

Fig.2 Overall functional design of edge gateways and IoT devices

图2 边缘网关和物端设备的总体功能设计图

物端设备在加入网络时需要将自己的基本情况向边缘网关注册,包括但不限于地址、位置、监测目标的种类和产生资源的能力等。

边缘网关是物端设备和应用之间的中间层,为物端设备产生的动态资源提供访问接口,解析资源获取请求并将其转发至物端设备。边缘网关由响应模块、决策模块、URI解析模块、转发模块、物联网端设备注册和资源监测模块五个功能模块组成。响应模块接受上层应用发起的请求,并将请求结果返回应用。决策模块收到应用的请求决定将请求转发到具体哪个节点上。URI解析模块根据决策模块计算请求转发的具体节点,结合该节点的访问地址,将应用请求的URI解析为能够访问到的具体节点和具体资源。转发模块将URI解析模块解析出的节点请求地址将请求转发至指定节点。

1.3 状态机制

Sealink 是物联网应用层协议,是无状态的协议。每个请求都是完全单独处理的,没有连续性,不会维护历史请求和响应。Sealink 本质上由一个请求和一个响应组成,用户请求一个特定的URI,边缘网关会解析请求并将其转发至某些物联网节点,最后将获取的资源交付至用户,如果网络中不存在该资源则响应专用的提示信息。

Sealink 协议提供3种服务质量:0级代表请求者希望获取任意一个时间点的动态资源,边缘网关在物联网节点中获取任意时刻的资源值就返回;1级代表请求者希望获取某个时间范围内的请求资源,时间范围由请求者指定,边缘网关直到能获取到时间范围内的资源值时才返回;2级代表请求者希望获取动态资源的最新值,边缘网关将距离请求时间最近的资源值返回。用户可以根据实际应用需求选择合适的服务质量。

2 Sealink 协议的设计与实现

2.1 URI 与 RESTful API

通过URI访问动态资源的方式可以有效解决动态资源位置不断变化的问题。由于实时性和动态性的特点,物联网节点产生的传感数据通常以数据流的方式呈现。为了将物联网节点产生的动态资源统一地管理和描述,首先需要将传感数据结构化。在本方法中,传感数据按照系统-节点-感知器-数据的层次结构进行描述和组织。系统代表一组按照某种网络拓扑组织的物联网节点的集合,由一个边缘网关统一管理。内容包括该应用可访问的资源列表、节点的索引以及其他基本信息。节点是一个真实物联网节点的抽象,是多个传感数据的集合。在物联网节点部署之前需要为其指定配置参数,包括其通信方式、访问地址、描述、位置和拥有的传感器情况。感知器代表在一个应用系统中某个特定的感知模块。它的配置参数包括id、状态、数据类型、数据等。数据由感知器产生,以键值对的方式呈现,包括时间戳和对应的值。

Sealink协议的URI具体的定义方式如图3所示。当用户发起web请求后,应用根据URI中的HOST地址访问边缘网关,由边缘网关负责解析具体的请求内容。

```
Sealink-URI = "Sealink:" "/" host [ ":" port ] [node] [object] [resource]
```

Fig.3 Definition of URI in Sealink

图3 Sealink 协议中 URI 的定义

```
(1) Sealink://<host_address:port>/object1/picture
(2) Sealink://192.168.0.101:3300/object1/picture
(3) Sealink://< host_address:port >/node1/object1/picture
```

Fig.4 URI application instance in Sealink

图4 Sealink 协议中 URI 应用实例

层次化的结构能够在不同的级别下利用传感数据.图4展示了 Sealink URI 的几个应用实例.比如在一个物联网系统中获取其最新采集的监测目标 object1 的图像信息时,可以直接请求资源.此时访问的 URI 由图4中的(1)所示,边缘网关收到该 URL 请求时,假设边缘网关通过 IP 地址与端口号访问对应节点,则边缘网关根据其决策模块和 URI 解析模块将该地址解析为图4中的(2),节点收到该请求时会判断能产生图像信息的感知器是否存在对应 object1 的最新资源,将判断结果反馈至边缘网关.边缘网关通过收集整理节点的回复信息将应用请求的资源返回至应用.若应用需要请求确定节点采集的某个目标的图像信息,则此时访问的 URI 为图4中的(3),边缘网关会解析该 URI 并将请求转发至该节点.

