

# 基于低能耗与高缓存命中并存的缓存替换算法<sup>①</sup>

胥琳, 林宇阳

(浙江工业大学 经贸管理学院, 杭州 310058)

**摘要:** 近年来无线传感器被广泛地利用在各个领域, 与之相关的优化节能研究也层出不穷. 作为信息共享、分发关键技术的缓存技术节能研究成为了研究热点之一. 从缓存替换算法的角度对缓存技术节能进行研究, 先对已有的缓存替换算法进行比较分析, 在继承二分法思想以及无线传感器网络中缓存替换策略的研究思想的基础上, 整合基于低能耗和高缓存命中中的两种替换算法, 构建出兼顾低能耗和高缓存命中双目标的缓存替换算法. 最后通过仿真验证该算法在平均延迟时间、能量消耗以及缓存命中三个方面均有不同程度的提升.

**关键词:** 无线网络; 缓存替换算法; 低能耗; 高缓存命中

引用格式: 胥琳, 林宇阳. 基于低能耗与高缓存命中并存的缓存替换算法. 计算机系统应用, 2017, 26(7): 189-194. <http://www.c-s-a.org.cn/1003-3254/5849.html>

## Cache Replacement Algorithm Based on Low Energy Consumption and High Cache Hit

XU Lin, LIN Yu-Yang

(College of Economics and Management, Zhejiang University of Technology, Hangzhou 310023, China)

**Abstract:** In recent years, wireless sensors are widely used in various fields, and the research about energy-efficiency strategies is emerging in an endless stream. As a key technology of information sharing and distribution, the energy-efficiency research about caching technologies has been one of research hot spots. Based on the cache replacement algorithm and the basic thought that adopts the replacement cost study, this research analyses the existing cache replacement algorithm firstly, then integrates the two cache replacement algorithms based on low energy consumption or high cache hit and refines one cache replacement algorithm that combines both advantages, which follows the on-demand cache placement policy based on dichotomy and the cache replacement strategy in wireless sensor network. At last, the simulation proves that, this algorithm has varied degrees of advancement in the following three aspects: the average delay time, energy consumption and cache hit.

**Key words:** wireless network; cache replacement algorithm; low energy consumption; high cache hit

无线传感器网络(wireless sensor network, WSN)的自组织、无中心、健壮性等特点使其广泛应用于军事、智能交通、环境监测、医疗卫生等领域. 由于传感器节点的电源能量有限性制约着传感器网络的生命周期, 节能研究一直是无线传感领域热门的方向之一.

在大数据融合的无线传感网络中, 多个节点往往会向同一个数据源节点请求相同的数据, 容易出现网

络拥堵、访问延迟等问题. 通过使用缓存技术, 设立缓存节点把高频访问的数据存储在缓存节点的缓存空间内, 能够避免用户在下次请求相同的信息时再次访问网络, 从而提高网络数据通讯性能. 缓存策略的优劣直接影响无线传感器网络的能量消耗和通信效率.

目前, 缓存策略研究主要包含缓存放置策略、缓存替换策略和缓存一致性策略. 缓存放置策略主要研

<sup>①</sup> 基金项目: 浙江省自然科学基金(LY13F020031)

收稿时间: 2016-10-27; 收到修改稿时间: 2016-12-05

究解决两个问题, 哪些节点作为缓存节点以及应该缓存怎么样的数据信息. 缓存替换策略主要研究如何替换缓存中数据项实现有限缓存空间的最优化利用. 缓存一致性策略主要研究如何以最低的代价保持源节点数据与缓存数据之间的一致性. 由于网络内节点间信息的高速传输使得缓存节点的缓存数据不断被替换, 缓存替换策略成为决定传感网络节点缓存性能优劣的重要因素, 因此本文从缓存替换策略角度进行研究.

