

节点位置信息驱动的物联网终端缓存管理机制的研究

陶 军 施书静 冯富琴 高 扬

(东南大学教育部计算机网络和信息集成重点实验室 南京 210096)

(江苏省计算机网络重点实验室 南京 210096)

(东南大学网络空间安全学院 南京 210096)

摘 要 物联网通过物体、传感设备与网络的关联来实现物体自动、实时的识别、定位、追踪和监控,并触发相关事件.面对持续采集或到达的数据流,物联网的移动终端间需要进行大量的数据交互.在节点缓存空间有限的条件下,合理的节点缓存管理机制可以在提高消息投递成功率的同时减少网络开销、降低终端能耗.然而,现有缓存管理机制都是从消息队列特征或静态属性方面展开研究,忽视了节点的位置、移动方向等特征,从而导致消息在网络中无方向性的扩散;或虽考虑了节点移动特征,但集中于时间维度,且对节点间相遇规律做了诸多假设,如节点间相遇服从独立同分布等,这些都限制了算法的实际应用场景.考虑到物联网中节点移动的空间规律性,该文在研究节点移动模型的基础上提出了节点位置信息驱动的终端缓存管理机制.在消息目的位置已知的情况下,该机制根据节点在随机移动模型和车载移动模型两种场景下位置和方向的移动特征,结合节点对网络态势的感知,分别计算消息的优先级;依据该优先级,设计节点间进行数据交换时的消息调度策略及节点缓存空间不足时的消息丢弃策略;并通过对移动方式随机(如人和动物的随机移动)和移动方式受限(如城市道路中车辆移动)两种场景的仿真结果分析,评估算法的性能表现.仿真实验表明,相比于 Drop Oldest(DO)等传统缓存管理策略,该文提出的缓存管理机制在不同缓存大小和不同程度网络拥塞的情况下体现出较高的性能优势,并在提升消息投递率的同时保持了较低的平均时延、网络负载率和平均跳数.

关键词 物联网;缓存管理;随机移动模型;车载移动模型

中图法分类号 TP393 **DOI号** 10.11897/SP.J.1016.2019.00252

Research on Location-Driven Buffer Management of End Nodes in Internet of Things

TAO Jun SHI Shu-Jing FENG Fu-Qin GAO Yang

(The Key Laboratory of Computer Networks and Information Integration, Ministry of Education, Southeast University, Nanjing 210096)

(Key Laboratory of Computer Network Technology of Jiangsu Province, Nanjing 210096)

(School of Cyber Science and Engineering, Southeast University, Nanjing 210096)

Abstract Internet of Things (IOT), which exploits the association among the objects, the sensing devices and the Internet, achieves the automatic and real-time management, i. e., recognition, localization, tracking, monitoring, and triggers the related events. Facing the continuous data flow collected from the monitored environment or transferred from other nodes, the mobile nodes of IOT have to conduct a large amount of data interactions. With the limited buffer size of nodes, a rational buffer management mechanism is expected to improve the delivery ratio of messages, reduce the network overhead and maintain a low level of energy cost in the nodes. However, the existing research efforts on the buffer management mechanisms, which mainly focus on the aspect of message queue features or the static attributes of messages, neglect some characteristics, e. g., the location and moving direction of nodes. Therefore, the untargeted spread of messages, which

收稿日期:2017-11-01;在线出版日期:2018-06-29. 本课题得到国家自然科学基金(61370209)、江苏省自然基金(BK20151416)资助.
陶 军,男,1975年生,博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为物联网技术、无线自组织网络(无线传感网、移动社交网络、车载网络). E-mail: juntao@seu.edu.cn. 施书静,女,1994年生,硕士研究生,主要研究方向为物联网技术. 冯富琴,女,1992年生,硕士,主要研究方向为 Ad Hoc 网络. 高 扬,男,1989年生,博士研究生,主要研究方向为无线网络.

results in the heavy network overhead, occurs in the network. Although some research efforts have considered the mobility pattern of the node, they mainly focus on the time dimension, e. g. , the time interval between any two contact events among nodes. Moreover, many assumptions of node contacts, e. g. , the encounter time interval among nodes follows the independent and identical distribution, result in a limited practical application scenario of the buffer management algorithms. Considering the spatial regularity of node mobility, this paper proposes a location-driven buffer management mechanism based on the analysis of different mobility models. With the knowledge of the message's destination location, the mobility characteristics of the node location-direction in both random mobility model and the vehicular mobility model are applied to calculate the priority of messages. Based on the message priority, the buffer scheduling policies, when two contacted nodes need to exchange messages with each other, and the necessary dropping policies when the buffer depletion occurs, are designed. The performance of the proposed mechanism is evaluated by organizing the simulation under both random movement scenarios, e. g. , human hold mobile devices or animal carried sensors, and the limited movement scenarios, e. g. , the taxis, buses and private cars in the urban roads. The results of simulation reveal that compared with other traditional buffer management policies, e. g. , Drop-Oldest (DO), the proposed mechanism shows a promoted performance. Under different buffer sizes and different degrees of network congestion, the proposed buffer management mechanism can improve the message delivery ratio, and reduce the average delivery latency, the network overload ratio and the average hops at the same time.

Keywords Internet of Things; buffer management; random mobility model; vehicular mobility model

1 引 言

物联网(Internet of Things, IOT)是物品、传感设备、移动设备及互联网间在任何时间、任意地点实施互联互通的网络,它将互联网的优势发挥到日常生活的各个方面^[1]. 物联网凭其全面感知、可靠传送和智能处理等特征在学术界、工业界受到了广泛关注. 物联网在构建智能城市、智能家庭、智能环境监测、智能制造和物流等方面得到了广泛的应用^[2]. 物联网的核心功能包括信息获取、信息传输和信息处理. 具备感知能力的终端节点通过感知物品的标签获取物理世界信息,并通过接入网关接入互联网. 这些终端往往具备移动性,如移动 RFID 读卡器、传感器等,并可以通过具备网络通信功能的有线、蓝牙、WIFI、Zigbee 等多种接口,实现节点间的数据通信. 它们采集到的信息通常暂存于终端节点缓存中,直到传给下一个终端节点或接入互联网中心处理平台中.

物联网中的移动终端通常具备标签读取与识别、移动传感和数据收集等功能,在丰富的物联网应用中扮演着不可或缺的角色. 如物流领域中,装备了

RFID 读取和条码扫描设备的手持 PDA 可以对包裹的物流信息进行智能采集,协助完成物流配送和仓储管理;在智能交通领域,具备 GPS 定位、汽车参数传感监测以及环境监测模块的车载移动终端不仅可以提供道路导航服务,还能实现交通路况、周围环境的实时监测;在地质勘探、森林防护以及野生动物监测等野外作业领域,根据实际需求部署具备定位、数据采集和即时通信的移动终端,周期性地数据进行数据汇聚,可降低人力采集的风险和成本,实现经济可靠的野外作业方式;除此之外,移动终端在智能医疗、军事作战等领域中也常承担着数据采集、汇聚和简单处理的角色.

由于移动终端节点的动态产生、移动性以及电量限制等条件,节点间无法建立长期稳定的端到端传输路径,尤其是在节点稀疏的环境中. 于是“存储-携带-转发”的报文传输方式在移动自组织网络中被广泛使用. 在这种模式下,节点接收报文后将其暂存于缓存队列中,直至遇到下一个转发机会. 基于“存储-携带-转发”模式,研究人员提出了大量时延容忍网络路由策略^[3-5]. 为了提高投递率,路由常以多副本方式进行,即允许网络中某时刻存在同一消

息的多个副本.然而,物联网中移动终端节点缓存空间、传输带宽及电量资源等有限,在这种情况下,路由策略的性能受到了极大影响.如文献[6]指出,虽然 Epidemic 路由策略在没有缓存和带宽限制的情况下可以达到最小传输时延和最大投递率,但是在缓存大小有限的情况下却表现很差.一方面,当两节点相遇时,由于接触时间和传输带宽的限制,并不能保证消息的全部转发;另一方面,当报文数量较多时容易造成缓存溢出,此时必须丢弃部分报文.对于采集应用级数据为主要任务的物联网移动终端来说,数据报文种类各异、大小不一且常以密集的频度出现.面对持续到达的数据流,需要合理的缓存管理机制来降低节点能耗、避免网络拥塞并提高传输效率.

节点的缓存管理机制涵盖两方面的内容:消息的调度策略和消息的丢弃策略.消息的调度策略是指在节点出现转发机会时,如何决定缓存中消息的传输次序,实现消息的调度.消息的丢弃策略是指当节点的缓存空间已满时,面对新到达的消息,需要节点判断接收新消息还是丢弃缓存中原有消息,并决定缓存中消息的丢弃次序,以降低网络的整体投递率损失.

