考虑,首先根据服务的空间属性关联关系构建服务网络,其次突发事件情况下根据服务的时间属性完成服务网络的修剪与演化。

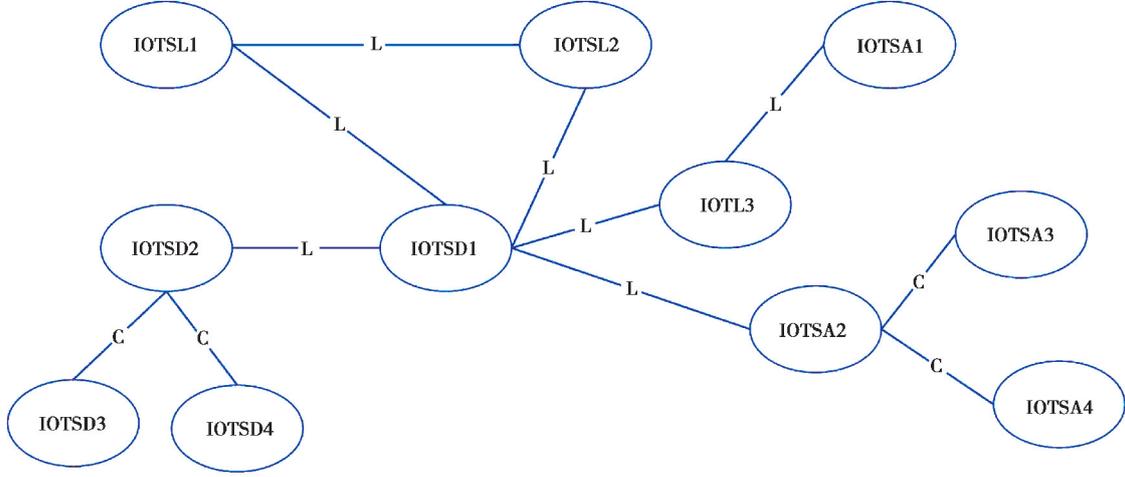


图 3 时空相关服务网络模型实例

Fig. 3 Examples of spatial-temporal properties related service network

**定义 2** IOT 服务的空间位置距离:用来衡量 2 个 IOT 服务之间的空间位置关系,从而度量服务之间的时空相关性,记作  $D_{\text{dist}}(S_{\text{IOT}_i}, S_{\text{IOT}_j})$ 。

距离度量用于权衡目标在多维空间中的距离,即分歧程度,经过计算如果距离结果越小(不小于 0)则说明目标间契合程度越强。典型的距离计算方法有欧几里得距离(Euclidean distance)、明可夫斯基距离(Minkowski distance)、曼哈顿距离(Manhattan distance)、切比雪夫距离(Chebyshev distance)等。笔者前期对几种距离的计算结果进行比较,选定欧式距离<sup>[16]</sup>作为 2 个 IOT 服务的空间位置距离。定义为

$$D_{\text{dist}}(S_{\text{IOT}_i}, S_{\text{IOT}_j}) = \sqrt{\sum_{k=1}^m (S_{\text{IOT}_i} \cdot P_{\text{ST}} \cdot S_* \cdot P_{\text{XY}_k} - S_{\text{IOT}_j} \cdot P_{\text{ST}} \cdot S_* \cdot P_{\text{XY}_k})^2}, \quad (1)$$

式中,  $1 \leq i, j \leq n$ ,  $S_{\text{IOT}_i}, S_{\text{IOT}_j} \in V$ ,  $S_{\text{IOT}_i} \cdot P_{\text{ST}} \cdot S_* \cdot P_{\text{XY}_k}$  表示服务  $S_{\text{IOT}_i}$  的第  $k$  个空间位置属性,  $m$  表示服务的空间位置属性的个数。

如前所述,服务的空间位置属性包含点状、线状和区域 3 种类型。为了简化服务空间位置距离计算,对不同类型的服务做了归一化处理。

如果  $S_{\text{IOT}_i} \cdot P_{\text{ST}} \cdot S_* \cdot P_{\text{XY}} \cdot P_{\text{type}} = "D"$ , 将  $S_{\text{IOT}_i} \cdot P_{\text{ST}} \cdot S_* \cdot P_{\text{XY}}$  构成一个  $m$  阶向量集合,表示为:

$$S_{\text{IOT}_i} \cdot P_{\text{ST}} \cdot S_* \cdot P_{\text{XY}} = \{(x_1, y_1), (x_1, y_1), \dots, (x_1, y_1)\}.$$

如果  $S_{\text{IOT}_i} \cdot P_{\text{ST}} \cdot S_* \cdot P_{\text{XY}} \cdot P_{\text{type}} \in \{"L", "A"\}$ , 对于服务空间位置属性为线状或区域的服务,将  $S_{\text{IOT}_i} \cdot P_{\text{ST}} \cdot S_* \cdot P_{\text{XY}}$  构成一个  $m$  阶向量集合,表示为:

$$S_{\text{IOT}_i} \cdot P_{\text{ST}} \cdot S_* \cdot P_{\text{XY}} = \{(x_1, y_1), \dots, (x_j, y_j), \dots, (x_l, y_l), \dots, (x_2, y_2)\}, 1 \leq l, j \leq m-2, x_j \leq x_l, y_j \leq y_l,$$

式中:  $(x_j, y_j)$  和  $(x_l, y_l)$  是在  $(x_1, y_1)$  和  $(x_2, y_2)$  组成的线段或区域中随机选取的点的坐标,并按照大小排序。