利用 URI 描述资源的方式可以将请求物联网节点的过程对应用透明,特别是访问动态资源时,应用可以不明确动态资源当前所在的具体位置,降低了应用获取动态资源的复杂度,应用开发的工作量,提高了物联网资源的互操作性.

RESTful 设计风格是互联网 web 服务中应用最广泛的实现方案.本文提出的 Sealink 协议也基于 RESTful 设计风格为上层应用提供服务,并且针对动态资源的特点重新设计了 RESTful 接口的实现方法.Sealink 中 RESTful 方法接口及其对应的意义如下.

GET:获取某个动态资源;PUT:为节点新建生成动态资源的配置文件.与互联网不同,物联网中的资源只能由物联网节点通过对环境数据的采集来生成,而节点生成资源的参数选项、配置文件等则需要用户根据所需情况进行定制;POST:通过执行请求体中的代码修改资源;DELETE:删除某个资源.

通过对动态资源的获取操作封装为 RESTful 设计风格的 API,在开发上层应用程序时可以使用在互联网中应用最广泛的 HTTP 协议与边缘网关通信,访问、操作物联网中的资源.

2.2 动态资源的可访问规律建模与预测

当物联网系统监测的动态目标在节点间不断移动时,应用在访问到该目标资源之前可能无法得知目标存在的节点位置,资源的动态性使得获取该资源最新值的过程变得复杂和困难.使用发布-订阅的方法并缓存该动态资源的历史数据有时也并不能得到理想的结果,这是因为在根据缓存或者节点发布的信息对资源进行访问的这段时间内资源的位置可能又发生了变化,导致访问地址突然失效,特别是动态资源的位置在节点间变化较快时,对该资源的访问请求响应率就会下降.在资源位置改变导致访问地址失效的情况下,为了访问该资源的最新动态需要轮询网络中的节点直到获取到最新状态为止.然而,物联网节点的海量性使得轮询节点带来巨大的通信开销,在资源受限的物联网设备上可能是难以承受的.

为了提高动态资源的请求响应率,降低访问延时,本文提出的 Sealink 协议利用人工神经网络算法建立动态资源预测模型,进而在获取动态资源的过程中加入预测机制,解决获取动态资源所需要面对的问题.本协议首先使用机器学习算法利用动态资源在节点间移动历史数据提取特征值并进行建模,模型具有根据目标的历史信息预测其在某个节点产生的概率的能力.当节点监测到新的动态资源时,节点会根据之前建立的预测模型对资源下一时刻的位置进行预测,预测结果是网络中每个节点在下一时刻能监测到动态目标的概率.

Sealink 协议是这样利用预测结果的.边缘网关对所有节点产生多种动态资源的概率维护一个概率矩阵 P , 其中 $P_{i,j}$ 代表网络内节点 i 产生资源 j 的概率.当边缘网关收到节点 i 产生动态资源 j , 并且节点 i 预测下一时刻最可能产生资源 j 的节点为 k 时,边缘网关使用算式(1)更新节点 i 产生 j 的概率.

$$\widetilde{P}_{i,j} = \alpha P_{i,j} + (1 - \alpha), 0 \leq \alpha < 1 \quad (1)$$

在概率矩阵中 $P_{i,j}$ 更新后,使用算式(2)对所有节点产生动态资源 j 的概率进行归一化.

$$\overline{P}_{i,j} = \frac{P_{i,j}}{\sum_{k=0}^n P_{k,j}} (i = 0, 1, 2, \dots, n) \quad (2)$$

边缘网关还会使用相同的方法更新节点 k 的概率向量.通过调节 α 的值可以区别对待动态资源的产生节点的预测节点,比如若在收到节点 i 产生动态资源 j 时 α 取 0.1, 而收到产生动态资源 j 的节点为 k 时 α 取 0, 这样就意味着不采用预测结果, 仅仅统计生成动态资源 j 的节点情况, 经常能够产生动态资源的热区概率自然会逐渐增加.再比如若在收到节点 i 产生动态资源 j 时 α 取 0.1, 而收到产生动态资源 j 的节点为 k 时 α 取 0.8, 则意味着应用更信赖对动态资源的预测位置, 在下一时刻倾向于先访问模型运算出的预测节点.