## 1 无线传感网络缓存替换技术研究现状

目前业内的研究成果, 无线环境下的缓存替换算法主要围绕基于替换代价的思想展开, 其中具有代表性的有以下几种:

(1) 合作缓存策略(GCCS): 通过设定收益函数, 以访问频率和节点间距离为变量, 选择函数值最小项为缓存替换对象<sup>[1]</sup>. (2) 协同缓存策略: 构造一个代价函数, 引入数据项大小、延迟时间、访问频率以及弱一致检测机制, 选择函数值大的数据项进行替换<sup>[2]</sup>. (3) LIRS缓存替换策略: 以使用最近(IRR)及最近度(R)来记录区块历史信息, 后使用“栈剪枝”技术实现缓存的动态替换<sup>[3]</sup>. (4) 按需缓存替换策略(BESS): 将数据项、存在时间、访问率作为权值建立权值函数, 替换出函数值较大的数据项<sup>[4]</sup>. (5) SAIU替换策略: 主要考虑缓存内数据对象的访问率、更新率和占用空间大小等因素, 构建的权值函数<sup>[5]</sup>. (6) Min-SAUD策略: SAIU替换策略的一种改进, 主要创新点在于增加了缓存数据获取延迟验证的参数, 考虑到缓存有效检验延迟对替换的影响, 因此改进的获益函数的参数设定也更为科学, 性能上要优于SAIU<sup>[6]</sup>. (7) 基于访问频率的LA2U策略和LAUD策略: 在传统的LFU算法的基础上考虑到对象的访问频率和更新频率, 很适合无线传感器网络节点缓存的使用<sup>[7]</sup>. (8) OUR策略: 基于数据更新频率和数据访问率为基础, 给出了全新的一种缓存替换策略<sup>[8]</sup>.

综合以上的替换缓存算法我们不难发现, 其基本思路是在既定假设下, 合理区分组成缓存获益函数的影响因素, 构建替换算法模型. 目前国内外现有关于无线传感器网络缓存替换算法的研究中基本采用MANET网络中基于替换代价的算法思想, 即沿用性能相对较好、算法复杂度较低的替换代价算法, 只是权值函数的构造因素有所不同.

通过整理总结, 现有策略算法中主要的权值函数

组成因素有: 数据对象大小、数据对象的访问频率、数据对象存储时间、数据的请求延迟、数据对象的更新频率.

## 2 兼顾低能耗与高缓存命中的替换算法

文本依次分析追求低能耗的缓存替换算法的影响因素和追求高缓存命中的缓存替换算法的影响因素, 进而推导出兼顾低能耗与高缓存命中的替换算法.

### 2.1 能耗感知的替换策略

假设某一区域内共有 $N$ 个传感器节点 $S_1, S_2, \dots, S_N$ , 整个网络中有1个或多个sink节点, 1个或多个源节点. 讨论sink节点从某个缓存节点中获取数据项系统消耗的总能量, 如果sink节点从缓存节点中寻找某个数据项失败, sink节点直接从源节点中获取数据项.

定义某个缓存节点为 $S_k$ , 缓存总共有数据项 $M$ 个, sink请求数据项 $A$ 的概率为 $P_a(a=1 \dots M)$ .  $d_a$ 表示sink节点请求数据项 $A$ , 应答节点到sink节点的距离, 若应答节点为缓存节点 $S_k$ 时, 则 $d_a=d_{ak}$ . 应答节点为数据源节点, 则 $d_a=d_{as}$ .

文献[9]研究表明在工作状态下节点感知、处理、空闲和休眠的能耗恒定的情况下, 节点能耗主要来源于节点发送和接收数据的消耗<sup>[9]</sup>, 由此可以分析出, 降低节点发送和接收数据时候的能耗是能耗感知缓存替换算法的核心目的. 则我们做一下假设:

1. 使用 $E_{gd}$ , 表示节点感知、处理、空闲和休眠时的能耗;
2. 使用 $E_{sf}$ , 表示节点单位距离接收和发送数据项的能耗.