**定义 3** 时空相关服务网络  $N_{\text{STSN}}$ :是记录服务之间时空关联关系的网络。它可表示为一个无向图

$$N_{\text{STSN}} = \langle V, E \rangle,$$

式中:  $V$  是无向图的顶点集,  $V = \{S_{\text{IOT}_1}, S_{\text{IOT}_2}, \dots, S_{\text{IOT}_n}\}$ ,  $S_{\text{IOT}_i} \in S_{\text{IOT}}, 1 \leq i \leq n$ ;  $E$  是无向图的边集,  $E = V \times V = \{(S_{\text{IOT}_i}, S_{\text{IOT}_j}, R) \mid S_{\text{IOT}_i}, S_{\text{IOT}_j} \in V, S_{\text{IOT}_i} \sim S_{\text{IOT}_j} (S_{\text{IOT}_i} \text{ 和 } S_{\text{IOT}_j} \text{ 存在某种时空关联关系}), R \text{ 是关系的类}$

型,  $1 \leq i, j \leq n$  }, 其中,  $R \in \{“C”, “L”\}$ , 当  $R = “C”$  时, 表示 2 个服务具有时空聚合关系, 当且仅当  $D_{\text{dist}}(S_{\text{IoT}_i}, S_{\text{IoT}_j}) \leq \delta_1$ ,  $\delta_1$  是具有聚合关系的服务空间位置距离的阈值, 记做:  $R(S_{\text{IoT}_i}, S_{\text{IoT}_j}) = C$ 。比如图 3 中的 IOTD2 和 IOTD3、IOTD4 具有时空聚合关系。时空聚合关系满足如下性质:

如果  $R(S_{\text{IoT}_i}, S_{\text{IoT}_j}) = C$ ,  $R(S_{\text{IoT}_i}, S_{\text{IoT}_k}) = C$ , 则  $R(S_{\text{IoT}_j}, S_{\text{IoT}_k}) = C$ 。

当  $R = “L”$  时, 表示 2 个服务具有时空连接关系, 当且仅当  $\delta_1 < D_{\text{dist}}(S_{\text{IoT}_i}, S_{\text{IoT}_j}) \leq \delta_2$ ,  $\delta_2$  是具有连接关系的服务空间位置距离的阈值, 记做:  $R(S_{\text{IoT}_i}, S_{\text{IoT}_j}) = L$ 。比如, 图 3 中的 IOTD2 和 IOTD1 之间具有时空连接关系。

**定义 4** 时空相关服务网络轮廓图  $P_{\text{STSN}}$ : 用来定义一个服务网络的服务区域或范围。它是由 2 个顶点构造的矩形区域。

由  $n$  个节点组成的服务网络图  $N_{\text{STSN}} = \{S_{\text{IoT}_1}, S_{\text{IoT}_2}, \dots, S_{\text{IoT}_n}\}$ , 则

$$\begin{aligned}
 P_{\text{STSN}} &= \langle (x_1, y_1), (x_2, y_2) \rangle, \\
 x_1 &\leq \forall S_{\text{IoT}_i}. P_{\text{ST}}. S_*. P_{\text{XY}}. x_k, \\
 y_1 &\leq \forall S_{\text{IoT}_i}. P_{\text{ST}}. S_*. P_{\text{XY}}. y_k, \\
 x_2 &\geq \forall S_{\text{IoT}_i}. P_{\text{ST}}. S_*. P_{\text{XY}}. x_k, \\
 y_2 &\geq \forall S_{\text{IoT}_i}. P_{\text{ST}}. S_*. P_{\text{XY}}. y_k, \quad 1 \leq i \leq n, \quad 1 \leq k \leq 2.
 \end{aligned}$$

### 3 分布式时空相关服务网络构建

#### 3.1 服务库的分布式架构

物联网环境下服务分布、规模大, 且随着网络及物联网技术发展会有持续的爆炸性增长, 采用集中式架构进行服务的注册与管理显然无法满足服务的实时处理需求, 而传统的 P2P 无中心的分布式架构在服务发现时会带来巨大的网络开销。参考文献[8-9]中的分布式服务架构, 参考 DNS(domain name system)架构、Chord 协议<sup>[17]</sup>设计了一套分层分组的分布式架构服务库, 如图 4 所示。