在上层应用访问某个动态资源时边缘网关按照概率矩阵 P 中的概率值的大小依次访问节点, 直到获取请求资源.若在边缘网关转发请求的过程中收到了某个节点发现资源的消息, 则结束请求转发的过程, 从该节点获取最新资源并返回.

3 实验与结果分析

为了更详细的展示本文讨论的协议的应用过程, 通过一个模拟实验来具体说明.本实验利用海鸥的 GPS 轨迹的公开数据集, 模拟物联网节点将海鸥作为检测目标, 利用 Sealink 协议获取海鸥最新位置的过程.

3.1 数据预处理

本文提出的 Sealink 协议针对性的解决访问物联网中动态资源的问题, 为了模拟目标的位置在物联网节点间不断移动的情况, 本实验在 movebank^[13] 公开数据集中选择了一个记录海鸥迁移 GPS 数据的数据集.数据集按照时间戳、经度、纬度、海鸥 ID 等其他信息的方式记录海鸥的迁移情况.本数据集共有 89 868 条数据, 记录了 126 只海鸥的迁移情况, 时间跨越 2009 年~2013 年.在 126 只海鸥中有 24 只海鸥的数据量超过 1 000 条, 本实验仅讨论这 24 只海鸥的 65 435 条数据.

首先进行数据预处理.一个物联网节点能够监测一定范围的空间, 而不是一个具体的经纬度坐标, 因此需要将经度、纬度的具体值映射为物联网节点的位置.本实验按照墨卡托投影(Mercator projection)^[14] 将经纬度的值映射到物联网节点.

墨卡托投影是荷兰地理学家墨卡托于 1569 年提出的一种地球投影方法.对于经过墨卡托投影为平面的世界地图, 通过切割的方式将世界地图划分为若干个地图单元, 划分成的每一块地图单元称为地图瓦片.地图瓦片具有唯一的瓦片等级(level)和瓦片坐标编号(tileX, tileY).瓦片等级越高, 组成世界地图的瓦片数越多.某一瓦片等级地图的瓦片是由低一级的各瓦片切割成的 4 个瓦片组成.

根据谷歌地图瓦片坐标系的定义^[15], 将墨卡托投影地图的左上角作为瓦片坐标系起点, 往左方向为 X 轴, X 轴与北纬 85.05° 重合且方向向左; 往下方向为 Y 轴, Y 轴与东经 180° (亦为西经 180°) 重合且方向向下.在某一瓦片层级 level 下, 瓦片坐标的 X 轴和 Y 轴各有 2^{level} 个瓦片编号, 因此瓦片地图上的瓦片总数为 $2^{level} \times 2^{level}$ 个.

由经纬度坐标(lng, lat)转换为瓦片坐标(tileX, tileY)的公式分别为(3), (4):

$$\text{tileX} = \left\lfloor \frac{\text{lng} + 180}{360} \times 2^{level} \right\rfloor \quad (3)$$

$$\text{tileY} = \left\lceil \left[\frac{1}{2} - \frac{\ln \left(\tan \left(\text{lat} \times \frac{\pi}{180} \right) + \sec \left(\text{lat} \times \frac{\pi}{180} \right) \right)}{2 \times \pi} \right] \times 2^{\text{level}} \right\rceil \quad (4)$$

在本实验中将地图瓦片的坐标作为节点的坐标,一个(tileX,tileY)坐标对应一个节点,即假设一个节点的检测范围覆盖一块地图瓦片.根据墨卡托投影和地图瓦片的概念可知,只要选择合适的 level 值,就能将一个地图瓦片的实际地理面积控制在一个物联网节点能监测的空间范围以内.例如,当 level 取 20 时,一块地图瓦片的实际地理面积约为 0.000464km².