目前已有的物联网缓存队列管理大多关注于消息的队列特征或消息类型、使用频度和内容关注度等局部信息^[7-8],全局网络信息的缺失导致基于局部信息的调度和丢弃策略无法做出全局性能优化的决策.如文献[6]指出,传统的 Drop-Tail 丢弃策略在延迟容忍网络中表现很差.考虑到车载终端、手机等移动设备多配备 GPS 定位功能,可获知当前地理位置,且大多数物联网应用场景的移动终端并非毫无规律,即使是随机移动模型也显示出一定的伪随机性^[9-10].如文献[9]指出,一维和二维平面中的随机游走模型中,节点终将移动至原点,这保证了模型中节点将绕其初始位置运动而不会随意漫游至一去不返;文中还指出,随机路点模型中,节点经过一段时间后将保持一个稳定的平均邻居节点率,即节点通信范围内的邻居节点数趋于稳定,且该模型中选择区域中心为路点或经过区域中心的概率较高.文献[10]则根据真实公交车 Trace 数据 UMass 的分析结果指出,公交车的移动在路由层面上具有周期性规律.在装备了电子导航系统的车辆上,通过获取司机的预定行车轨迹,可以预测车辆目的地和移动规划,掌握车辆移动规律.本文基于不同场景的移动模型,根据终端节点在场景中的移动位置特征,在节点可获知所携带消息的目标位置及自身位置的情况

下,为网络移动节点的缓存设计合适的消息调度和丢弃策略,并考察该缓存管理机制在消息的投递成功率、平均投递时延、网络开销以及平均跳数方面的性能.

2 相关工作

随着物联网体系结构的日益完善,针对物联网的研究层出不穷.物联网的系统架构、协议体系、系统应用、RFID 以及传感器网络相关的软硬件技术都引起了业界人士的关注.在移动节点的相关研究中,大量的路由策略被提出,一些适用于移动无线网络场景的缓存队列管理也作为路由策略的辅助手段被提出.

传统的无线 Ad hoc 网络缓存管理策略依赖于消息队列的特性,如头部丢弃、尾部丢弃策略^[11]等.它们的实现比较简单,在节点缓存空间大、网络带宽不受限的情况下,可以表现较好的网络性能.然而,物联网移动终端带宽和缓存空间有限,此类缓存管理策略在物联网中的应用性能并不具备优势.

利用消息的局部属性,一些启发式缓存管理策略被提出.如文献[12]利用消息的跳数来制定路由和缓存管理策略;文献[13]根据消息的传输速度来估计消息状态并做出调度和丢弃决策;文献[14]则根据消息大小、消息生存时间(Time to Live, TTL)以及消息在节点中停留的时间来制定网络中的拥塞控制策略.根据不同场景下的网络服务质量(Quality of Service)需求,缓存管理模型可以选择多种消息属性作为参数进行消息效用计算.然而,消息队列特性和局部属性都无法呈现出网络的全局状态,依靠独立的本地信息不足以做出对全局性能优化的消息调度和丢弃决策.因此,研究人员引入了部分全局信息来协助制定缓存管理策略.

网络的全局信息包括节点总数、网络中消息的副本数以及消息在网络中的分布状态等.而物联网中移动终端组成的网络是分布式的,节点无法同步获得网络的全局信息.为了获取网络中消息副本数,文献[15]使用额外的控制信道来收集网络状态信息,并据此针对最小化平均时延、最小化丢包率以及最小化最大时延三个不同最优化目标计算消息效用,提出了 RAPID 策略;而文献[16]则指出, RAPID 使用的效用方程并非是最优的,因其缺少消息成功投递通知而且没有考虑消息产生前网络中的事件.因此文献[16]不仅考虑了网络中消息当前副本数,还考

虑了被丢弃或成功投递的消息副本数,提出了基于全局信息的调度策略 GBSD. 由于全局消息获取在分布式环境难以实现,作者进一步阐述了使用统计学习估计全局状态的方法,提出了基于历史信息的调度策略 HBSD. 然而,GBSD 假设消息的 TTL 是无限的,即足够保证所有的消息可以在 TTL 过期之前传递给目标节点,这并不符合实际场景. 文献 [17] 提出的 EBMP 对此进行了改进,考虑了消息的剩余生存时间、消息当前寿命以及估计的消息当前副本数,综合做出决策.

基于以上分析,可以发现目前的研究工作多采用消息队列特征或静态属性展开研究,部分研究^[16]考虑到了节点的移动模型,但其利用的是节点间相遇时间信息,且对节点移动模型限定了诸多假设,如节点间相遇间隔服从独立同分布的指数分布等. 它们并没有考虑节点的空间移动特征,如节点的位置、移动方向等. 然而物联网与真实的物理世界紧密相连,移动节点和消息的位置变化都表现出显著的社会特征. 例如在对城市的交通状况进行监测时,车辆的移动轨迹是限定在城市道路中的;野生动物监测中,安装在野生动物身上的终端节点的移动范围也通常有范围限制;在周期性进行的森林防护,电力巡检等场景中,信息采集终端常以规划好的路线移动. 通过对移动模型的分析,可以预估物联网终端的移动轨迹,从而掌握物联网消息(数据)在网络中的分布和传播特点,为多消息副本缓存管理优化创造更多契机.

本文通过对随机移动模型和车载移动模型中节点移动特征的分析,模拟现实生活中物联网节点的移动场景. 其中随机移动模型可以模拟人类和其它动物的移动场景,车载移动模型可以模拟车辆在道路中的移动场景,它们代表了物联网中的两类典型应用场景,即移动轨迹随机和移动线路受限. 通过分析上述两类场景中节点位置和移动方向对缓存管理的影响,可为手持设备信息采集、借助野生动物移动行为的环境监测以及车联网数据分发等数据传输应用设计缓存管理策略. 上述应用既要保证数据投递率,也需维护数据的时效性. 然而野外环境、空气质量及车流密度等数据短时间内可维持稳定,由于终端节点的移动性和节点相遇的不确定性,此类应用可允许不同时间长度的消息时延.

3 节点位置信息驱动的缓存管理

物联网缓存管理包含两方面的内容:(1) 当消

息到达时,因为两个节点的相遇时间、带宽受限等因素,对方缓存中的消息往往不能全部复制过来,所以需要设置消息的优先级,即设计合适的调度策略;(2) 当节点的缓存空间不足或溢出时,节点需判断是否丢弃本地消息来接收新的消息,若要丢弃则需要决定优先丢弃哪些消息,即设计合理的丢弃策略. 本文在多副本消息路由情况下,基于现有移动模型,根据节点的移动位置信息和辅助的全局知识来制定合理的缓存管理机制.

3.1 全网状态信息获取

为了更加清晰地阐述本文思路,首先将所用到的符号定义如下:

$N = \{n_i\}$: 网络中所有移动节点的集合;

r : 节点的传输半径;

IDn_i : 移动节点 n_i 的 ID;

IDm_j : 消息 m_j 的 ID;

TTL_{m_j} : 消息 m_j 的初始生存时间;

T_{m_j} : 消息 m_j 自产生开始所经过的时间;

S_{m_j} : 消息 m_j 的大小;

R_{m_j} : 消息 m_j 的剩余生存时间, $R_{m_j} = TTL_{m_j} - T_{m_j}$.

定义 1. 节点相遇. $\forall n_i \in N$ 和 $\forall n_j \in N$, 且 n_i 和 n_j 不同, 如果物联网中节点 n_i 和 n_j 之间的欧几里德距离 $|n_i, n_j| \leq r$, 那么节点 n_i 和 n_j 相遇.

为了更好地研究消息的调度策略和丢弃策略,节点需要获取所携带消息的全局网络状态信息为 $m_j(T_{m_j})$, 即消息 m_j 从产生起经过时间 T_{m_j} 后在网络中的副本数. 同时节点可以通过定位服务获取以下本地信息:(1) (x_{n_i}, y_{n_i}) : 节点 n_i 当前位置的坐标;(2) (x_{m_j}, y_{m_j}) : 消息 m_j 的目的位置坐标;(3) θ_{n_i} : 节点 n_i 的移动方向与方向坐标 x 轴的夹角, 夹角范围为 $0 \sim 2\pi$.

节点 n_i 的元数据信息由节点信息表 $Table_{n_i}$ 表示, 包括 IDn_i 、 M_{n_i} 、 L_{n_i} 、 (x_{n_i}, y_{n_i}) 以及 θ_{n_i} . 其中 M_{n_i} 表示终端 n_i 中携带的所有消息及其基本信息列表, 每个列表项 $List_{m_j}$ 描述了消息 m_j 的当前状态, 如图 1 所示, 节点 n_i 携带的消息列表 M_{n_i} 包含了 $List_{m_1}$ 、 $List_{m_2}$ 和 $List_{m_3}$ 等; L_{n_i} 表示节点 n_i 中 M_{n_i} 的最后 (Last) 更新时间.

