

引用格式:严圣华,王昌达.物联网实时任务传输中链路传感节点的内存效率优化[J].科学技术与工程,2018,18(3):57—62  
Yan Shenghua, Wang Changda. Memory efficiency optimization of link sensing nodes in real-time task transfer of Internet of things[J]. Science Technology and Engineering, 2018, 18(3): 57—62

# 物联网实时任务传输中链路传感节点的内存效率优化

严圣华 王昌达\*  
(江苏大学研究生院,镇江 212016)

**摘要** 在物联网实时任务传输中,不同应用对链路传感节点内存的需求量不同,容易出现部分节点空间内存多,部分节点内存紧张的情况。为此,提出一种新的链路传感节点内存效率优化方法。介绍了 TLSF 的二级离散表管理机制和 TLSF 方法的弊端,在此基础上,通过 TC-TLSF 方法对物联网实时任务传输中链路传感节点内存效率进行优化处理。TC-TLSF 方法通过模糊控制判断链路传感节点释放内存的过程中是否马上将空闲内存块结合在一起,通过最大隶属度原则对待释放内存块的模糊特征参数所属等级进行判断,求出合并系数,对得到的系数和动态阈值进行比较,判断是否延迟合并操作。介绍了模糊特征参数和模糊规则,确定动态阈值。通过对内存分配和内存释放的控制实现内存效率优化。实验结果表明,所提方法内存效率高。

**关键词** 物联网 实时任务 传输 链路传感节点 内存效率

**中图法分类号** TP302; **文献标志码** A

近年来,网络和信息技术逐渐发展,物联网变成网络应用的关键,其涉及无线传感技术和检测技术等多种技术<sup>[1]</sup>。物联网是一种新兴技术,主要包括网络层、感知层与应用层。在物联网中,不同层之间需完成实时任务传输,完成远程控制<sup>[2,3]</sup>。

随着用户和信息量的逐渐增加,人们对链路传感节点内存效率的要求越来越高<sup>[4]</sup>。所以,怎样高效管理内存,提高内存利用率,已经变成当前物联网领域亟需解决的问题。

在物联网实时任务传输中,本文提出一种新的链路传感节点内存效率优化方法。实验结果表明,所提方法在内存效率方面的性能很高。

## 1 TLSF 内存效率优化方法概述

### 1.1 TLSF 的二级离散表管理机制

TLSF 方法通过二级离散表对内存进行管理。通过 2 的幂次方将第一级空闲内存划分成  $n$  个等级,索引  $i$  代表内存是  $[2^i, 2^{i+1})$ ; 第二级利用线性分割把  $[2^i, 2^{i+1})$  平均划分成  $m$  个部分<sup>[5]</sup>。针对各

级,均通过一个位图进行索引,对该区间中是否存在空闲内存进行标识,1 代表有,0 代表无。

### 1.2 TLSF 方法的弊端

虽然 TLSF 方法的执行时间短,然而其选择的立即合并机构在对大小相近内存进行重复分配的过程中,未有效利用重复分配的优势,所以性能不佳<sup>[6]</sup>。

其弊端主要有:在物联网实时任务传输过程中,对链路传感节点内存进行分配时,立即合并机制在很大程度上会使计算开销大大提高;在对内存进行释放的过程汇总,立即合并机制会造成额外开销,如果不对释放的内存块进行合并,则会减少计算复杂度,加快释放速度<sup>[7]</sup>。

## 2 物联网实时任务传输中链路传感节点内存效率优化

现通过 TC-TLSF 方法对物联网实时任务传输中链路传感节点内存效率进行优化处理,其主要通过模糊控制阶段和动态阈值确定阶段实现。

### 2.1 模糊控制方法

TC-TLSF 方法通过模糊控制判断链路传感节点释放内存的过程中是否马上将空闲内存块结合在一起,详细流程图见图 1。

通过最大隶属度原则对待释放内存块的模糊特征参数所属等级进行判断,求出合并系数,对得到的

2017 年 6 月 27 日收到  
第一作者简介:严圣华(1975—),男,江苏镇江人,硕士,副教授。研究方向:物联网、计算机网络、信息管理等。E-mail: ysh201701@163.com。  
\*通信作者简介:王昌达(1971—),男,江苏镇江人,博士,博士研究生导师,教授。研究方向:信息安全技术、网络通信与云计算。