则我们可知对于某一缓存数据项 $A$ 消耗的能耗 $E_a$ 为:

$$E_a = E_{gd} + E_{sf} * P_a * d_a \quad (1)$$

假设某一缓存节点 $S_k$ 此时替换出数据项 $D$ , 新加入的数据项 $C$ . 数据项 $D$ 大小用 $Size_d$ 表示, 数据项 $C$ 大小用 $Size_c$ 表示, 为了使新加入的数据项 $C$ 能顺利的加入缓存中, 则数据项 $D$ 、 $C$ 必须满足以下要求:

$$Size_d \geq Size_c \quad (2)$$

该假设中我们先假定数据项大小相同, 则替换前后该节点的能耗差为:

$$E_{c-d} = P_c * d_c * E_{sf} - P_d * d_d * E_{sf} \quad (3)$$

$E_{c-d}$ 为能耗差,  $P_c$ 、 $P_d$ 表示sink节点请求数据项 $C$ 、

$D$ 的概率,  $d_c$ 、 $d_d$ 表示sink节点请求数据 $C$ 、 $D$ 时应答节点到sink节点的距离. 由于数据项 $C$ 与数据项 $D$ 的替换发生在一个缓存节点上, 根据 $d_a$ 的定义, 上式中 $d_c=d_d=d$ , 则:

$$E_{c-d} = d * (P_c - P_d) * E_{sf} \quad (4)$$

其中 $d$ 和 $E_{sf}$ 都是定值. 为了使替换前后缓存节点的耗能最优, 则 $E_{c-d}$ 的值要最大, 那么 $(P_c - P_d)$ 的值必须最大, 即替换入的数据项 $C$ 的访问率要大于替换出的数据项 $D$ 的访问率. 结合文献[9]研究成果, 考虑到数据项大小、数据项产生时间, 我们可以得到能耗角度的替换函数:

$$gain_1 = \frac{Size * Time}{P} \quad (5)$$

$Size$ 表示数据项的大小,  $Time$ 表示数据项占据缓存的时间,  $P$ 表示数据项的访问率, 当缓存溢出时, 节点计算各个数据项的 $gain_1$ 值, 替换出 $gain_1$ 值最大的数据项.

基于能耗感知的缓存替换函数(5), 从分析工作状态下的能量消耗角度出发, 有利于降低整个网络能耗. 但是也存在不足, 对于新存入缓存的数据项, 它的访问率 $P$ 的值很小, 按照其策略, 新存入的数据项可能很快就会被替换出缓存. 根据数据缓存的时间局部性规律<sup>[10,11]</sup>, 如果一个数据项被访问, 则可能很快被再次访问. 由此可知访问率 $P$ 无法全面的体现数据项的热度.

另一方面, 传输单个bit数据消耗的能量和处理上千条命令消耗的能量相同<sup>[12]</sup>, 可以运用相对完整的运算来全面的体现数据项的热度. 因此, 算法(5)存在缓存节点缓存命中不高的缺点, 必须引入高缓存命中率这一概念.

## 2.2 低能耗和高缓存命中共存的替换算法

本文简化整个高缓存命中的过程, 提取保持高缓存命中的两个函数权值, 即相关最近度 $U$ , 表示介于数据项最后一次和倒数第二次引用之间访问的其它数据项的数量, 和最近度 $I$ , 表示数据项从最后一次引用到当前访问的其它数据项的数量. 如表1所示, 表中虚拟时间是基于引用序列定义的, 其中一次引用代表一个基准时间单位, 符号“X”表示在一个虚拟时间单位内被访问. 如, 数据项 $A$ 在时间单位9时被访问. 表中已标明在时间单位10时各数据项的 $U$ 、 $I$ 值. 当某个数据项在一个时间序列内只被引用一次的时候, 将 $U$ 取值为缓存的数据项容量 $M$ 的两倍, 取 $2M$ 是一种相对无穷大得思想, 为了算法计算更加方便<sup>[3]</sup>.