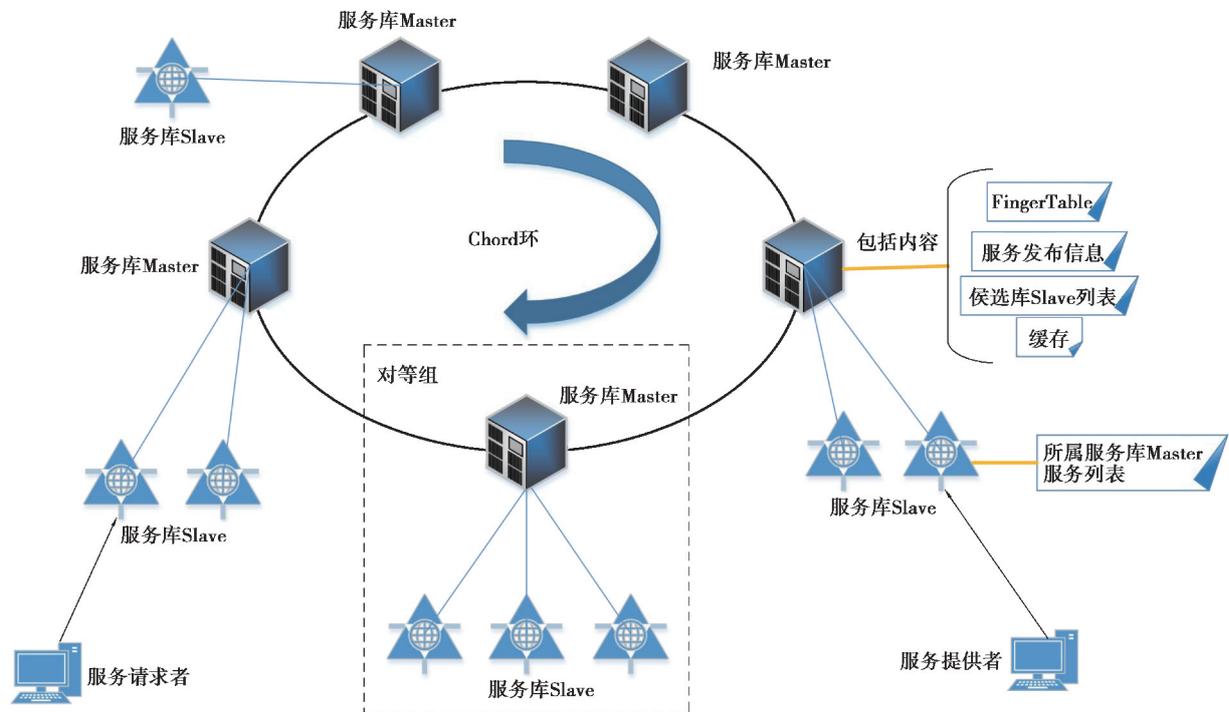


图 4 服务库的分布式架构

Fig. 4 Distributed architecture of service registries

一组具有相似特征的服务库组成一个对等组, 分为一个 Master 节点和若干 Slave 节点, Slave 中保存各

服务的元数据信息, Master 中保存本组所有 slave 节点中的服务目录, 并设置 Master 不可用状态下的备选 Master 列表。为了提升服务的分布式发现效率, 所有对等组的 Master 节点连接成一个环, 每个节点维护一个前驱后继地址表, 以便能迅速遍历服务库发现所需服务。这种混合式体系结构, 充分利用了集中式和分布式两种体系结构的特点, 它在分布式体系结构的基础上, 将多个节点分组, 减少了全局服务发现的可能性, 从而降低了服务发现的通信开销和响应时间。同时, 它还能很好地应对随着服务数量和服务请求量增加而带来的可扩展性问题。

### 3.2 分布式时空相关服务网络构建

为了保证服务网络在分布式服务库中的连续性, 减少服务发现的时间开销, 在分布式时空相关服务网络构建时重点考虑遵循以下原则:

1) 尽量将具有时空关联关系的服务放到一个服务库对等组中, 也就是尽量保证一个对等组中存储的服务网络是连通的;

2) 服务数量很少时, 难以在服务之间建立起有效关联, 如果服务随机注册, 随着服务注册数量的增多, 则会导致多个分布式节点的环境中存在多个包含或交叉的服务网络, 会影响服务发现的性能。所以为了解决服务的冷启动问题, 服务库对等组创建时, 会初步设置当前对等组中的服务区域范围, 后期根据服务网络的生长情况, 该服务区域范围会逐步演化;

3) 每个 Slave 节点(记作  $S_{lave}$ )中, 存储时空相关服务网络图  $N_{STSN}$  和服务网络轮廓图  $P_{STSN}$ ,  $P_{STSN}$  用于快速判断当前服务库节点中服务网络的服务区域;

4) 每个 Master 节点(记作  $M_{aster}$ )中, 将对等组中所有 Slave 节点的所有服务网络的轮廓图进行合并, 得到当前对等组中所有服务网络的服务网络轮廓图  $P_{STSN}$ 。如果 Master 所在对等组中包含  $n$  个 Slave 节点, 则  $M_{aster}.P_{STSN} = S_{lave1}.P_{STSN} \cup S_{lave2}.P_{STSN} \cup \dots \cup S_{laven}.P_{STSN}$ 。 $M_{aster}.P_{STSN}$  依然表示成一个矩形区域。

当有新的服务加入或者服务节点退出时, 服务网络会发生变化。算法 1 和算法 4 分别是新服务加入算法和服务删除算法。值得注意的是, 随着时间变化, 服务的可用性会发生变化, 服务网络也会相应发生变化。这会带来巨大的服务网络更新开销。所以, 文中提出基于事件的服务网络演化算法, 见算法 5。

算法 1: 新服务加入服务库算法

输入: 新服务  $S_{IOT} = \langle P_{rof}, P_{roc}, G, P_{ST} \rangle$

输出: 新服务加入服务库的某一节点并构建服务网络

- 1) 随机找到一个服务库节点  $S_{lave1}$ ;
- 2)  $STMATCH(S_{IOT}, P_{ST}, S_{lave1}.P_{STSN})$  // 判断当前服务的时空属性是否在  $S_{lave1}$  的  $P_{STSN}$  区域内;
- 3) 如果是, 则跳转 4, 否则跳转 6;
- 4)  $insertS(S_{IOT}, S_{lave1}.N_{STSN})$  // 将新服务直接插入  $S_{lave1}$  中服务网络的合适位置, 采用深度优先或广度优先算法对服务网络图进行遍历, 找到合适位置插入服务;
- 5) 更新  $S_{lave1}.P_{STSN}$  与  $M_{aster1}.P_{STSN}$  ( $M_{aster1}$  是  $S_{lave1}$  所在对等组的主节点), 返回;
- 6) 找到  $S_{lave1}$  所在对等组的主节点  $M_{aster1}$ ;
- 7)  $STMATCH(S_{IOT}, P_{ST}, M_{aster1}.P_{STSN})$  // 判断当前服务的时空属性是否在  $M_{aster1}$  的  $P_{STSN}$  区域内;
- 8) 如果是, 则跳转 9, 否则跳转 12;
- 9)  $M_{aster1}$  返回与  $S_{IOT}$  服务区域相对应的 Slave 节点  $S_{lave2}$ ;
- 10)  $insertS(S_{IOT}, S_{lave2}.N_{STSN})$  // 将新服务直接插入  $S_{lave2}$  中服务网络的合适位置;
- 11) 更新  $S_{lave2}.P_{STSN}$  与  $M_{aster1}.P_{STSN}$ , 返回;
- 12) 根据 Chord 协议跳转找到  $M_{aster1}$  的后继地址  $M_{aster2}$ ;
- 13) 判断  $M_{aster2}$  的服务网络是否与当前服务的时空属性相匹配;
- 14) 如果不匹配, 重复步骤 12~13, 直到找到合适的服务库节点, 将新服务插入网络, 返回;
- 15) 如果所有服务库节点都不匹配, 则说明新服务不属于已有服务网络内部节点, 则根据原则(2)中提到的对等组的服务区域范围, 找到相应区域的对等组, 由对等组的 Master 节点决定当前服务加入哪个 Slave 节点的服务网络。