接着,将数据集中的时间戳分离成月份和日期,并且将时间转换为从 0:0:0 到该时刻所经历的秒数,然后,按照上述公式选择合适的 level 值将数据集中的经度和纬度分别转换为瓦片坐标(tileX,tileY),也即将海鸥的 GPS 运动轨迹进行瓦片序列化.再然后,统计该数据集中不同的瓦片个数 K,有多少个瓦片个数就表示有多少个物端节点并把每个物端节点一一编号.最后,设置步长为 L,从按照时间先后排序的序号为 1~N 的数据集中依次取长度为 L 的连续序列形成训练数据集.

3.2 神经网络模型及训练结果

利用神经网络算法处理 GPS 数据已经有一些研究成果,比如利用 BP 神经网络^[16]、马尔可夫链^[17]等模型进行轨迹预测,经过实验和比较,本实验选择 LSTM 模型进行实验.

LSTM 是一种时间递归神经网络(RNN)由于独特的设计结构,LSTM 适合于处理和预测时间序列中间隔和延迟非常长的重要事件.因此 LSTM 适合预测动态目标的节点位置.

本实验采用的网络模型结构如图 5 所示,其中:LSTM 单元接收顺序数据作为输入,包括海鸥所在的节点坐标等随时间变化的元数据.而对于不随时间变化的,如海鸥的编号 BID 等信息的则是利用嵌入层(embedding 层)处理.然后利用一个融合层(merge 层)连接嵌入层和 LSTM 层的输出,以便将它们被馈送到一个一维的标准全连接层(dense 层).为了对概率进行归一化,在全连接层后加入 softmax 函数层.最后得出下一时刻所有 K 个物端设备上存在动态资源的概率,取概率最大的物端设备作为预测结果.

在该网络模型中处理 GPS 数据的 LSTM 单元可以有若干层,层数越多网络模型越复杂,能够学习到的信息也越多.本实验验证了 LSTM 的层数对预测准确度的影响,结果如图 6 所示.

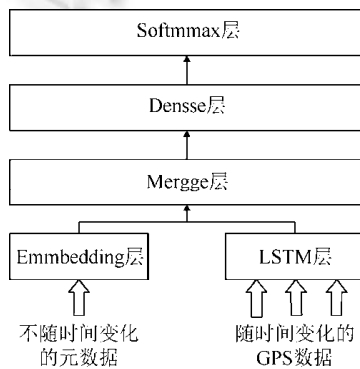


Fig.5 Network model structure

图 5 网络模型结构

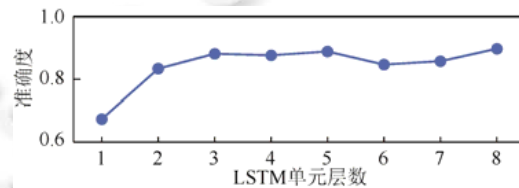


Fig.6 Influence of the number of layers of LSTM unit on prediction accuracy

图 6 LSTM 单元层数对预测准确度影响

从图中可以看出当 LSTM 的层数超过 3 层时,预测准确度稳定在 85%~90%之间.考虑到模型复杂度越高带来的计算开销越大,本实验在网络模型中,使用 3 层 LSTM 单元进行接下来的实验.

然后,本实验验证了该模型在数据集上的稳定性.将数据量大于 1 000 条的 24 只海鸥的 65 435 条数据按照

海鸥的 ID 分为 4 个集合,分别进行训练和验证.将每个集合的数据按照 80%和 20%的比例分为训练数据和测试数据,将训练模型放入网路中训练后利用训练数据检验模型的预测准确度.使用交叉验证的方式,经过多次训练和预测,得出预测准确度如图 7 所示.从图 7 中可以看出网络模型在该数据集预测准确度较为稳定,对集合 2 的预测准确度小于其他集合的原因是该数据集中每只海鸥的数据数都在 1 000~1 200 之间,数据量较少于其他集合.

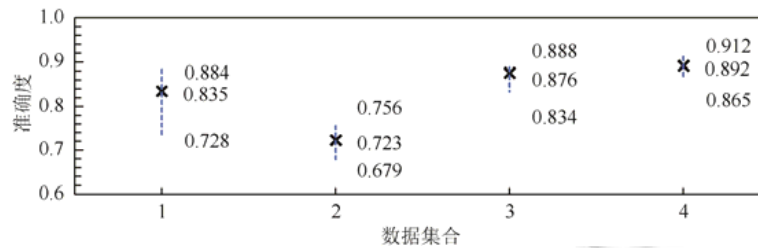


Fig.7 Prediction accuracy of LSTM models on different data sets

图 7 LSTM 模型在不同数据集合上的预测准确度

除 LSTM 模型之外,本文还利用 BP 神经网络进行了对比实验.在上述海鸥数据集中按照相同的方法将其分为 4 个部分,得到的预测准确度如图 8 所示.