表1 相关最近度和最近度的实例

数据项											I	U	
A									X			0	2M
B		X					X					2	3
C				X								4	2M
D				X	X							3	1
E		X					X	X				1	1
虚拟时间	1	2	3	4	5	6	7	8	9	10	当前时刻为10		

由上表可以看出 $I$ 的值越高, 说明块未被引用的时间跨度越久, 即块的“活跃度”越低;  $U$ 的值越高, 说明块在缓存内的活跃周期越长, “活跃度”越低. 因此, 我们可以通过 $U$ 和 $I$ 动态灵敏的监测缓存内每个块的“活跃度”, 并构建基于高命中的替换函数:

$$gain_2 = U * I \quad (6)$$

$U$ 表示相关最近度,  $I$ 表示最近度, 当缓存溢出时, 替换出 $gain_2$ 值最大的数据项.

综上, 根据函数(5), 从能耗的角度, 缓存需要替换出本身较大且占据缓存较长时间的数据项. 而从函数(6)可知, 缓存需要替换出“活跃度”较低的数据项. 由于函数(5)中的访问率 $P$ 无法全面的体现数据项的热度, 因此使用最近度 $I$ 与相关最近度 $U$ 乘积即函数(6)替换访问率 $P$ , 由此综合考虑能耗和缓存命中率两个方面, 得到了兼顾能耗和缓存命中率的缓存替换算法:

$$gain_3 = Size * Time * U * I \quad (7)$$

$Size$ 表示数据项的大小,  $Time$ 表示数据项占据缓存的时间,  $U$ 表示相关最近度,  $I$ 表示最近度, 当节点缓存溢出时, 计算每个数据项的 $gain_3$ 值, 并替换出 $gain_3$ 值最大的数据项.

替换算法(7), 简化了文献[3]高缓存命中算法中复杂的“栈剪枝”操作使用相关最近度 $U$ 和最近度 $I$ 两指标动态灵敏的反映缓存数据项的活跃度, 在吸取文献[9]能耗感知算法的优点的基础上, 实现算法的多目标优化, 构建了兼顾低能耗和高缓存命中的缓存替换算法. 下一节将对该策略进行仿真测试.

## 3 仿真试验

本文使用NS2工具2.35版本对模型进行仿真实验. 实例检验上一节所提替换算法模型的实际优化效果.

### 3.1 仿真试验环境设置

根据前文论述, 网络中需数据源节点、sink节点和中间节点三种类型的节点. 数据源节点只采集数据信

息,不转发来自邻居节点的数据信息,sink节点请求感兴趣的数据信息,中间节点转发数据信息.在缓存发现方法上,采用二分法的思想在定向路由扩散协议下建立的sink节点到数据源节点的路径在寻找缓存节点,使选择的缓存节点分散在网络中,并用跳数来衡量节点间的距离.本文的仿真试验环境如图1所示,30个节点随机分布在某一区域中自发组网,并在该区域内随机选择源节点和sink节点,图中比较分散的蓝色的圈表示sink节点发出的兴趣请求信息的传播路径,集中的红色圈表示源节点采集到数据信息并返回给sink节点的路径.

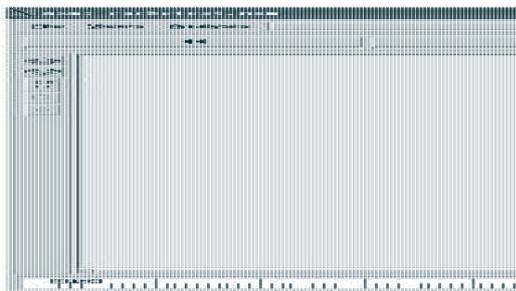


图1 仿真环境下的网络节点分布图

### 3.2 仿真试验参数设置

仿真过程采用的是定向扩散路由协议,此外,sink节点请求数据的速率不能大于数据源节点感知数据的速率,因此在本文的仿真试验中,设置sink节点平均每0.5 s发送一次兴趣请求,数据源节点平均每1 s感知1个数据信息,并且计算得到了仿真中需要设置的能量阈值为105.5 J.本文的仿真试验参数设置如表2所示<sup>[13]</sup>.