算法 2:  $STMatch(S_{IOT}.P_{ST}, S_{lave1}.P_{STSN})$  //判断当前服务的时空属性是否在  $S_{lave1}$  的  $P_{STSN}$  区域内

输入:  $S_{IOT}.P_{ST}, S_{lave1}.P_{STSN}$

输出: true 或 false

如果  $S_{lave1}.P_{STSN}.x_1 \leq S_{IOT}.P_{ST}.S*.P_{XY}.x_1, S_{IOT}.P_{ST}.S*.P_{XY}.x_2 \leq S_{lave1}.P_{STSN}.x_2 \& \& S_{lave1}.P_{STSN}.y_1 \leq S_{IOT}.P_{ST}.S*.P_{XY}.y_1, S_{IOT}.P_{ST}.S*.P_{XY}.y_2 \leq S_{lave1}.P_{STSN}.y_2$  则返回 true, 否则返回 false。

算法 3:  $insertS(S_{IOT}, S_{lave2}.N_{STSN})$  //将新服务直接插入  $S_{lave2}$  中服务网络的合适位置

输入:  $S_{IOT}, S_{lave2}.N_{STSN}$

输出: 新的  $S_{lave2}.N_{STSN}$

1)  $S_{TSNtemp} = S_{lave2}.N_{STSN}.V = \{S_{IOT1}, S_{IOT2}, \dots, S_{IOTm}\}$ ;

2) 如果  $R(S_{IOTi}, S_{IOTj}) = C$ , 则找到所有和  $S_{IOTi}$  和  $S_{IOTj}$  构成聚合关系的服务, 递归查找, 直到所有与  $S_{IOTi}$  和  $S_{IOTj}$  构成直接或间接聚合关系的服务都被找到, 构成集合  $I = \{S_{IOTi}, S_{IOTj}, \dots\}$ ;

3)  $S_{TSNtemp} = S_{TSNtemp} \ominus I \cup \{S_{IOTi}\}$  //将所有具有聚合关系的服务从处理集合中去除, 只保留 1 个;

4) 重复 2, 直到  $S_{TSNtemp}$  中的所有服务都不具有聚合关系;

5) 计算  $dist(S_{IOT}, S_{IOTi}) \mid S_{IOTi} \in S_{TSNtemp}$ , 并对结果进行排序;

6) 如果  $dist(S_{IOT}, S_{IOTi}) \leq \delta_1$ , 则更新  $S_{lave2}.N_{STSN}$ , 在  $S_{IOT}, S_{IOTi}$  之间增加一条边, 使得  $R(S_{IOTi}, S_{IOTj}) = C$ , 如果  $\delta_1 \leq dist(S_{IOT}, S_{IOTi}) \leq \delta_2$ , 则更新  $S_{lave2}.N_{STSN}$ , 在  $S_{IOT}, S_{IOTi}$  之间增加一条边, 使得  $R(S_{IOTi}, S_{IOTj}) = L$ ;

7) 返回  $S_{lave2}.N_{STSN}$ 。

算法 4: 在服务网络中删除一个服务节点

输入: 服务网络  $N_{STSN}$ , 待删除节点  $S_{IOT}$

输出: 删除节点后的新服务网络  $N_{STSN}$

1) 在  $N_{STSN}$  中检索  $S_{IOT}$ ;

2) 找到  $S_{IOT}$  的所有前驱和后继节点, 构成 2 个集合  $P_{IOTP}$  和  $C_{IOTC}$ ;

3) 分别取出  $P_{IOTP}$  和  $C_{IOTC}$  中的第一个服务节点, 根据 2.2 节中定义的几种时空关联关系判断两个节点是否存在聚合或连接关系, 如果存在, 则在两者之间建立关联关系;

4) 重复步骤 3, 直到  $P_{IOTP}$  和  $C_{IOTC}$  中的所有节点都进行了两两比较, 并建立了关联关系;

5) 删除  $S_{IOT}$  节点及与其关联的所有的边;

6) 输出当前服务网络  $N_{STSN}$ 。

算法 5: 基于事件的服务网络  $N_{STSN}$  演化算法

输入: 服务网络  $N_{STSN}$ , 事件发生时间  $t$

输出: 演化后的服务网络  $N_{STSN}$

1) 复制服务网络  $N_{STSN}$  为  $N_{STSN}''$  (保证  $N_{STSN}$  相对稳定, 不会随时间改变);

2) 按照深度优先或广度优先算法对  $N_{STSN}''$  进行遍历, 找到每一个服务  $S_{IOTi}$ ;