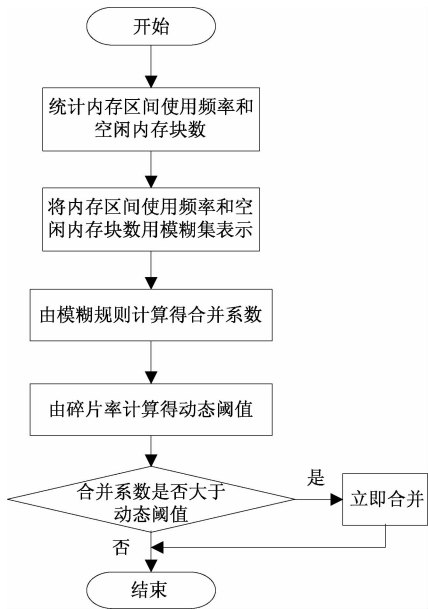


图 1 模糊控制方法流程图  
Fig. 1 Flow chart of fuzzy control algorithm

系数和动态阈值进行比较,判断是否延迟合并操作<sup>[8,9]</sup>。在物联网实时任务传输中,文章分析的模糊特征参数通常为内存区间使用频率与空闲内存块个数,一般将三角形隶属度函数看作模糊输入和输出隶属度函数。

2.1.1 模糊特征参数

物联网实时任务传输中,链路传感节点内存区间的使用频率即某周期内采用某区间内存块的概率,其能够反映相应内存区间对物联网的重要程度<sup>[10]</sup>。物联网工作一阶段后,通过区间 $s_j$ 的内存分配次数占总分配次数的比值( $C_{s_j}/C_{all}$ )对内存区间 $s_j$ 的使用频率进行描述,记作 $F(s_j)$ 。

假设物联网实时任务传输中链路传感节点内存区间总量是 $N$ ,则其需符合式(1)所示的条件:

$$\begin{cases} 0 \leq F(s_j) \leq 1, & j = 1, 2, \dots, N \\ \sum_{j=1}^N F(s_j) = 1 \end{cases} \quad (1)$$

通过历史内存使用频率对当前内存分配状态进行描述,从而判断是否推迟合并<sup>[11]</sup>。然而历史内存分配状态对当前时刻的影响是随时间间隔的变化而变化的,时间间隔越大,历史分配状态对当前时刻的影响越小。现通过遗忘因子求出当前内存使用频率,公式为

$$F(s_j, 0) = \sum_{t=1}^m f(t) F(s_j, t) \quad (2)$$

式(2)中, $t$ 为间隔时间; $m$ 为从物联网工作开始到当前时刻运算的总次数; $f(t)$ 为遗忘因子; $F(s_j, t)$ 为内存区间 $j$ 距离此刻的内存使用率。在 $t = 0$ 的