消息 m_j 的基本信息由 $List_{m_j}$ 表示, 包括 IDm_j 、 $Status_{m_j}$ 、 T_{m_j} 、 TTL_{m_j} 、 S_{m_j} 以及 (x_{m_j}, y_{m_j}) , 其中 $Status_{m_j}$ 表示消息 m_j 当前的状态, 它的可能取值有两个为: DELIVERED 和 BUFFERED, DELIVERED 表示消息 m_j 已经投递到目的地, 因此要从终端的缓

存中删除; *BUFFEREED* 代表消息 m_j 仍然在节点的缓存中, 投递状态未知。

当网络中节点数少, 网络状态稀疏时, 节点可以通过图 1 所示数据交换方式获取消息的本地和全局信息. 每个终端既存储本节点的基本信息, 同时也存储自己所携带消息的基本信息. 当两个节点相遇时, 它们将记录对方节点的 ID 和消息的基本信息. 同时, 交换合并属于对方的来自其它节点的元数据表和消息基本信息. 节点存储消息的基本信息最新的“最后更新时间”, 只有在上次交换后发生了数据更新的情况下才会进行数据交换. 通过这个过程, 节点将会获得统一且准确的全局网络状态, 全局参数 $m_j(T_{m_j})$ 可以通过统计节点的消息基本信息中的 *Status* 状态获得. 当网络规模较小时, 系统经过一定时间的初始化运行后, 节点间数据交换次数减少, 系统趋于稳定。

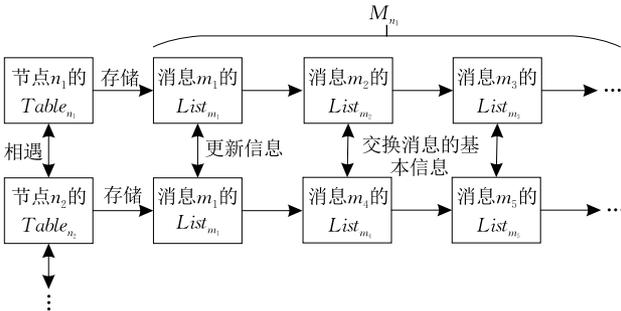


图 1 节点交换信息过程

当网络规模较大时, 网络中的消息副本数则可以使用参考文献[15-18]中提出的方法来进行估测, 如基于统计规律的概率计算方式^[18]、使用控制信道进行消息敛播的方式^[15]、基于历史信息的统计学习方式^[16]以及基于本地转发次数和消息跳数的估测方式^[17]等. 通过这些手段可以通过较低的计算开销近似地获得消息当前在网络中的副本数。

3.2 随机移动模型驱动的缓存管理机制

移动模型用于描述移动用户的移动方式以及用

户所在的位置、速度和加速度等, 这些都将随着时间的变化而变化. 在“存储-携带-转发”的数据传输模式下, 节点间的信息交互仅发生在节点相遇时, 节点的相遇事件与节点的移动模型息息相关, 节点移动方式对确定的协议性能有着至关重要的作用. 因此研究物联网中的缓存管理机制时, 基于恰当的移动模型, 显得尤为必要^[19]. 而目前对于节点移动模型的研究多集中于时间信息, 即节点间相遇事件间隔、节点停留时间等, 节点的空间信息, 如节点位置、方向则受到较少的关注. 在获知消息的目的位置的情况下, 利用节点的当前位置和移动方向进行缓存管理, 可以优先处理目的位置与节点移动轨迹相近的消息, 避免消息在网络中无方向性地四处扩散。

本节首先讨论物联网中移动终端移动方式不受限的场景, 即移动终端可以按任意方向和速度进行移动, 并利用随机移动模型刻画这种移动行为。

3.2.1 随机移动模型

目前广泛使用的随机移动模型有随机路点模型 (Random WayPoint model, RWP)^[20]、随机游走模型 (Random Walk model, RW) 和随机方向模型 (Random Direction model, RD)^[21], 它们的运动特征如图 2 所示。

图 2(a) 表示的为 RWP 模型. RWP 模型中节点的移动规则为: (1) 节点随机选择网络中的一个路点 (waypoint); (2) 节点选择在 $[minspeed, maxspeed]$ 上服从均匀分布的速率朝 waypoint 移动; (3) 当节点到达一个 waypoint 后, 选择在 $[0, T_{max}]$ 上服从均匀分布的暂停时间进行停留; (4) 暂停时间结束后, 继续选择下一个 waypoint 重复上述步骤。

图 2(b) 表示的为 RW 模型. RW 模型中节点的移动规则为: (1) 节点随机选择在 $[0, 2\pi]$ 上服从均匀分布的方向; (2) 节点随机选择在 $[minspeed, maxspeed]$ 上服从均匀分布的速率; (3) 节点按选择

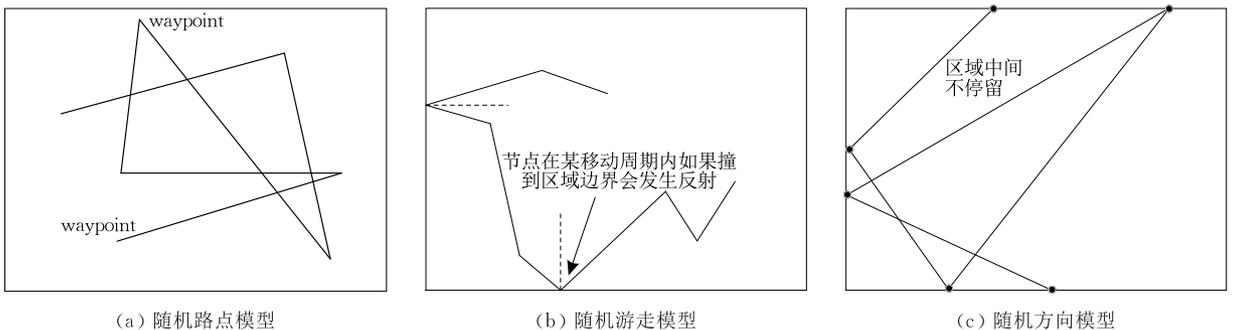


图 2 典型物联网终端的随机移动模型

的方向和速率移动一定距离或一定时间后停止此轮运动; (4) 节点继续选择下一个方向与速率重复上述步骤. RW 也可以在有边界限制的场景下工作, 节点在边界上根据反射定理反射回限制区域内部.

图 2(c) 表示的为 RD 模型. RD 模型与 RW 模型的移动规则相似, 区别在于 RD 模型只能工作在限定的区域内, 节点每次按随机选择的速率和方向移动到区域边界, 随机停留一段时间后再进行下一阶段运动, 节点的移动轨迹是由区域的弦构成.

这三种随机移动模型有着类似的移动特性, 如 RD 模型可以看作 RW 模型增加了暂停时间和边界限制后的特例, 也可以看作是 RWP 模型把目的地限制在区域边界时的特例; 而 RWP 模型将暂停时间设置为零后也可以模拟简单的 RW 模型. 文献[9]中指出, RWP 模型已被多个 ad hoc 网络协议仿真研究使用, 可以模拟真实场景中人类的移动模式. 而 RW 模型更适合用于半静止状态的网络仿真; RD 模型则不适用于真实场景仿真, 因为移动设备一般不会严格地到达区域边界后(如城市边缘)再改变移动路径. 基于以上考虑, 我们选择 RWP 模型作为代表对运动方向和速率不受限的随机移动场景下节点移动位置特征进行分析. 其它随机移动模型可以通过对 RWP 模型增设限制条件进行模拟.

3.2.2 基于方向信息的消息权重

假设网络中有两种类型的节点: 固定节点和移动节点. 固定节点为产生用于转发的源消息的节点, 它在一段时间内停留在某一位置. 移动节点则是在移动模型下移动, 能够从固定节点和移动节点之间“存储-携带-转发”消息的中继节点.

假设节点在 RWP 模型中移动, 为得到随机移动模型中节点的移动位置信息, 将 RWP 模型的研究区域置于二维坐标系的第一象限中, 节点的位置坐标用 (x, y) 表示, 节点的移动方向与 x 坐标轴的夹角用 θ 表示. 用 ω_{m_j} 表示消息 m_j 基于方向信息的消息权重值, ω_{m_j} 值越大代表移动节点越可能到达消息的目的区域. 图 3 为用于计算 ω_{m_j} 值的一个移动模型示例. 如图 3 所示, 移动节点 n_p 携带了四个消息, 分别是 m_1, m_2, m_3 和 m_4 . 已知四个消息的目的位置坐标 $(x_{m_1}, y_{m_1}), (x_{m_2}, y_{m_2}), (x_{m_3}, y_{m_3}), (x_{m_4}, y_{m_4})$, 分别用 D_1, D_2, D_3 和 D_4 表示, 节点 n_p 的位置坐标是 (x_{n_p}, y_{n_p}) , 节点的移动方向与 x 轴的夹角是 θ . 一方面, 当消息的目的位置距离当前节点的位置更近时, 消息更容易到达目的地; 另一方面, 当消息的目的位置与当前节点位置连线和当前节点的

移动方向构成的夹角更小时, 说明消息的直线传输方向与节点的移动方向更相近, 消息更趋近于被送往目的地. 如图 3 所示, n_p 携带的四个消息中, 消息 m_2 的目的位置与当前节点位置连线与节点移动方向的夹角最小, 被传输至目的地的可能性大. 考虑到这两个方面, 消息 m_j 的方向权重值 ω_{m_j} 由式(1)的方式计算, 令节点移动方向与节点与消息目的位置之间的夹角为 β (如图 3 所示), 则:

$$\omega_{m_j} = \frac{\cos \beta}{\sqrt{X^2 + Y^2}} \quad (1)$$

其中, $X = x_{m_j} - x_{n_p}, Y = y_{m_j} - y_{n_p}, \beta = \arctan \frac{|Y|}{|X|} - \theta$.