### 3.3 仿真实验结果及分析

根据前面分析,为了得到新算法相较于其它现存算法对网络系统的性能改善效果,本文的仿真试验选择了平均延迟时间、平均能量消耗和缓存命中率这三个仿真目标,将本文的核心算法分别同BESS和LIRS算法做比较,具体实现如下.

#### 3.3.1 平均延迟时间

本文设置仿真环境为:1个源节点,分别对应1、2、3、4、5个sink节点,那么平均延迟时间=网络总延迟时间/数据项的总个数,其中网络总延迟时间等于各sink节点的延迟时间之和.具体结果如图2显示,平均时延随仿真网络中节点数量变化而变化的情况,本文算

法相较于BESS算法,平均时延降低15.07%~23.18%;相较于LIRS算法,平均时延降低11.01%~18.46%.仿真设置各算法选择的缓存节点在sink节点5跳范围内,随着sink节点数量的不断增加,网络中的信息传输密度更高,导致节点间数据传输通路堵塞,从而增加信息的反馈时间,进而增加sink节点获取数据的时间不断提高.本文算法中的缓存放置策略采用BESS算法中基于二分法思想的按需缓存放置策略,所选择的缓存节点可以分散在网络中,这样的好处在于sink节点在获取缓存数据时有不同路径的多种选择,从而一定程度上缓解网络堵塞的情况.

表2 仿真试验参数设置

参数	设定值
环境/m <sup>2</sup>	800*800
节点个数/个	30~80
相邻节点延迟/ms	15
缓存大小/KB	100~300
数据项有效时间/s	30
仿真时间/s	300
带宽/Mbps	2
每个对象大小/KB	1
缓存内对象的个数/个	100~300
数据库中对象个数/个	500
每个节点的初始能量/J	1000
节点发送能量/J	0.66
节点接收消耗的能量/J	0.395
节点处于空闲和休眠消耗的能量/J	0.035

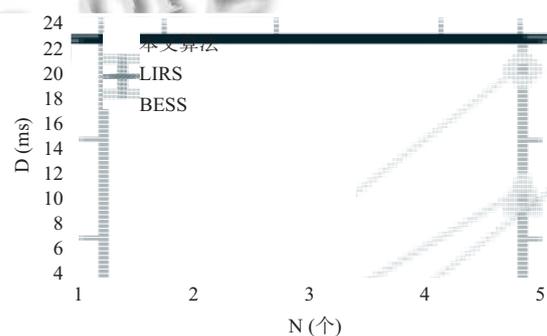


图2 平均时延随sink节点数量的变化

#### 3.3.2 平均能量消耗

本文设置仿真环境为:2个sink节点,2个源节点,整个网络节点数量为30到80个之间.平均消耗能量=网络总消耗能量/节点总个数;网络总消耗能量=网络初始能量-仿真结束时网络的剩余能量.图3显示节点的平

均能耗随网络密度变化而变化的情况,相较于BESS算法,本文算法节点的平均能耗降低4.71%~8.95%;相较于LIRS算法,本文算法节点的平均能耗降低20.31%~21.36%。LIRS算法采用传统的缓存发现策略,选择的缓存节点集中在sink节点附近,虽能缩短sink节点缓存发现的距离,但同质缓存节点的密集容易导致网络能量空洞,影响网络生命周期的同时,也造成维持空闲缓存节点的能量浪费。本文算法采用的缓存发现策略,能够将缓存节点较为合理的分散到整个网络中,避免LIRS算法缓存发现的弊端,同时去除了LIRS算法为动态维持LIR集和HIR集而进行复杂的“栈剪枝”操作,因此节点的平均能耗较LIRS算法有较大幅度的减少。