3) 如果  $t \notin S_{IOTi}.P_{ST}.T^*$ , 则说明当前服务处于不可用状态, 则调用算法 4 从  $N_{STSN}''$  中删除  $S_{IOT}$ ;

4) 重复步骤 3, 直到所有服务都处理完毕;

5) 输出当前服务网络  $N_{STSN}''$ 。

## 4 基于时空相关服务网络的服务发现实例验证

下面结合前述的高速公路紧急事件处置场景具体说明如何基于时空相关服务网络实现服务的快速发现。

### 4.1 IOT 服务列表

场景涉及的部分服务列表如表 1 所示。

表 1 高速公路紧急事件处置场景中涉及的部分服务  
Table 1 Some services in the Scenario of highway emergency

服务名称	服务类型	服务空间属性类型	说明
收费站服务	实体服务	D	由收费站封装的实体服务,可获得过车收据
监控摄像头服务	实体服务	D	由路段监控摄像头封装的实体服务,可获得车辆监控数据
警车	实体服务	L	提供警车服务,服务区域是高速上的路段(线)
救护车	实体服务	L	提供救护车服务,服务区域是高速上的路段(线)
路段车数量统计服务	功能服务	L	提供车辆数量统计功能,服务区域是路段(线)
路段车速计算服务	功能服务	L	提供车速计算功能,服务区域是路段(线)
收费站流量计算服务	功能服务	D	计算某收费站的车流量,服务区域和收费站点一样
事故级别判定服务	功能服务	*	根据输入计算事故级别,无固定服务区域
计算应急车辆调度类型及范围	功能服务	*	根据输入计算需要调度哪些应急车辆和调度范围,无固定服务区域
计算收费站管控范围	功能服务	*	根据输入计算收费站的管控范围,无固定服务区域
收费站流量预测	功能服务	D	预测收费站在某一时间段的车流量,服务区域和收费站点一样
生成收费站管控方案	功能服务	*	根据输入生成收费站的管控方案,无固定服务区域

4.2 时空相关服务网络

根据第 4 节描述,基于上述服务构建了一套分布式时空相关服务网络,并保证尽可能将连通性网络放在一个对等组中。考虑到在高速公路场景下服务位置之间存在一定的顺序关系,比如高速公路上的收费站之间存在先后顺序,一定是从一个收费站入口到一个收费站出口,所以采用有向图构建时空相关服务网络,边的方向代表服务之间的位置先后顺序。另外,为了保障方法的有效和实用,在时空相关服务网络中增加了服务的可达性验证,通过地图与时空相关服务网络相叠加去掉不可达的服务关联关系。图 5 是服务在地图上的位置图,图 6 是构建出的时空相关服务网络。