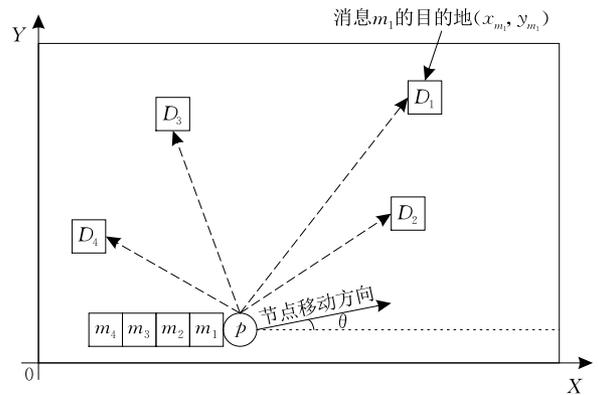


图 3 基于方向信息的消息权重模型

3.2.3 MOB 缓存管理机制

本节利用消息的方向权重, 设计随机移动模型驱动的终端缓存管理机制 (random Mobility model-driven Buffer management mechanism, MOB), 该机制包含丢弃策略和调度策略.

(1) MOB 的丢弃策略

为了能够更细粒度地解决节点的缓存管理问题, 除了考虑上面提出的基于方向信息的消息权重值, 也不能忽视消息属性的影响, 如消息的生存时间、消息大小以及消息副本数等. 时间因素的引入满足了场景中对数据时效性的要求; 消息大小因素则旨在减少报文丢弃操作, 降低场景中网络开销. 式(2)计算了消息的丢弃权重值, 它表示消息被传递至目的位置的可能性, 权重值越大, 可能性越高, 消息的丢弃优先级越低. 当节点的缓存空间满时, 将根据消息的丢弃权重自小到大依次丢弃.

$$d_{m_j} = \omega_{m_j} \cdot (\log(R_{m_j}) + m_j(T_{m_j})) \cdot S_{m_j}^{-1} \quad (2)$$

其中, d_{m_j} 为消息 m_j 丢弃 (dropping) 权重值, 时间单位为秒. 由于消息的 R_{m_j} 初始值远高于消息的 $m_j(T_{m_j})$ 值, 为了将它们数量级上进行归一化, 对消息的 R_{m_j} 值取自然对数, 利用对数函数使处于不

同量纲的值处于同一数量级。

式中 ω_{m_j} 为消息方向权重值,它代表了消息当前位置和移动方向与目的位置之间的关联程度; $\log(R_{m_j}) + m_j(T_{m_j})$ 代表了消息在网络中的扩散能力. 剩余生存时间少且副本数少的消息被优先丢弃, 因其在网络中生存时间长而被转发的次数少, 或已被多数节点丢弃, 即该消息在剩余时间被扩散到目的位置的概率较小; $S_{m_j}^{-1}$ 代表了单位大小消息所占的权重, 物联网中报文形式各异、大小不一, 优先丢弃较大的报文可以容纳更多的新报文数, 并减少丢弃报文的次数。

RWP 移动模型中, 节点在到达一个 waypoint 时会暂停一段时间, 接着选择下一路点. 节点中消息的丢弃权重更新和排序发生在两个时刻: ① 节点改变运动方向向下一路点移动时; ② 节点与其它节点相遇时. 当节点与其它节点进行数据交换时, 若缓存空间不足则会根据消息的丢弃权重排序结果, 比较最小丢弃权重值与对方节点发来消息的丢弃权重, 若本地权重较大则不接收, 否则从权重最小的消息开始丢弃. 该过程如算法 1 所描述.

算法 1. MOB 的丢弃算法.

输入: 相遇节点的一个消息 m

输出: 本地消息丢弃

1. 更新本地消息丢弃权重并排序
2. IF $m.size > Buffer.size$ THEN
3. reject 消息 m ;
4. END IF
5. //msg 是本地缓存中丢弃权重最小的消息
6. IF $d_{msg} > d_m$ THEN
7. reject 消息 m ;
8. END IF
9. WHILE $Buffer.freesize < m.size$ DO
10. IF $msg == m$ THEN
11. reject 消息 m ;
12. ELSE
13. delete 消息 msg ;
14. $Buffer.freesize = m.size$;
15. END IF
16. END WHILE

(2) MOB 的调度策略

消息的调度策略和丢弃策略并不相同, 当两个节点相遇的时候, 它们的移动方向未必是相同的, 故加入消息方向权重值的优先级并不具有参考价值. 本文在调度策略中不考虑消息 m_j 的方向权重值 ω_{m_j} , 当两个节点相遇时, 利用式(3)计算传输节点中

消息 m_j 的调度优先级 s_{m_j} 来调度消息, 换言之, 调度策略的权重更新时间发生在两个节点的相遇时刻:

$$s_{m_j} = \log(R_{m_j}) + m_j(T_{m_j}) \quad (3)$$

算法 2 为 MOB 的调度算法, 表示当节点 n_i 和节点 n_j 相遇时, 两者间消息传输的过程. 利用式(3)计算节点 n_i 和节点 n_j 中消息的调度优先级值, Q_{n_i} 表示节点 n_i 中根据优先级值从高到低排序的消息队列 (Queue), Q_{n_j} 表示节点 n_j 中根据消息的优先级值从高到低排序的消息队列, 而节点 n_j 可复制的节点 n_i 中消息队列长度 Q_{head} 根据其缓存空间大小而定.

算法 2. MOB 的调度算法.

输入: Q_{n_i}, Q_{n_j}

输出: 节点 n_i 将消息队列 Q_{head} 复制到节点 n_j 中

1. IF $Q_{n_i}.length < n_j.freebuffer.size$ THEN
2. sending Q_{n_i} to n_j ;
3. END IF
4. IF Q_{n_i} 中最低优先级 $> Q_{n_j}$ 中最高优先级 THEN
5. sending Q_{n_i} to n_j ;
6. END IF
7. IF Q_{n_i} 中最高优先级 $< Q_{n_j}$ 中最低优先级 THEN
8. // Q_{head} 是 Q_{n_j} 可复制消息队列;
9. $Q_{head}.length < n_j.freebuffer.size$ THEN
10. sending Q_{n_i} to n_j ;
11. END IF
12. END IF

3.3 车载移动模型驱动的缓存管理机制

3.3.1 车载移动模型

随机移动模型常用于模拟二维平面上物联网节点移动方向和速率不受限的移动模式. 然而真实世界中物联网中节点运动却是受限的, 例如物联网中承担感测任务的车辆行驶需要按照预定的道路规划进行, 无法随机选择方向和速率; 在野外作业中, 河流、森林等障碍区域导致节点无法自由移动. 这种情况下, 随机移动模型中提出的基于方向的消息权重不再有效. 一方面, 节点当前位置与消息的目标位置距离不能直接体现消息传至目标区域的概率高低, 例如在单行的道路上, 若消息的目标位置位于逆向的道路上, 即使与当前节点的位置很近也很难被投递至终点; 另一方面, 节点当前位置与消息目标位置连线和节点当前移动方向夹角也不再适用, 方向角小的位置可能由于车辆在路口的突然变向而逐渐远离.

为了模拟物联网中更真实的节点运动场景, 可以通过限制节点移动的自由度来添加约束条件, 使其只能按照预先设定的拓扑进行运动. 考虑到车载

移动终端已被广泛用于数据采集. 本节将重点考虑常见的车载移动模型在物联网中的应用. 在车载移动模型中, 节点的移动性被赋予了一些特征, 如高速公路节点的移动方式受到了严格的限制, 网络中会定期的出现密集和稀疏的区域, 并在交叉路口出现用户群集或交通阻塞来模拟车辆等待红灯和车流高峰时刻. 常见的车载移动模型包括高速公路移动模型(Freeway Mobility model, FM)和曼哈顿网格模型(Manhattan Grid Mobility model, MGM), 它们的运动特征如图 4 所示.