情况下,获取内存区间 $j$ 当前内存使用频率。

2.1.2 模糊规则

将物联网实时任务传输中链路传感节点内存使用频率与空闲内存的数量看作模数输入<sup>[12]</sup>,获取二者的隶属度函数,见图2和图3。

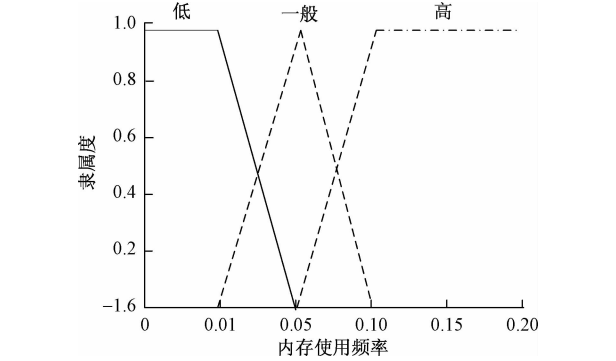


图 2 内存使用频率隶属度函数  
Fig. 2 Memory usage frequency membership function

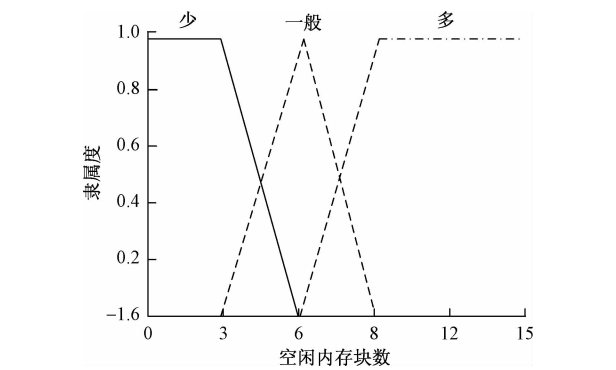


图 3 空闲内存块数隶属度函数  
Fig. 3 Free memory blocks membership function

为了便于计算,加快物联网实时任务传输速率,通过合并系数控制表代替模糊输出隶属度函数<sup>[13,14]</sup>,合并系数控制表见表1。通过表1中的数据可获取模糊输出,也就是合并系数,将其和动态阈值进行比较,从而判断是否马上进行合并操作。

表 1 合并系数控制表  
Table 1 Merger factor control table

| 内存关注度 | 块数   |      |      |
|-------|------|------|------|
|       | 少    | 一般   | 多    |
| 低     | 0.27 | 0.51 | 0.75 |
| 一般    | 0.21 | 0.36 | 0.51 |
| 高     | 0.15 | 0.21 | 0.27 |

2.2 动态阈值确定

动态阈值主要取决于物联网的碎片率,其可体现某一阶段物联网的碎片率状态,在碎片率发生变化的情况下,动态阈值也相应出现改变,然而其在某段时间内可保持不变<sup>[15]</sup>。通过物联网实时任务传输中实际分配内存的最大值和请求内存最大值之差

占请求内存最大值的比计算碎片率,公式为

$$F = (M_{\text{real}} - M_{\text{req}}) / M_{\text{req}} \tag{3}$$

式(3)中,  $F$  为碎片率,  $M_{\text{real}}$  为物联网实时任务传输中实际分配内存的最大值,  $M_{\text{req}}$  为请求内存最大值之差的占请求内存最大值,在物联网实时任务传输处于正常状态下,  $M_{\text{real}} \geq M_{\text{req}}$ 。

随着碎片率的逐渐增加,合并阈值将逐渐降低<sup>[16]</sup>。在合并系数一样的情况下,空闲内存块被合并的数量越多,碎片率越低。如果碎片率较低,则会造成合并阈值增加。动态阈值可通过式(4)求出。

$$T = \begin{cases} 0, & F > F_h \\ (F_h - F) / (F_h - F_1), & F_1 \leq F \leq F_h \\ 1, & F < F_1 \end{cases} \tag{4}$$

式(4)中,  $T$  为合并阈值。通过  $F_h$  与  $F_1$  对物联网内存碎片率进行控制,不管此刻合并系数值是多少,均需马上合并;在  $F < F_1$  的情况下,延迟合并<sup>[17]</sup>。

### 2.3 FTC-TLSF 方法描述

FTC-TLSF 内存效率优化方法通过对内存分配和内存释放的控制实现内存效率优化。内存分配的基本思想如下:首先找到大小是  $r$  的一级索引  $f$  与二级索引  $s$ , 对一个阶段内存总分配次数与  $f$ 、 $s$  索引的内存区间的分配次数,按照  $f$ 、 $s$  值获取合理的内存块与链表的表达,如果是空,则返回错误。再对某阶段内各内存区间的使用频率进行更新处理,求出碎片率,对阈值进行更新,从对应的链表中获取内存块,同时对一二级位图进行调整;最后,对新获取内存块大小是否需分割进行判断,如果需要,则对其进行分割,获取和剩余内存块相应的  $f'$ 、 $s'$ , 把剩余内存块插入相应链表,并返回相应地址。

FTC-TLSF 方法内存释放的基本思想如下:首先按照释放内存块值获取一级和二级索引值  $f$ 、 $s$ , 求出与其相应的内存区间合并系数;再比较合并系数与阈值,如果合并系数超过阈值,则将其和前后空闲内存块合并在一起,通过依据新内存块值求出  $f$ 、 $s$ , 最后把内存块插入  $f$ 、 $s$  描述的内存区间链表中。