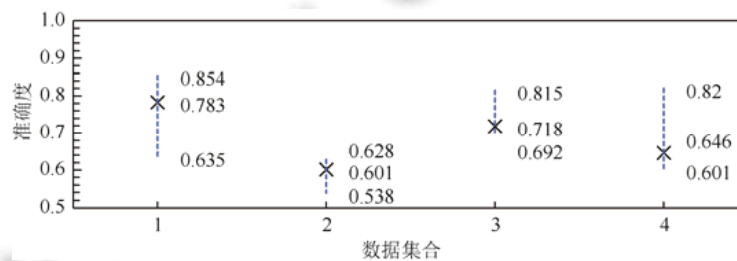


Fig.8 Prediction accuracy of BP models on different data sets

图 8 BP 模型在不同数据集合上的预测准确度

通过图 8 可以看出通过 BP 神经网络的预测准确度不稳定,并且,与图 7 对比也可以看出 BP 神经网络的预测准确度总体上低于 LSTM 模型的预测准确度,由此可知 LSTM 适合预测动态目标的节点位置.因此,本文的实验都是基于 LSTM 模型进行.

3.3 网络拓扑结构

在数据集中选择 3 只海鸥作为观察目标经过统计,假设在海鸥的路径中每个地图瓦片上放置一个物联网设备,取公式(3)、公式(4)中 $level$ 为 22 时,需要不同的物端设备个数为 2 669 个.此时每个物端设备覆盖的实际地理面积为 28.995 平方米,符合物端设备的实际监测面积.将这 2 669 个节点按照物联网应用中最常见的多跳网络的方式组织起来,规定每个节点最多拥有 n 个邻居节点.若以网关为根,则节点的网络拓扑结构类似 n 叉树.在本实验中假设 n 取 5 且使用一个边缘网关设备,则这 2 669 个物端设备组成的五叉树的高度不超过 5,即从网关到节点的最大跳数为 5.

3.4 实验结果讨论

本实验验证了在 Sealink 协议的支持下,在物联网设备组成的多跳物联网结构中,访问动态资源的性能.Sealink 协议的性能由获取到动态资源之前的请求被网关转发的次数来体现.本实验还对比分析了使用 http 协议按照随机转发和就近转发两种方式访问动态资源,所谓随机转发是指边缘网关将请求随机转发至网络内任意节点,就近转发是指边缘网关优先将请求转发至距离其跳数较低的节点上.

本实验进行 200 次随机请求,在节点监测到海鸥时将该消息与预测结果一同发给边缘网关,边缘网关利用算式(1)更新概率矩阵,在收到监测消息时,其更新系数 α 取 0.1;收到预测位置时,其更新系数 α 取 0.8.得到的实验结果如图 9 所示.图中展示了分别使用 3 种请求方法,在 1 次转发、3 次内转发和 5 次内转发内就能获得最新资源的请求次数.由图中可以看出使用 Sealink 协议时有超过 80%的请求能够在一次转发后就获得结果,超过 90%的请求能够在不超过 5 次转发后获得结果.因为海鸥在节点间的移动与节点到网关的距离无关,所以使用 HTTP 的两种访问方式得到的结果大致相同,只有大约 10%的请求能在被网关 5 次转发内获得结果.综上,Sealink 协议能够在更少的转发次数也即更短的时间内获得响应.