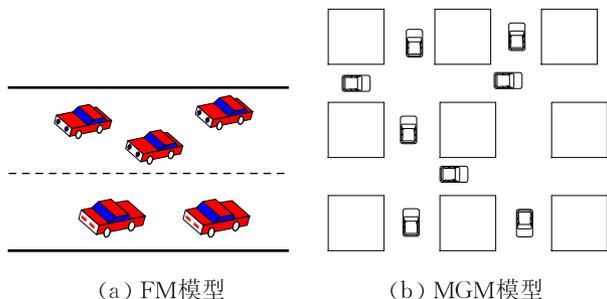


图 4 车载移动模型

图 4(a)所示为 FM 模型, 它使用随机图来模拟城市地图, 即在预先定义好的地图中限定节点在一些通路上移动. 该 FM 将车辆节点限制在一条双向多车道的高速路上, 初始时节点随机放置在图中各边上. 每个节点随机选择一个目的地, 并沿着这条边通过最短路径到达目的地. 一旦到达, 节点暂停 T_{pause} 时间后再选择下一个目的地, 进入下一次移动过程. 它与 RWP 模型的区别在于: (1) 在 FM 模型中, 限定每个移动节点只能在自己道路上行走; (2) 在 FM 模型中, 移动节点当前的速度依赖于它先前的速度; (3) 在同一条高速通路上, 若两个移动节点的距离在安全距离范围内, 后面车辆速度不能超过前面节点的速度.

图 4(b)所示为 MGM 模型, 它使用网格道路的拓扑结构. 移动节点可以沿着水平街道和垂直街道网格进行移动, 经过路口时利用概率来选择移动方向, 例如在十字路口, 移动节点以 0.25、0.25 和 0.5 的概率选择左转、右转或直行. 与 FM 相似, 在同一条车道上, 节点的速度受限于前面的节点速度.

图 5 显示了本文的一个 MGM 模型研究场景. 在一个城市中, 节点均沿着城市的街道进行移动, 一般城市街道均为双向车道, 每个节点都安装有 GPS 系统, 在节点移动中可以知道通信范围内其它节点的移动方向. 假设节点 A 携带有消息 m_1 、 m_2 、 m_3 , 三个消息的目的位置是图中红点所示. 节点 A 的通信

范围是 r , 在通信范围内有三个移动节点, 分别是移动节点 B、C、D. 那么当节点 A 的缓存空间溢出时, 节点 A 将缓存中的消息进行优先级排序, 则 m_3 被丢弃的概率最大, 因为按照节点 A 当前的移动方向和路径, 节点 A 不会到达消息 m_3 的目的位置. 当进行节点调度时, 消息 m_2 传给节点 B 的优先级比传给节点 D 的高, 虽然节点 D 比节点 B 离 m_2 的目的地更近, 但是按照当前移动方向, 节点 D 在远离 m_2 的目标位置, 而按照城市地图路径来说, 节点 B 在靠近 m_2 的目标位置.

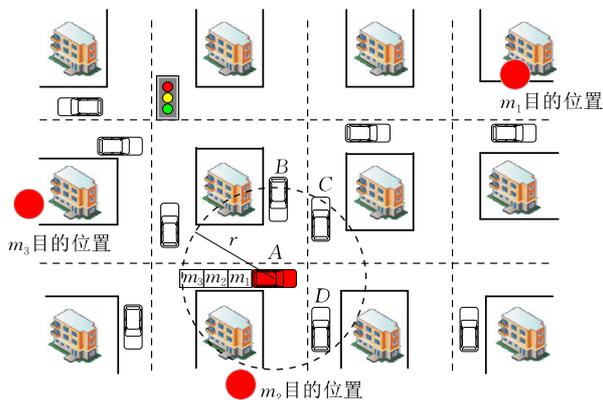


图 5 车载移动模型研究场景

基于 Trace 的移动模型(Trace-Based Mobility model, TBM)则通过收集真实的车辆移动数据提出符合实际情况的移动模型. UMassDieselNet^①、MIT Reality Mining^② 和 Cabspotting^③ 等都对 Trace 进行了收集. UMassDieselNet 提供了 Amherst 市公交系统的移动轨迹数据, 并分析了节点间的相遇规律, 它包含 40 辆公交车, 接触持续的时间相对短暂. Cabspotting 为 San Francisco 的出租车安装 GPS 以及通信模块, 能够周期性地各个出租车的位置信息传回数据中心, 实时观察出租车的移动情况. Reality Mining 项目记录了校园中 100 个携带具备蓝牙功能的智能手机的学生和职工为期 9 个月的移动轨迹和相遇数据. 此外, 文献[22]指出, 不同于服从独立同分布理论的随机移动模型推导出的指数分布, 人类携带的移动设备 Trace 展现出的是幂律分布的特性, 随机移动模型不适合用于模拟人类移动 Trace. 因此, 为了进一步模拟真实场景, 本文对基于上海出租车的真实 GPS 数据构造的 SUVnet

① UMass Diverse Outdoor Mobile Environment (DOME). <http://prisms.cs.umass.edu/dome> 2017, 7, 19

② MIT Media Lab: Reality Mining. <http://reality.media.mit.edu> 2017, 7, 19

③ The Cabspotting Project. <http://canspotting.org>. 2017, 7, 19

移动模型^[23]进行了研究,超过 4000 辆出租车的位置和相遇信息被该数据集记录,表 1 描述了车载 GPS 数据信息字段。

表 1 上海出租车 GPS 数据信息字段

| 参数 | 说明 |
|-------|------------|
| 经度 | 位置信息 |
| 纬度 | 位置信息 |
| 车辆状态 | 1:载客; 0:空客 |
| 采集时间点 | 时间信息 |

3.3.2 基于地理信息的信息权重

FM 模型、MGM 模型和 TBM 模型等具有的共同点是限制了节点的自由度,需要节点按照固定的轨迹移动,在计算消息基于地理信息的权重时,可以不考虑节点与消息目的地之间的距离问题. 因为即使距离很近,节点沿着路径移动也可能需要较多的时间. 所以只考虑节点沿着路径的移动方向. 图 6 显示了如何计算一个基于地理信息的信息权重 φ .

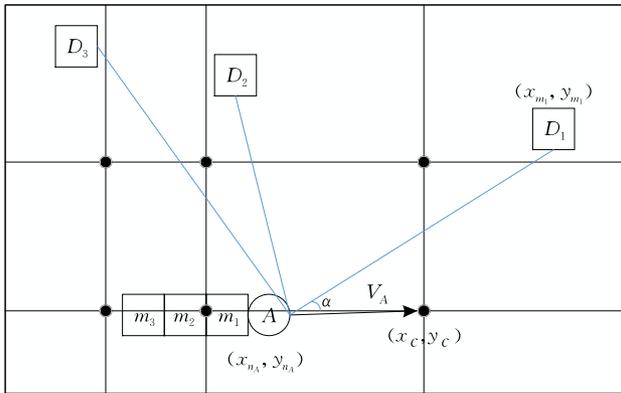


图 6 基于地理信息的信息权重

如图 6 所示,区域内部的直线均表示现实场景中的道路,黑点表示道路交叉的十字路口. 已知区域内部所有十字路口的经纬度信息 (x_c, y_c) , 移动节点 A 的经纬度 (x_{nA}, y_{nA}) , 消息 m_1 的目的位置 (x_{m1}, y_{m1}) , 节点的速度方向 V_A 可以用节点前进方向的十字路口经纬度与当前节点位置信息获得. 那么,基于地理信息的信息 m_j 权重值 φ_{m_j} 可以通过式(4)获得:

$$\varphi_{m_j} = \cos\alpha = \frac{X_A X_B + Y_A Y_B}{\sqrt{X_A^2 + Y_A^2} \cdot \sqrt{X_B^2 + Y_B^2}} \quad (4)$$

其中, $X_A = x_c - x_{nA}$, $X_B = x_{m_j} - x_{nA}$, $Y_A = y_c - y_{nA}$, $Y_B = y_{m_j} - y_{nA}$.