## 3 实验结果分析

### 3.1 运行效率和内存管理空间开销分析

本节从运行效率和内存管理空间开销两个方面,对提出的内存效率优化方法和当前常见的分区方法、伙伴系统方法进行比较。测试样本数据是随机形成的 500、5 000 与 50 000 个请求,其主要由内存分配请求与内存释放请求构成,总内存依次是 500 内存块、5 000 内存块、50 000 内存块,每个内存块是 5 KB。

表 2 和表 3 分别为本文方法、分区方法、伙伴系

统方法运行时间和内存管理空间开销比较结果。

表 2 3 种方法运行时间比较结果  
Table 2 Comparison results of running time between the three methods m/s

| 内存块数   | 方法     | 请求组数 |       |        |
|--------|--------|------|-------|--------|
|        |        | 500  | 5 000 | 50 000 |
| 500    | 本文方法   | 1.9  | 8.7   | 67.9   |
|        | 分区方法   | 2.2  | 9.4   | 76.2   |
|        | 伙伴系统方法 | 5.8  | 26.5  | 116.7  |
| 5 000  | 本文方法   | 3.8  | 13.3  | 93.3   |
|        | 分区方法   | 6.3  | 29.1  | 113.1  |
|        | 伙伴系统方法 | 4.9  | 16.7  | 135.7  |
| 50 000 | 本文方法   | 6.1  | 17.7  | 132.8  |
|        | 分区方法   | 6.9  | 22.3  | 144.5  |
|        | 伙伴系统方法 | 7.5  | 175.2 | 176.9  |

表 3 3 种方法内存开销比较结果  
Table 3 Comparison results of memory overhead between the three methods KB

| 内存块数   | 方法     | 请求组数 |       |        |
|--------|--------|------|-------|--------|
|        |        | 500  | 5 000 | 50 000 |
| 500    | 本文方法   | 8.1  | 13.5  | 14.1   |
|        | 分区方法   | 12.2 | 17.5  | 22.3   |
|        | 伙伴系统方法 | 11.8 | 18.9  | 23.7   |
| 5 000  | 本文方法   | 8.9  | 75.3  | 159.4  |
|        | 分区方法   | 32.3 | 98.5  | 196.7  |
|        | 伙伴系统方法 | 22.7 | 105.9 | 223.8  |
| 50 000 | 本文方法   | 9.2  | 73.8  | 209.5  |
|        | 分区方法   | 13.5 | 112.6 | 298.3  |
|        | 伙伴系统方法 | 15.8 | 109.7 | 273.5  |

分析表 2 和表 3 中的数据可知,本文方法在时间效率方面明显高于分区方法和伙伴系统方法,尤其是针对请求组数和内存较大的情况,本文方法的优势更显著。

从管理内存开销的角度分析,和分区方法、伙伴系统方法相比,本文方法的内存开销明显更少,这主要是因为分区方法和伙伴系统方法处理的链路传感节点较多,大大增加了内存开销。

### 3.2 读写性能测试

读写性能是衡量物联网实时传输中链路传感节点内存效率优化有效性的关键指标,本节将分区方法、伙伴系统方法作为对比,对本文方法想性能进行测试。表 4 描述的是 3 种方法下文件读写速度比较结果。

分析表 4 可知,文件块越大,则写入带宽与读取带宽值越高。但本文方法的写入带宽与读写带宽值一直高于分区方法和伙伴系统方法,说明本文方法的读写速率优于其他两种方法。

表 4 3 种方法下文件读写速度比较结果  
Table 4 Comparison of document read and write speed in three methods

| 方法     | 文件块大小/KB | 写入带宽/(GB·s <sup>-1</sup> ) | 读取带宽/(GB·s <sup>-1</sup> ) |
|--------|----------|----------------------------|----------------------------|
| 本文方法   | 1        | 2.15                       | 4.79                       |
|        | 5        | 3.22                       | 7.52                       |
| 分区方法   | 1        | 0.16                       | 0.93                       |
|        | 5        | 0.45                       | 2.68                       |
| 伙伴系统方法 | 1        | 0.29                       | 1.71                       |
|        | 5        | 1.07                       | 2.39                       |