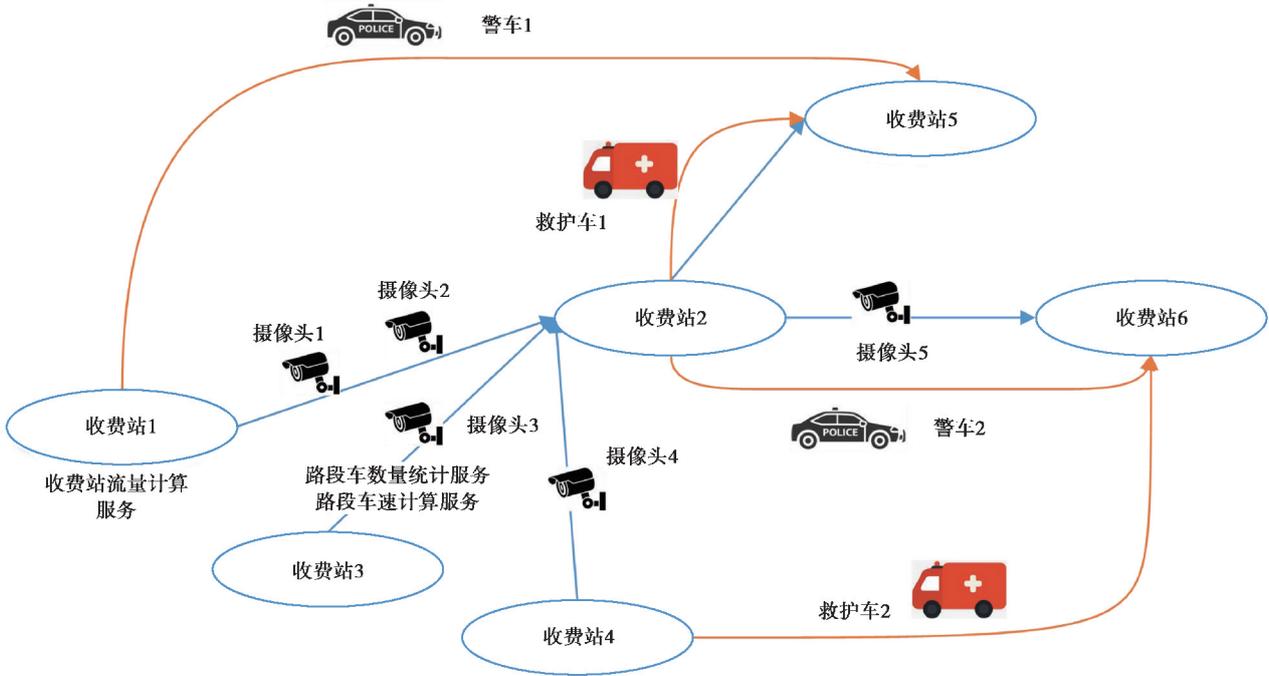


图 5 IOT 服务位置图

Fig. 5 IOT service location map

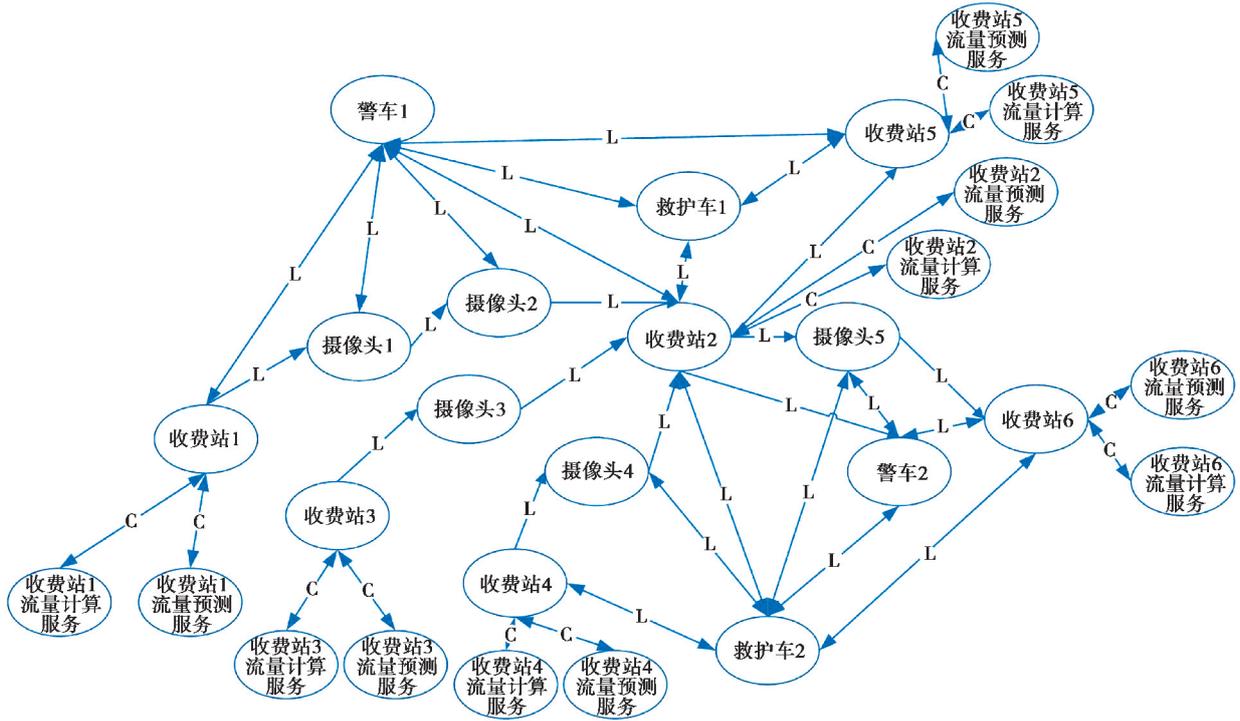


图 6 时空相关服务网络图

Fig. 6 Spatial-temporal properties related service network

### 4.3 基于服务网络的服务发现

假定在图 5 中,摄像头 5 和收费站 6 之间的某一点发生了事故。在图 1 所示场景中,任何一个服务都无法满足需求,需要快速发现一组服务进行相应的处置。基于服务网络的服务处置如算法 6 所示。

算法 6:基于服务网络的服务发现,基本思想是给定服务集合  $I$ ,在服务网络  $N_{STSN}$  中找到所有与  $I$  中元素具有连接关系的服务构成结果集 1,继续在  $N_{STSN}$  中查找与结果集 1 中所有服务具有聚合关系的服务集合,构成结果集 2,将结果集 2 中的服务及其关联关系构建子服务网络  $N_{STSN}'$ 。

输入:服务网络  $N_{STSN}$ ,服务集合  $I = \{S_{IOT_1}, S_{IOT_2}, \dots, S_{IOT_n}\}$ ,查找方向(正向、反向)

输出:一组服务集合,构成子服务网络  $N_{STSN}'$ ,有  $N_{STSN}' \subseteq N_{STSN}$

1)初始化  $N_{STSN}' = \emptyset$ ;

2)如果  $\{S_{IOT}\} \neq \emptyset$ ,取出节点  $S_{IOT_i} \in I$ ,否则返回  $N_{STSN}'$ ;

3)如果  $S_{IOT_i} \notin N_{STSN}.V$ ,则在  $I$  中删除  $S_{IOT_i}$ ,跳转步骤 2;

4)如果查找方向为正向 &&  $\exists S_{IOT_j} \in N_{STSN}.V \&\& R(S_{IOT_i}, S_{IOT_j}) = C$  或  $L$ ,则

$$\{N_{STSN}'.V = N_{STSN}'.V \cup \{S_{IOT_i}, S_{IOT_j}\};$$

$$N_{STSN}'.E = N_{STSN}'.E \cup (S_{IOT_i}, S_{IOT_j}, R), R = R(S_{IOT_i}, S_{IOT_j});$$

如果查找方向为反向 &&  $\exists S_{IOT_j} \in N_{STSN}.V \&\& R(S_{IOT_j}, S_{IOT_i}) = C$  或  $L$ ,则

$$\{N_{STSN}'.V = N_{STSN}'.V \cup \{S_{IOT_i}, S_{IOT_j}\};$$

$$N_{STSN}'.E = N_{STSN}'.E \cup (S_{IOT_j}, S_{IOT_i}, R), R = R(S_{IOT_j}, S_{IOT_i});$$

5)在  $I$  中删除  $S_{IOT_i}$ ,跳转步骤 2;

6)返回  $N_{STSN}'$ 。

在图 1 场景中应用上述服务发现算法过程如下:

1)输入事故发生位置和事故发生时间;