3.3.3 VEB 缓存管理机制

车载移动模型驱动的缓存管理机制 (VEhicular mobility model driven Buffer management mechanism,

VEB) 利用车载移动模型中限制节点自由移动的特性,设计合理的适用于车载模型的缓存管理机制,这个机制同样包含调度策略和丢弃策略。

考虑基于地理信息的信息权重值 φ 、消息的剩余生存时间、网络中副本数以及消息大小,按照式(5)计算消息 m_j 的丢弃权重 pd_{m_j} . 该式同样适用于允许一定消息延迟的低开销应用场景. 丢弃算法按照权重大小从权重最小的消息开始丢弃;另一方面,消息的调度权重 ps_{m_j} 与其相同,调度算法从权重最大的消息开始传递. 即:

$$p_{m_j} = \varphi_{m_j} \cdot (\log(R_{m_j}) + m_j(T_{m_j})) \cdot S_{m_j}^{-1} \quad (5)$$

图 7 展示了本节提出的 VEB 缓存管理机制的算法流程,该流程主要分为两个部分,一部分是缓存溢出时的丢弃策略,另一部分是消息的调度策略. 当节点相遇时,若需进行消息的交换,则首先判断接受消息的节点缓存是否已满,未滿则更新发送消息节点的消息调度权重,根据权重优先交换权值大的消息;缓存已满时则更新消息丢权重并排序,根据排序结果判断接受新消息或丢弃本地消息,丢弃时根据权重优先丢权重值小的消息。

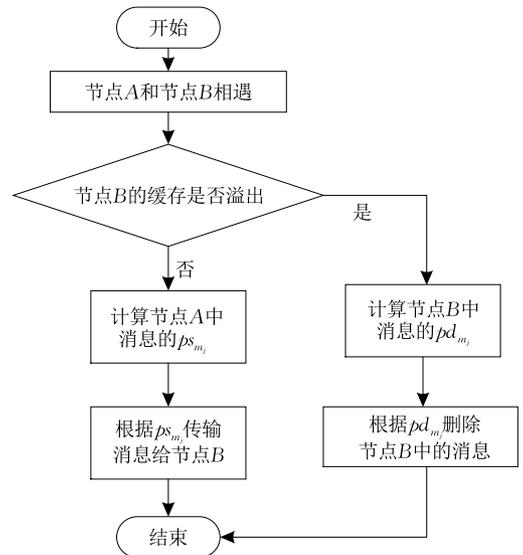


图 7 VEB 缓存管理机制

4 仿真实验

4.1 对比算法与性能指标

实验采用了两种多副本路由策略——Epidemic 路由策略和 Spray&Wait 路由策略,其中 Spray&Wait 按照二分的方式散发消息. 随机移动模型采用 RWP 模型,车载移动模型采用曼哈顿模型和上海出

租车 Trace 模型. 将本文缓存管理机制与以下缓存管理算法进行了对比分析:

- (1) DO: 优先丢弃最早进入节点的缓存消息.
- (2) MOFO: 优先丢弃最多转发次数消息.
- (3) SHLI: 优先丢弃最短剩余生存时间消息.
- (4) GBSD^[16]: 基于全局信息的消息调度和丢弃.

其中, DO、MOFO(MOst-FORwarded)和 SHLI(SHortest-Life-time)^[11]为传统缓存管理算法,文献^[11]指出,在基于队列信息的缓存策略中,MOFO可以获得最佳的消息投递率,而 SHLI则在平均时延上表现最佳,一直以来被视为缓存策略性能对比基准. 而基于全局信息的调度和丢弃策略 GBSD综合考虑了消息的本地状态和部分全局状态,它在最大化投递率和最小化传输时延两个问题上分别计算消息效用,本文所讨论的为其最大化投递率的情况,并对其在 Epidemic 路由策略下的性能表现与本文提出的缓存管理机制进行了对比.

根据物联网数据传输的性能要求,结合本文研究的应用场景,本文将从如下指标来评估缓存管理机制的性能^[24]:

(1) 消息投递率(Delivery Rate). 模拟时间内由源节点生成并成功传送到目的地的消息数量与网络中生成的所有消息数量之比.

(2) 平均时延(Average Latency). 从源节点发送消息开始到消息到达目的地所花费的时间.

(3) 网络负载率(Overhead Ratio). 指成功投递一个消息消耗的转发次数,即网络中转发的消息次数减去成功投递的消息数目,再与成功投递消息数目相比. 网络开销越低,网络转发越高效.

(4) 平均跳数(Average Hop). 网络中所有消息的副本经历的跳数的总和与网络中生成的所有消息数量之比.

4.2 实验环境

使用 ONE(Opportunistic Network Environment simulator)仿真工具^[25],开展相关测试. 表 2 列出了 RWP 模型设置参数情况,该设置基于 ONE 提供的默认场景,它也是机会网络仿真中使用最多的场景. 本文在此基础上针对物联网数据采集需求和移动设备特点进行了少许修改. 仿真场景的大小为 1000 m×1000 m; 仿真时长 12 h; 区域中移动节点数量为 40 个; 模拟人行速度, 移动节点的移动速度在 0.5 m/s~1.5 m/s 之间变化; 为了适用 ZigBee、蓝牙、Wi-Fi 等短距离无线通信技术, 设置消息传输带宽 250 kbit/s, 传输半径为 100 m; 考虑到物联网中消息大小不一,

设置消息大小在 500 Kb(普通文本消息)~1024 Kb(一般图片消息)范围变化; 物联网中的消息,如车流、环境气候等有一定的时效性,设置消息的生存时间在 30 min~90 min 间变化; 为模拟不同的消息生成频率,设置整个网络消息生成间隔时间变化范围为 1 s(消息生成频度高)~40 s(消息生成频度低). 消息的源节点为网络中随机指定的某个移动节点,消息的目的地为随机选取的某个区域. MGM 模型仿真区域内有 120 个移动节点,其余参数与 RWP 模型相同. 上海出租车 Trace 信息字段如上文中表 1 所示.

表 2 RWP 模拟参数设置

| 参数 | 默认值 |
|------|-----------------|
| 场景大小 | 1000 m×1000 m |
| 节点数量 | 40 |
| 传输速度 | 250 kbit/s |
| 消息大小 | 500 Kb~1024 Kb |
| 时间间隔 | 1 s~40 s |
| 传输半径 | 100 m |
| 移动速度 | 0.5 m/s~1.5 m/s |
| 生存时间 | 30 min |
| 仿真时间 | 12 h |

4.3 实验结果及性能分析

4.3.1 MOB 机制仿真结果及性能分析

为了模拟网络中的不同拥塞级别,我们考察 MOB 机制在不同缓存大小下的性能表现. 缓存大小变化范围为 2 MB~10 MB,缓存大小为 2 MB 时节点可以缓存的消息数少,网络处于高拥塞状态;缓存大小为 10 MB 时节点缓存队列变长,投递率等性能指标趋于稳定.

图 8 描述了在 Epidemic 和 Spray&Wait 路由策略下缓存空间的变化对消息投递率造成的影响. 随着缓存空间的变大,节点携带的消息变多,节点间相遇时会交换更多的消息,消息投递到目的地的概率增大,消息的投递率随着缓存空间增大而增大. 从图 8(a)和图 8(b)可以看出,本文提出的 MOB 缓存管理机制投递率始终高于其它缓存管理机制,可保

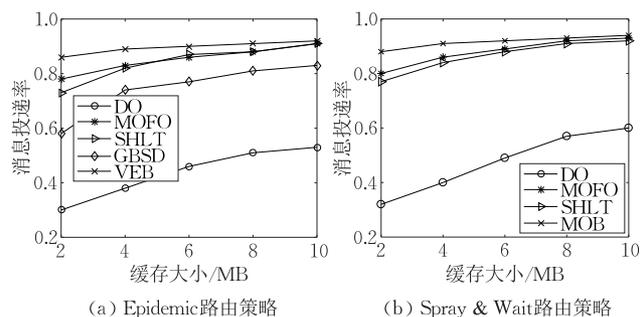


图 8 投递率随缓存大小变化

证投递率达 80%。此外,运用 Spray&Wait 路由策略相比于用 Epidemic 路由策略投递率明显要高,这是因为前者由于缓存的限制以及消息的副本数的无限制导致消息频繁地被丢弃,而后者由于拥有副本数控制机制使得缓存溢出的情况得到了缓解。仿真结果表明,DO 算法在消息投递率方面远远低于其它算法,MOFO 和 SHLI 算法的性能相近,而本文提出的 MOB 则始终优于其它算法。

图 9 描述了两种路由策略下消息的平均时延随着缓存空间变化的情况。增大节点缓存空间可提高消息被中继节点缓存的概率,从而使得消息以相对较快的速率到达目的位置。因此,随着缓存空间的逐渐增大,消息的投递时延应该呈现逐渐降低趋势。但是,当缓存空间增大到一定的时候,消息的平均时延会趋于平稳,此时网络中需要传递的消息总数与节点的缓存空间之间达到一个稳态。当缓存空间较小的时候,本节提出的 MOB 缓存管理机制并没有优势,这是由于 SHLI 优先丢弃剩余生命时间最短的消息,而 DO 优先丢弃队列中最早的消息,很大程度上也是队列中剩余生命时间最短的消息,而本文中的丢弃策略重在保证消息的投递率,虽然考虑了消息剩余生命时间,但当网络中副本数的影响超过时间影响时,会进行时延上次优的丢弃决策。注意到缓存空间超过 6 MB 时,本文的 MOB 策略在平均时延上开始优于其它策略,随着硬件水平的不断提高,物联网中移动终端的缓存空间很容易满足该条件,因此本文中的 MOB 缓存管理机制在现实场景中仍保持一定优势。