3.3 Mono 测试

Mono 测试即令物联网负载量从低开始申请,每间隔相同周期增加相同的负载量,直到负载达到最高。然后令负载量从高开始释放,每间隔相同的手气减少相同的负载量,直到负载量达到最低。

Mono 测试主要是为了验证本文方法的功能有效性,详细的实验配置见表 5。

表 5 实验详细配置  
Table 5 Detailed configuration of the experiment

| 虚拟机/台 | 初始内存/MB | 最大内存/GB | 总内存/GB | 预留值/GB |
|-------|---------|---------|--------|--------|
| 3     | 500     | 2       | 2      | 200    |

在 Mono 测试过程中,通过本文方法得到的物联网实时任务传输中,虚拟机内存变化情况见图 4。

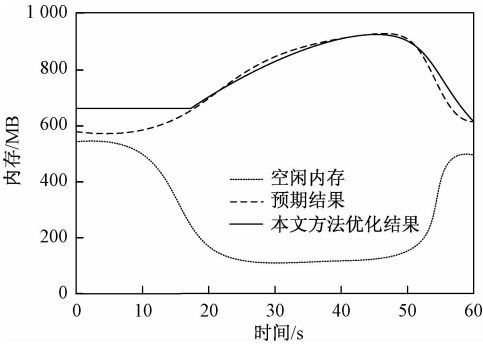


图 4 本文方法下虚拟机内存变化曲线  
Fig. 4 The memory change curve of the virtual machine under this method

分析图 4 可以看出,在实验初始状态下,当物联网实时任务传输负载增加的情况下,本文方法下虚拟机内存无显著性变化,而在空闲内存达到预留值 200 MB 后,虚拟机内存才开始出现变化,呈先上升后下降的趋势。除此之外还可以看出,本文方法对内存效率的优化结果和预期结果基本相同,可有效

实现内存的合理分配。

3.4 静态负载测试

静态负载测试主要利用人工操作在物联网中不同虚拟机中申请负载,对物联网实时任务传输时遭遇的不同内存问题进行模拟,对不同虚拟机内存变化情况进行分析。该实验将 NBD 方法和模糊阈值方法作为对比,观察本文方法在内存效率优化方面的优势。

该实验详细配置见表 6。

表 6 实验详细配置  
Table 6 Detailed configuration of the experiment

| 虚拟机/台 | 初始内存/MB | 最大内存/GB | 总内存/GB | 预留值/GB |
|-------|---------|---------|--------|--------|
| 12    | 500     | 2       | 6      | 120    |

实验在 5 s 时,向物联网中输入内存干扰。图 5 是在物联网实时任务传输中遭遇内存干扰时,本文方法、分区方法、伙伴系统方法的内存变化情况。

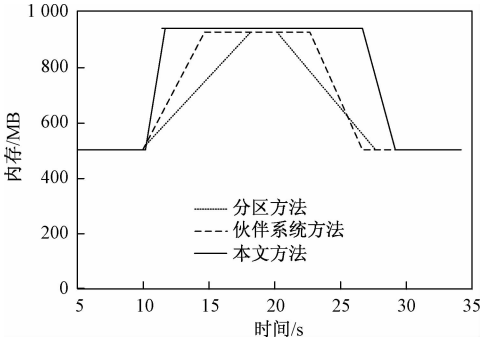


图 5 受干扰情况下 3 种方法内存变化曲线  
Fig. 5 Memory change curve of three methods in case of interference

分析图 5 可以看出,在出现干扰的情况下,本文方法能够快速调整至目标内存,明显优于分区方法和伙伴系统方法,说明本文方法的内存效率更优。还可以看出本文方法的适应能力较强,受干扰影响程度较小,性能更优。

3.5 内存分配利用率测试

内存分配利用率能够有效体现链路传感节点内存效率优化效果,本节将分区方法和伙伴系统方法作为对比,对 100 次、500 次、1 000 次、3 000 次、5 000 次、7 000 次和 10 000 次内存分配时间和内存利用率进行比较,结果用表 7 进行描述。