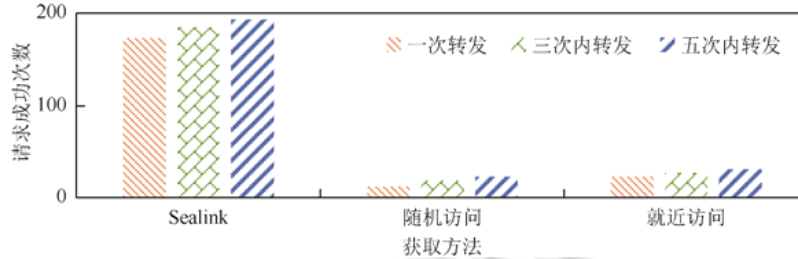


Fig.9 Comparison of the number of forwarding between Sealink and HTTP

图 9 使用 Sealink 协议与 HTTP 协议转发次数的比较

本实验还验证了预测准确度对协议性能的影响.根据调整迭代次数、LSTM 模型层数等参数将预测准确度控制在 60%、70%、80%和 90%左右(误差不超过 $\pm 5\%$),统计 200 次请求中在 5 次转发内就能获得正确结果的请求次数,如图 10 所示.由图可以看出 Sealink 协议对预测准确度的依赖较高,当预测的准确度较高时协议能够获得较高的性能,而当预测准确度较低时协议的性能也随之下降,例如当预测准确度在 50%左右时 200 次请求中只有不超过 55%的请求能在 5 次转发内获得资源.

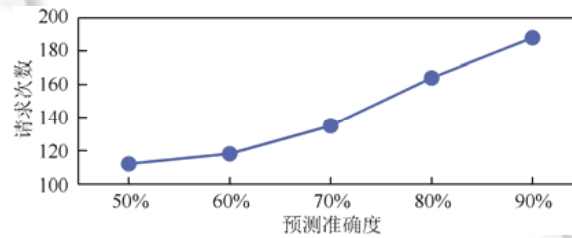


Fig.10 The impact of prediction accuracy on the average number of requests

图 10 预测准确度对平均请求次数的影响

本实验测试了预测准确度约为 90%时,使用 Sealink 与 HTTP 两种协议获取动态资源的平均延时与网络中节点个数的关系.首先,本实验利用 raspberry pi zero w^[18]作为物端设备,网关与物端设备通过 Wi-Fi 通信,测试了网关到设备一跳网络的访问延时情况.假设访问延时的概率分布符合正态分布,经过 100 次的测量和统计,拟合出的正态分布的概率密度函数式(5)中 $\mu=377.04$; $\sigma=203.114$.

$$f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (5)$$

以此实验数据为基础,且假设多跳网络的访问延时与网关到设备间的跳数呈线性正相关关系进行接下来的实验.在地图瓦片的坐标转换公式(3)、公式(4)中,level 取值越小覆盖海鸥路径所需要的不同的节点总数越少.经过统计,在实验所用的数据集中,当 level 分别取 4、5、6、7、8 时,所需节点总数分别为 6、19、41、82、132.节点按照第 4.3 节阐述的网络模型组成多跳网络,并且访问网络中节点的延时根据式(5)的概率分布取值.实验统计了 200 次随机请求中每次请求的平均延时,如图 11 所示.

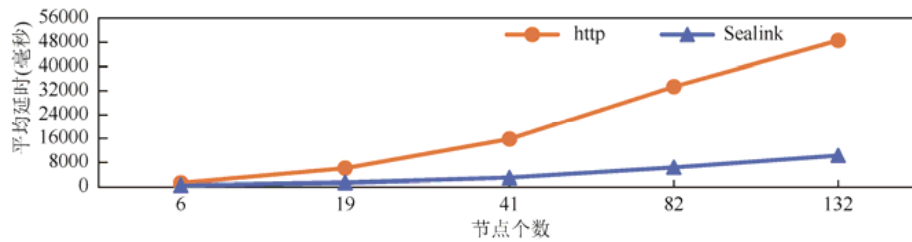


Fig.11 The relationship between the average delay of accessing dynamic resources and the number of IOT nodes

图 11 访问动态资源的平均延时与节点个数的关系

动态资源的位置变化会使其访问地址失效,当使用 HTTP 协议时,网关只能在网络节点中轮询,直到获得动态资源为止.节点个数越多导致轮询节点带来的时间开销越大,因此 HTTP 协议的平均延时受节点个数的影响较大.然而,Sealink 协议在访问动态资源的过程中加入预测机制,正确的预测结果能够让网关在一次转发后就能获得资源而不受节点个数的影响.由图中可以看出,网络中节点个数越多,Sealink 协议能够降低的平均延时越大.经过统计,与 HTTP 协议相比,Sealink 平均降低了 75.5%的平均延时.