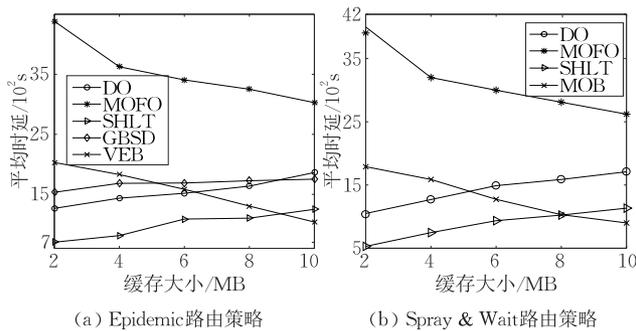


图 9 平均时延随缓存大小变化

图 10 为在两种路由策略下缓存空间的变化对网络负载率造成的影响。缓存空间增大使得节点携带的消息数量增多,消息的投递概率增大,成功投递的消息数量增多,网络负载率逐渐降低。在 Spray&Wait 路由策略下,缓存管理算法的网络负载率明显低于在 Epidemic 路由策略下,原因在于副本数量得到了控制。在这些算法中,DO 算法相较于

其它算法网络负载率明显较高,而其它算法的网络负载率比较接近。当缓存空间为 2 MB 时,本文提出的 MOB 机制网络开销较 MOFO 稍大,这是因为 MOFO 无论何时都优先丢弃转发次数最多的消息,避免了此类消息继续在网络中扩散,而本文的 MOB 机制则更关注消息的投递率,副本数多的消息传输速度相对较快,不会被优先丢弃。注意到缓存空间大于 6 MB 时,本节提出的 MOB 网络开销比较接近于 MOFO,因此一定程度上减少了网络资源的浪费。

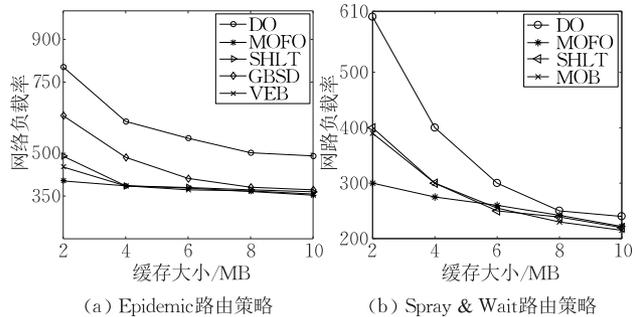


图 10 网络负载率随缓存大小变化

图 11 为在两种路由策略下消息副本的传输跳数随节点缓存空间大小的变化情况。随着节点缓存空间的增大,节点传输的消息副本增加,消息可以获得更大的投递率和更少的平均跳数。MOFO 算法和其它算法相比平均跳数稍高,这是因为 MOFO 中转发次数最多的消息其传输过程中平均跳数也最多。而本节提出的 MOB 缓存管理机制,由于在消息存储过程中,会按照投递目标进行存储,具有方向性和目标性,所以投递后平均跳数相对较低。

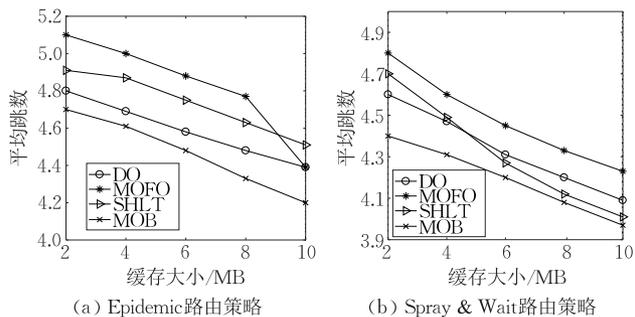


图 11 平均跳数随缓存大小变化

综上所述,本文提出的 MOB 缓存管理机制在消息投递率上表现最优,保证了多副本路由策略在缓存空间有限的情况下的消息投递成功率。当缓存空间小于 6 MB 时,本文的 MOB 策略在平均时延和网络负载率上付出了一定的开销,但仍接近于在这两个指标中表现最好的 SHLI 和 MOFO 策略。而在缓存空间大于 6 MB 的场景中,本文的 MOB 策略可以在提高消息的投递率同时保持较低的网络时延,

在降低平均跳数的同时减少网络负载率. 考虑到该缓存大小在物联网移动终端中可得到满足, 因此, MOB 缓存管理机制在 RWP 模型中有较好的性能表现.

4.3.2 VEB 仿真结果及性能分析

相较简单的传感器节点, 车载移动终端往往具备了简单的计算功能, 在物联网中可以当作数据汇聚节点. 车载网中节点数量多, 移动速度更快, 道路结构复杂, 节点移动方式受限, 节点间通信时间相对较短, 网络环境更为复杂, 给消息传输带来不确定性. 且车载移动终端硬件设备更为完善, 可以获取更多的数据类型, 缓存空间相对充足. 因此在对车载移动模型进行仿真时, 本文将缓存大小范围设置为 2 MB~20 MB, 更清晰地展现 VEB 策略在仿真场景下的性能表现.

图 12 描述了在曼哈顿模型和上海出租车 Trace 中, 四种缓存管理算法 DO、MOFO、SHLI 和 VEB 随着缓存空间增大消息投递率的变化情况. 结果表明 VEB 缓存管理机制在消息投递率方面明显优于其它三种经典算法, 这是因为 VEB 缓存管理机制运用节点移动的信息, 可以更准确的投递消息, 而 DO 算法的表现始终欠佳.

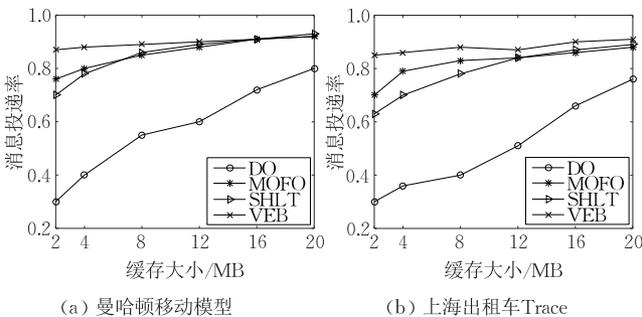


图 12 投递率随缓存大小变化

图 13 描述了在曼哈顿模型和上海出租车 Trace 中, 四种缓存管理机制的消息平均时延随着缓存空间变化的情况. 从图中可以看出, 相比于 SHLI 和 DO 策略, 当缓存空间较小的时候, VEB 缓存管理机

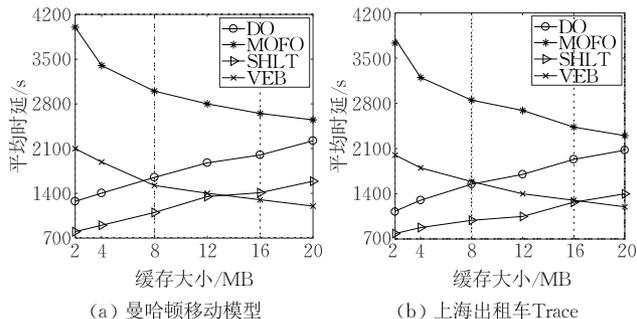


图 13 平均时延随缓存大小变化

制并没有表现良好, 这与 RWP 模型中的仿真结果相似. 注意到缓存空间逐渐增大时, VEB 的平均时延会渐渐降低到一个比较低的值. 所以, 本节的 VEB 缓存管理机制在缓存空间比较大时仍保持平均时延上的较优表现, 比较适用于车载移动模型.

图 14 描述了在曼哈顿模型和上海出租车 Trace 中, 缓存空间变化时网络负载率的变化情况. 在这四种算法中, DO 算法相较于其它三种算法在两种模型下网络负载率一直比较高, 而其它三种算法的网络负载率则相对比较接近. VEB 网络负载率在两种模型下和另外两种基本相似, 但随着缓存空间的增大, 网络负载率稍小于另外两种, 相对而言减少了网络资源的浪费.

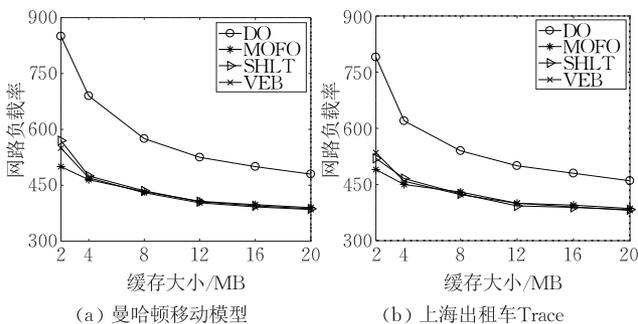


图 14 网络负载率随缓存大小变化

图 15 为在曼哈顿模型和上海出租车 Trace 中, 平均跳数随着缓存空间增大的变化情况. 综合图 15(a)和图 15(b)来看, 这四种算法的平均跳数都会随着缓存空间的增大而减小, 因为当节点的缓存空间增大时, 节点携带的消息更多, 丢弃消息的频率降低, 从而有更多的消息会尽快到达目的区域, 则消息传递需要的平均跳数会减少. VEB 缓存管理机制在两种模型下相较于其它三种算法, 平均跳数稍有优势, 均低于其它三种算法的平均跳数.