分析表 7 可知,在相同内存分配次数下,与分区方法、伙伴系统方法相比,本文方法的分配时间最短,内存利用率最高,进一步验证了本文方法的有效性。

表 7 3 种方法内存利用率和分配时间比较结果

Table 7 Comparison of memory utilization and allocation time between the three approaches

| 内存分配<br>次数/次 | 本文方法        |             | 分区方法        |             | 伙伴系统方法      |             |
|--------------|-------------|-------------|-------------|-------------|-------------|-------------|
|              | 分配时<br>间/ms | 内存利<br>用率/% | 分配时<br>间/ms | 内存利<br>用率/% | 分配时<br>间/ms | 内存利<br>用率/% |
| 100          | 1.35        | 99.35       | 2.08        | 98.69       | 1.97        | 98.22       |
| 500          | 6.07        | 99.26       | 9.11        | 97.22       | 10.22       | 95.34       |
| 1 000        | 11.51       | 98.93       | 13.26       | 95.34       | 15.75       | 91.68       |
| 3 000        | 34.98       | 98.52       | 39.17       | 93.25       | 41.26       | 88.29       |
| 5 000        | 62.26       | 98.31       | 71.06       | 91.22       | 70.55       | 83.37       |
| 7 000        | 91.28       | 97.22       | 102.66      | 88.37       | 98.37       | 81.22       |
| 10 000       | 123         | 95.13       | 135.79      | 81.25       | 139.36      | 76.37       |

4 结论

提出一种新的物联网实时任务传输中链路传感节点内存效率优化方法。介绍了 TLSF 方法及其弊端,在此基础上,通过 TC-TLSF 方法对物联网实时任务传输中链路传感节点内存效率进行优化处理。经实验验证,所提方法内存效率高。

参 考 文 献

1 钱晓军,范冬萍,吉根林. 物联网环境下实时任务传输的分簇调度算法. 计算机科学,2016;43(11):176—179  
Qian Xiaojun,Fan Dongping, Ji Genlin. Cluster scheduling algorithm for real time tasks transmission in internet of things. Computer Science,2016;43(11):176—179

2 郑杰,李建平. 物联网传感网络路由改进设计算法研究. 科技通报,2017;33(3):92—95  
Zheng Jie,Li Jianping. Research on improved design algorithm of internet of things sensor network. Bulletin of Science and Technology, 2017;33(3):92—95

3 Liu X,Hou K M,De V C, *et al.* Memory and energy optimization strategies for multithreaded operating system on the resource-constrained wireless sensor node. Sensors,2014;15(1):22—48

4 王凯,杨剑锋,郭成城,等. 共享内存并行编程最优同步方法的研究. 科学技术与工程,2015;15(8):99—102  
Wang Kai,Yang Jianfeng,Guo Chengcheng, *et al.* Research of the most optimal synchronization method in shared-memory parallel programming. Science Technology and Engineering,2015;15(8):99—102

5 穆元彬,史清江,徐伟强,等. 无线传感网中节点能量和链路容量约束的二阶分布式流控制方法. 电子学报,2015;43(5):1000—1006  
Mu Yuanbin, Shi Qingjiang, Xu Weiqiang. Node energy and link capacity constrained distributed flow control in wireless sensor networks; a second-order Method. Acta Electronica Sinica, 2015; 43(5): 1000—1006

6 薛建彬,王丹. 油气生产物联网系统的 TD-LTE 专网方案设计. 兰州理工大学学报,2016;42(5):106—111  
Xue Jianbin,Wang Dan. Design project of TD-LTE private network for oil and gas production IOT system. Journal of Lanzhou University of Technology,2016;42(5):106—111

7 杨浏. 一种面向监测区域的链路质量和覆盖保证的节点调度算法. 计算机测量与控制,2015;23(5):1624—1627  
Yang Liu. Sensor node scheduling algorithm for monitoring area based on coverage and link quality guarantee. Computer Measurement & Control, 2015;23(5):1624—1627

8 Shin M, Joe I. Energy management algorithm for solar-powered energy harvesting wireless sensor node for internet of things. Iet Communications, 2016;10(12):1508—1521