2)根据事故位置定位所属服务库节点区域;

3)根据事故发生时间利用算法 5 更新该服务库中的服务网络,保证服务网络上的所有服务都属于可用状态;

4)在服务网络中根据事故位置,查找距离事故点最近的摄像头服务,得到摄像头 5;

5)调用摄像头 5 服务查看事故情况,车辆损毁程度、有无人员受伤、有无车辆行驶不畅等;

6)调用算法 6,输入当前服务网络、{摄像头 5}、查找方向为反向,得到在服务网络中与“摄像头 5”反向关联的所有服务,得到服务集合{收费站 2,警车 2,救护车 2}及其位置关系,如图 7 所示子服务网络,推送给用户;

7)如果上述服务无法满足用户需求,则继续调用算法 6,输入当前服务网络、{收费站 2,警车 2,救护车 2}、查找方向为反向,以这些服务为中心,沿着箭头反方向查找服务,得到图 8 所示子服务网络,推送给用户;

8)重复上述操作,直到满足用户需求。

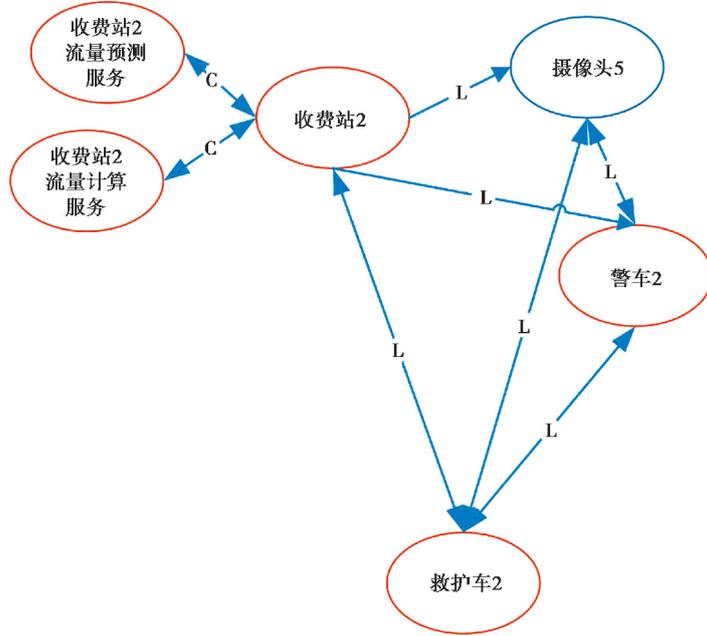


图 7 服务发现结果 1

Fig. 7 Service discovery result 1

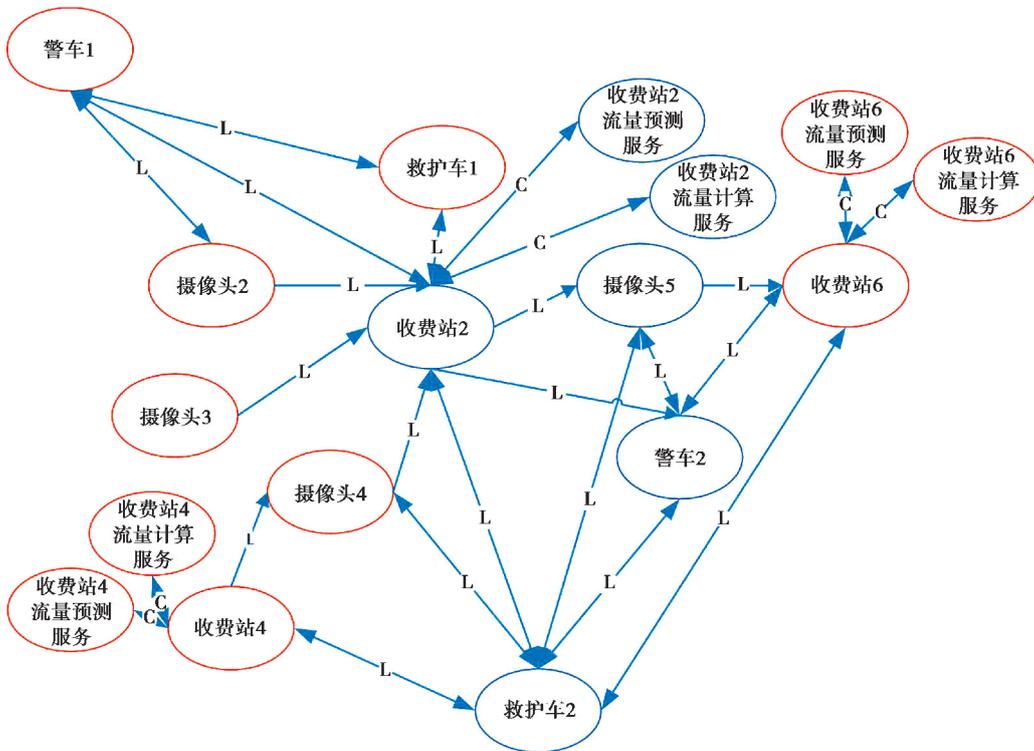


图 8 服务发现结果 2

Fig. 8 Service discovery result 2

#### 4.4 分析与讨论

在传统的基于关键词的服务发现或者基于语义的服务发现方法中,通常处理的是一些离散的服务,服务之间没有任何关联关系,所以需要全局检索,通常花费比较长的时间。而且如果用户对得到的服务集合结果不满意,需要进一步检索时,仍然需要进行全局检索,无法利用第一次的检索结果,这些都会造成服务发现的时间开销。