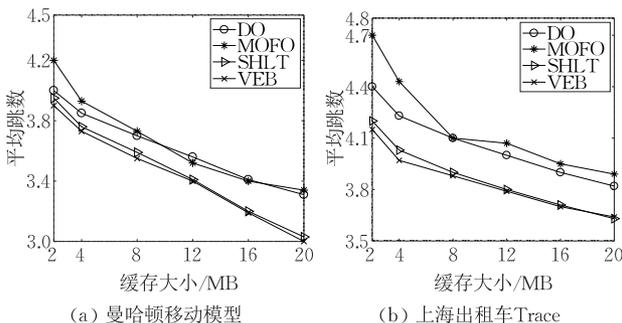


图 15 平均跳数随缓存大小变化

综合以上在两种模型下的四个仿真性能对比试验, 可以看出本文提出的车载移动模型驱动的 VEB 缓存管理机制在车载移动模型和基于 Trace 的移动

模型中依旧保持与随机移动模型 RWP 中一致的性能表现,在提高消息投递率的同时保持了平均时延、网络负载率和平均跳数上的低消耗。

5 结束语

物联网作为一种新型的物物互联的网络体系结构,近年来得到了广泛的关注. 本文对节点位置信息驱动的物联网缓存管理机制进行了重点研究,结合不同的移动模型中节点位置和移动方向特征,制定了不同的缓存管理机制来提高网络性能,并依次从理论推导、算法研究和实验仿真三方面对相关方法进行分析与验证. 依据本文提出的相关方法为设计物联网终端节点的缓存管理策略奠定了坚实的基础,为物联网数据采集方案的性能提升提供了相应手段.

致 谢 感谢中国教育和科研计算网华东(北)地区网络中心对本文实验数据提供的存储和计算服务!

参 考 文 献

- [1] Want R, Schilit B N, Jenson S. Enabling the Internet of Things. *Computer*, 2015, 48(1): 28-35
- [2] Ngu A H H, Gutierrez M, Metsis V, et al. IoT Middleware: A survey on issues and enabling technologies. *IEEE Internet of Things Journal*, 2017, 4(1): 1-20
- [3] Vahdat A, Becker D. Epidemic routing for partially-connected Ad Hoc networks. Duke University, North Carolina, USA; Technical Report CS-2000-06, 2000
- [4] Spyropoulos T, Psounis K, Raghavendra C S. Spray and wait: An efficient routing scheme for intermittently connected mobile networks//*Proceedings of the ACM SIGCOMM Workshop on Delay-Tolerant Networking*. Philadelphia, Pennsylvania, USA, 2005: 252-259
- [5] Lindgren A, Doria A, Schelen O. Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE Mobile Computing & Communications Review*, 2003, 7(3): 19-20
- [6] Zhang X, Neglia G, Kurose J, et al. Performance modeling of epidemic routing//*Proceedings of the International Conference on Research in Networking*. Berlin, Germany, 2006: 827-839
- [7] Falaki H, Mahajan R, Kandula S, et al. Diversity in smartphone usage//*Proceedings of the International Conference on Mobile Systems, Applications, and Services*. New York, USA, 2010: 179-194
- [8] Jin L, Chen Y, Wang T, et al. Understanding user behavior in online social networks: A survey. *IEEE Communications Magazine*, 2013, 51(9): 144-150
- [9] Camp T, Boleng J, Davies V. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing*, 2002, 2(5): 483-502
- [10] Zhang X, Kurose J, Levine B N, et al. Study of a bus-based disruption-tolerant network-mobility modeling and impact on routing//*Proceedings of the International Conference on Mobile Computing and Networking*. Montréal, Québec, Canada, 2007: 195-206
- [11] Lindgren A, Phanse K S. Evaluation of queuing policies and forwarding strategies for routing in intermittently connected networks//*Proceedings of the International Conference on Communication System Software and Middleware*. New Delhi, India, 2006: 1-10
- [12] Burgess J, Gallagher B, Jensen D, et al. MaxProp: routing for vehicle-based disruption-tolerant networks//*Proceedings of the IEEE International Conference on Computer Communications*. Barcelona, Spain, 2007: 1-11
- [13] Liu Y, Wang J, Zhang S, et al. A buffer management scheme based on message transmission status in delay tolerant networks//*Proceedings of the Global Telecommunications Conference*. Kathmandu, Nepal, 2011: 1-5
- [14] Wei K, Guo S, Zeng D, et al. A multi-attribute decision making approach to congestion control in delay tolerant networks//*Proceedings of the IEEE International Conference on Communications*. Sydney, Australia, 2014: 2742-2747
- [15] Balasubramanian A, Levine B, Venkataramani A. DTN routing as a resource allocation problem//*Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*. New York, USA, 2007: 373-384
- [16] Krifa A, Spyropoulos T, Spyropoulos T. Message drop and scheduling in DTNs: Theory and practice. *IEEE Transactions on Mobile Computing*, 2012, 11(9): 1470-1483
- [17] Shin K, Kim S. Enhanced buffer management policy that utilises message properties for delay-tolerant networks. *IET Communications*, 2011, 5(6): 753-759
- [18] Chitti S H, Prasad N R, Ovsthus K, et al. A novel buffer management for delayed delivery traffic in future networks//*Proceedings of the International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronics Systems*. Aalborg, Denmark, 2014: 1-5
- [19] Kaveevitvachai S, Ochiai H, Esaki H. Message deletion and mobility patterns for efficient message delivery in DTNs//*Proceedings of the IEEE International Conference on Pervasive Computing and Communication Workshops*. Mannheim, Germany, 2010: 760-763
- [20] Broch J. A performance comparison of multi-hop wireless ad hoc network routing protocols//*Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*. New York, USA, 1998: 85-97

- [21] Bettstetter C. Mobility modeling in wireless networks; Categorization, smooth movement, and border effects. ACM SIGMOBILE Mobile Computing & Communications Review, 2001, 5(3): 55-66
- [22] Le T, Kalantarian H, Gerla M. A buffer management strategy based on power-law distributed contacts in delay tolerant networks//Proceedings of the International Conference on Computer Communication and Networks. Waikoloa, USA, 2016; 1-8
- [23] Luo P, Huang H, Shu W, et al. Performance evaluation of vehicular DTN routing under realistic mobility models// Proceedings of the Wireless Communications and Networking Conference. Las Vegas, USA, 2008; 2206-2211
- [24] Zhang L, Yu B, Pan J. GeoMob: A mobility-aware geocast scheme in metropolitans via taxicabs and buses//Proceedings of the IEEE Conference on Computer Communications. Toronto, Canada, 2014; 1279-1787
- [25] Keränen A, Ott J, et al. The ONE simulator for DTN protocol evaluation//Proceedings of the International Conference on Simulation TOOLS and Techniques. Brussels, Belgium, 2009; 55



TAO Jun, born in 1975, Ph. D., professor. His research interests include Internet of Things, wireless Ad Hoc networks (wireless sensor networks, mobile social networks and vehicular networks).

SHI Shu-Jing, born in 1994, M. S. candidate. Her research interest is Internet of Things technology.

FENG Fu-Qin, born in 1992, M. S. Her research interest is Ad Hoc networks.

GAO Yang, born in 1989, Ph. D. candidate. His research interest is wireless networks.

Background

Internet of Things (IOT) aims at realizing automatic and real time object recognition, tracking, monitoring and events triggering. As a combination of physical objects, sensors and networks, message interaction among different nodes have become a key research point in IOT. Nowadays there have been lots of works focusing on data collecting, forwarding and computing. However, the limited buffer size and energy supply of IOT nodes have made traditional routing policies restricted, thus efficient buffer management becomes an important role in promoting IOT performance.

Typical buffer management policies are mostly based on characteristics of message queues. Some enhanced methods take either local or global attributes of messages into consideration. However, they seldom consider nodes' mobility.

Analysis of mobility models of nodes in IOT can help improve the message delivery ratio and reduce the network overhead.

In this paper, a mobility model driven buffer management of node ends in IOT has been proposed. The nodes' mobility model, together with the global network knowledge, is applied to conduct the message scheduling and dropping. The simulation results reveal that the proposed mechanism outperforms other traditional buffer management policies in terms of the message delivery ratio, the average delay, the network overload and the average hops.

This work is supported by the National Natural Science Foundation of China (No. 61370209) and the Jiangsu Provincial Natural Science Foundation (No. BK20151416).