9 王甜甜,王颖. 物联网信息传输数据精确性优化仿真研究. 计算机仿真,2016;33(10):258—261  
Wang Tiantian,Wang Ying. Internet information transmission data accuracy optimization simulation research. Computer Simulation,2016; 33(10):258—261

10 徐善永,李豹,黄友锐,等. 物联网中链路稳定和能量感知混合模型的组播路由协议. 安徽理工大学学报,2016;36(1):19—24  
Xu Shanyong,Li Bao,Huang Yourui, *et al.* A multicast routing protocol based on the hybrid model of link stability and energy aware in internet of things. Journal of Anhui University of Science and Technology,2016;36(1):19—24

11 Ahmed A A. A comparative study of QoS performance for location based and corona based real-time routing protocol in mobile wireless sensor networks. Wireless Networks,2015;21(3):1015—1031

12 谢娜. 对 ARM 和 Linux 的物联网远程监控终端设计分析及实现. 电子设计工程,2017;25(1):163—165  
Xie Na. Design analysis and implementation of remote monitoring terminal for Internet of things in ARM and Linux. Electronic Design Engineering,2017;25(1):163—165

13 李超良,刘琴,谢永明,等. 基于同态加密的物联网隐私保护计算方案. 计算机工程与应用,2015;51(6):22—26  
Li Chaoliang,Liu Qin,Xie Yongming, *et al.* Privacy protection method in internet of things based on homomorphic encryption. Computer Engineering and Applications,2015;51(6):22—26

14 朱艺华,徐骥,田贤忠,等. 无线传感器网络应用简单 Reed-Solomon 编码的低能耗和低时延可靠数据收集方案. 计算机学报,2015;38(10):2106—2124  
Zhu Yihua,Xu Ji,Tian Xianzhong, *et al.* Energy-efficient and low-delay reliable data gathering scheme applying simple reed-solomon Code for Wireless Sensor Network. Chinese Journal of Computers, 2015;38(10):2106—2124

15 赵琳,孙国栋,陈志泊,等. 传感网中链路干扰优化的拓扑控制综述. 计算机工程与应用,2015;51(10):93—102  
Zhao Lin, Sun Guodong, Chen Zhibo. Link-interference-aware topology control in wireless sensor networks;a survey. Computer Engineering and Applications, 2015;51(10):93—102

16 丁维龙,韩燕波. 一种数据流处理环境下的节点副本放置方法. 电子与信息学报,2014;36(7):1755—1761  
Ding Weilong, Han Yanbo. A replica placement method during data stream processing. Journal of Electronics & Information Technology, 2014;36(7):1755—1761

17 马武彬,刘明星,邓苏,等. 面向物联网的语义空间资源索引构建及其查询优化算法. 系统工程与电子技术,2014;36(1):173—178  
Ma Wubin,Liu Mingxing,Deng Su, *et al.* Spatial resource index construction and query optimization algorithm for Internet of Things. Systems Engineering and Electronics,2014;36(1):173—178

# Memory Efficiency Optimization of Link Sensing Nodes in Real-time Task Transfer of Internet of Things

YAN Sheng-hua, WANG Chang-da \*  
( Graduate School, Jiangsu University, Zhenjiang 212016, China)

[ **Abstract** ] In the real-time task transmission of internet of things, different applications have different memory requirements for link sensing nodes, which tend to have more memory in some nodes and more memory in some nodes. Therefore, a new method for optimizing memory efficiency of link sensing nodes is proposed. The management mechanism of TLSF two level discrete table and the disadvantages of TLSF method were introduced. On this basis, the memory efficiency of link sensing node is optimized by using TC-TLSF method in the real-time task transfer of internet of things. The TC-TLSF method by the fuzzy control process of judging the release of memory if the sensor node link will immediately free memory block together, by the principle of maximum membership degree to the fuzzy characteristic parameters of free memory block belongs to grade judgment, calculate the coefficient of consolidation, comparing the coefficient and dynamic threshold to determine whether the delay of the merge operation. Fuzzy feature parameters and fuzzy rules are introduced to determine the dynamic threshold. Through memory allocation and memory release control, memory efficiency optimization is achieved. Experimental results show that the proposed method has high memory efficiency.

[ **Key words** ] Internet of things      real-time tasks      transmission      link sensing nodes      memory efficiency