文中提出的方法重点关注服务的时空相关属性,在服务管理阶段首先基于服务的时空聚合关系和时空连接关系构建时空相关服务网络,使服务不再是分散、孤立的,而是形成了一张有方向、有连接的网络。将传统的服务发现问题,转化为图的遍历与检索问题。在服务发现时,能够从网络图中任意一点出发,迅速发现与之相关联的所有节点,如果得到的结果无法满足用户需求,则在当前结果基础上,进一步扩大查找范围,迅速发现与之相关联的所有节点,直至满足用户需求。在此过程中,避免了服务的全局匹配与查找,可以大大节省时间。

## 5 结 论

为了改善基于时空相关属性的 IOT 服务发现效率,提出时空相关服务网络模型,通过服务之间的时空聚合关系和时空连接关系将分散、独立的 IOT 服务连接起来形成有序的时空相关服务网络,将传统的基于语法或语义进行服务发现的方法转换成图中的节点查找问题,并重点讨论了分布式时空相关服务网络的构建与演化算法,最后通过高速公路紧急事件处置场景讨论基于服务网络的服务发现机制。接下来将进一步通过实验对上述结果进行验证。同时,假定服务库之间已经根据位置关系进行了划分,忽略了在多个服务库的服务网络中进行检索并对结果进行合并与融合的过程,这将是接下来的研究重点。

#### 参考文献:

- [1] Bouguettaya A, Singh M, Huhns M, et al. A service computing manifesto: the next 10 years[J]. Communications of the ACM, 2017, 60(4): 64-72.
- [2] Cassar G, Barnaghi P, Wang W, et al. A hybrid semantic matchmaker for iot services[C]//IEEE International Conference on Green Computing & Communications. New York, USA: IEEE Computer Society, 2012.
- [3] Cassar G, Barnaghi P, Moessner K. A Probabilistic Latent Factor approach to service ranking[C]//IEEE International Conference on Intelligent Computer Communication & Processing. New York, USA: IEEE, 2011.
- [4] Chen F Z, Lu C H, Wu H, et al. A semantic similarity measure integrating multiple conceptual relationships for web service discovery[J]. Expert Systems With Applications, 2017, 67: 19-31.
- [5] Fethallah H, Amine B, Amel H. Hybrid web service discovery based on fuzzy condorcet aggregation[C]//East European Conference on Advances in Databases and Information Systems. Cham: Springer International Publishing, 2015.
- [6] Adeleye O, Yu J, Yongchareon S, et al. Constructing and evaluating an evolving web-api network for service discovery[C]//International Conference on Service-Oriented Computing. Cham: Springer International Publishing, 2018.
- [7] Wu J, Chen L, Zheng Z B, et al. Clustering Web services to facilitate service discovery[J]. Knowledge and Information Systems, 2014, 38(1): 207-229.
- [8] 张莹, 黄厚宽, 杨冬, 等. 基于 Chord 的带有 QoS 的语义 Web 服务发现方法研究[J]. 电子与信息学报, 2009, 31(3): 711-715.  
ZHANG Ying, HUANG Houkuan, YANG Dong, et al. Research on chord-based semantic service discovery with QoS[J]. Journal of Electronics & Information Technology, 2009, 31(3): 711-715.(in Chinese)
- [9] 曹志强, 孟祥华, 邓水光. JXord: 一种面向语义 Web 服务发现的混合式 P2P 网络[J]. 小型微型计算机系统, 2013, 34(12): 2712-2717.

- CAO Zhiqiang, MENG Xianghua, DENG Shuiguang. JXord: a hybrid P2P network for semantic web service discovery[J]. Journal of Chinese Computer Systems, 2013, 34(12): 2712-2717.(in Chinese)
- [10] 陈海明, 石海龙, 李勣, 等. 物联网服务中间件: 挑战与研究进展[J]. 计算机学报, 2017, 40(8): 1725-1749.
- CHEN Haiming, SHI Hailong, LI Meng, et al. Service middleware for Internet of Things: challenges and approaches[J]. Chinese Journal of Computers, 2017, 40(8): 1725-1749.(in Chinese)
- [11] Devices Profile for Web Services Version 1.1, OASIS Standard, 2009. <http://docs.oasis-open.org/ws-dd/dpws/1.1/os/wsdd-dpws-1.1-spec-os.pdf>.
- [12] Yen I L, Bastani F, Hwang S Y, et al. From software services to IoT services: the modeling perspective[M] // Serviceology for Services. Cham: Springer International Publishing, 2017: 215-223.
- [13] Yen I L, Bastani F, Zhu W, et al. Service-Oriented IoT Modeling and Its Deviation from Software Services[C] // 2018 IEEE Symposium on Service-Oriented System Engineering (SOSE). New York, USA: IEEE Computer Society, 2018.
- [14] Huang B, Bouguettaya A, Neiat A G. Discovering Spatio-Temporal Relationships among IoT Services[C] // 2018 IEEE International Conference on Web Services (ICWS). New York, USA: IEEE, 2018.
- [15] Martin D, Burstein M, Hobbs J, et al. OWL-S: Semantic markup for web services[J]. W3C member submission, 2004, 22(4).
- [16] 王建新, 王柏人, 曲鸣, 等. 基于改进欧式距离的硬件木马检测[J]. 计算机工程, 2017, 43(6): 92-96.
- WANG Jianxin, WANG Boren, QU Ming, et al. Hardware Trojan detection based on improved euclidean distance[J]. Computer Engineering, 2017, 43(6): 92-96.(in Chinese)
- [17] Stoica I, Morris R, Karger D, et al. Chord: A scalable peer-to-peer lookup service for internet applications[J]. ACM SIGCOMM Computer Communication Review, 2001, 31(4): 149-160.

(编辑 詹燕平)