4 总结与展望

为了提高物联网设备的互操作性,针对物联网应用中动态资源访问位置不断变化带来的问题和挑战,本文提出的面向动态资源的物联网应用层协议 Sealink 采用 RESTful 设计风格,使用 URI 描述动态资源,简化获取动态资源的过程,并且将预测机制加入获取动态资源的过程中.在公开数据集的实验表明,Sealink 协议可以减少获取动态资源时转发请求的次数,降低延时,提高请求响应率.

在未来的工作中将使用更多的数据集进行实验和研究,从而优化对动态资源移动轨迹的统一建模方法等其他影响协议性能的因素,进而提高 Sealink 协议的性能和易用性.

References:

- [1] Lee I, Lee K. The internet of things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, 2015, 58(4):431–440.
- [2] MQTT.org. Mq telemetry transport. 2018. <http://mqtt.org/>
- [3] Chen HM, Cui L. Design and model checking of service oriented software architecture for Internet of things: A survey. *Chinese Journal of Computers*, 2016,39(5):853–871. (in Chinese with English abstract).
- [4] Guinard D, Trifa V, Mattern F, *et al.* From the internet of things to the Web of things: Resource-oriented architecture and best practices. *Architecting the Internet of things*. Berlin, Heidelberg: Springer-Verlag, 2011. 97–129.
- [5] <http://www.thingworx.com/iot-platform>
- [6] <https://www.ptc.com/cn/development/axeda>
- [7] <https://www.sierrawireless.com/products-and-solutions/sims-connectivity-and-cloud-services/iot-cloud-platform/>
- [8] <https://www.realtime.io/>
- [9] Shelby Z, Hartke K, Bormann C. The constrained application protocol (CoAP). 2014.
- [10] Hou CD, Li D, Qiu JF, *et al.* SeaHttp: A resource-oriented protocol to extend REST style for Web of things. *Journal of Computer Science and Technology*, 2014,29(2):205–215.
- [11] Masinter L, Berners-Lee T, Fielding RT. Uniform resource identifier (URI): Generic syntax. 2005. <https://tools.ietf.org/html/rfc3986>
- [12] Gers FA, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 2000,12(10): 2451–2471.
- [13] <https://www.movebank.org/>
- [14] Sun D, Pu YX. *Map Projections*. 2nd ed., NanJing: Nanjing University Press, 2012. (in Chinese)

- [15] Kou MM, Wang QZ, Tan TD. Research on Google map algorithm and application. Computer Technology and Development, 2012, (4):204–206. (in Chinese with English abstract)
- [16] Liu YY, Xu XK, Wei JM, *et al.* Application of BP neural network to GPS differentiation positioning. Journal of Northeastern University Natural Science, 2008,29(11):1536–1539. (in Chinese with English abstract)
- [17] Peng Q, Ding ZM, Guo LM. Prediction of trajectory based on Markov chains. Computer Science, 2010,37(8):189–193. (in Chinese with English abstract)
- [18] <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>

附中文参考文献:

- [3] 陈海明,崔莉.面向服务的物联网软件体系结构设计及模型检测.计算机学报,2016,39(5):853–871.
- [14] 孙达,蒲英霞.地图投影.第2版,南京:南京大学出版社,2012.
- [15] 寇曼曼,王勤忠,谭同德.Google Map 数字栅格地图算法及应用.计算机技术与发展,2012,(4):204–206.
- [16] 刘英英,徐香坤,魏静敏,等.BP神经网络在差分GPS定位技术中的应用.东北大学学报(自然科学版),2008,29(11):1536–1539.
- [17] 彭曲,丁治明,郭黎敏.基于马尔可夫链的轨迹预测.计算机科学,2010,37(8):189–193.



廖伟源(1993—),女,河南信阳人,硕士,主要研究领域为物端计算,边缘计算.



崔莉(1962—),女,博士,研究员,博士生导师,CCF高级会员,主要研究领域为物联网,传感器技术.



李栋(1979—),男,博士,副研究员,CCF专业会员,主要研究领域为物端计算,边缘计算.