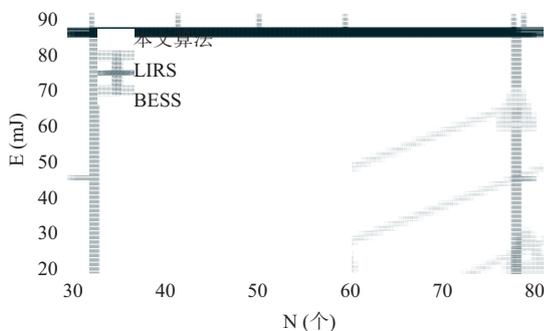


图3 平均能耗随网络节点数量的变化

### 3.3.3 缓存命中率

本文设置仿真环境为:1个sink节点,2个源节点,网络中节点个数恒定为30个,缓存节点的缓存大小为100-300 KB。那么缓存命中率=sink节点收到来自缓存节点的信息数量/sink节点发出的请求数量。图4显示sink节点的缓存命中率随缓存节点缓存空间大小变化而变化的情况。相较于BESS算法,本文算法的缓存命中率提高5.33%~12.70%;相较于LIRS算法,本文算法的缓存命中率提高1.71%~4.94%。缓存空间越大,意味着能够缓存更多的数据对象,进而缓存命中的概率越高。相较于LIRS算法,本文算法不仅考虑了最近度和相关最近度,还综合考虑了数据对象的大小和缓存一致性有效下的缓存时间,仿真结果表明本文算法的缓存命中率有小幅度提升;相对于BESS算法,本文算法使用最近度和相关最近度这两个能更好刻画缓存对象“热度”的因素来取代BESS算法中的数据对象访问率和更新频率,仿真结果表明本文算法的缓存命中率达到较大幅度的提升。

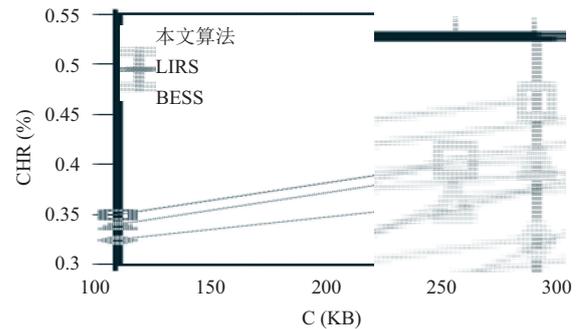


图4 缓存命中率随缓存空间大小的变化

## 3 结语

本文主要研究无线传感器网络缓存技术中的缓存替换策略。在总结当前研究现状的基础上,以能耗感知缓存替换算法为切入点,针对其中的不足,引入高缓存命中思想,使用最近度 $I$ 与相关最近度 $U$ 这两个更能刻画缓存数据“热度”的函数因素来取代能耗感知算法中同样刻画缓存数据“热度”的访问率 $P$ ,同时移除了复杂的“栈剪枝”操作<sup>[14]</sup>,得到全新的追求低能耗和高缓存命中共存的缓存替换算法。仿真实验表明新算法在平均延迟时间、平均能量消耗和缓存命中率三个方面均有所提高。

本文所研究的缓存替换策略的网络节点功耗管理模型为动态功耗管理(DMP)策略下的电池理想模型。但是现实中电池的最大寿命不完全与节点最小能耗对等,实际电池释放能量有着非线性的特点,可以继续研究电池模型对于缓存节点的选择和能量阈值设定的影响。另一方面,本文采用的是替换算法中基于二分法思想的能量按需缓存放置策略研究成果<sup>[15]</sup>,因此如何选择更加动态适合网络传输情况的缓存放置策略是今后需要进一步研究的课题。

## 参考文献

- 1 Chauhan N, Awasthi LK, Chand N. Global cooperative caching for wireless sensor networks. 2011 World Congress on Information and Communication Technologies. Mumbai, India. 2011. 235-239.
- 2 Dimokas N, Katsaros D, Tassiulas L, et al. High performance, low complexity cooperative caching for wireless sensor networks. *Wireless Networks*, 2011, 17(3): 717-737. [doi: 10.1007/s11276-010-0311-x]
- 3 Jiang S, Zhang XD. LIRS: An efficient low inter-reference recency set replacement policy to improve buffer cache

- performance. Proc. of the 2002 ACM SIGMETRICS International Conference on Measurements and Modeling of Computer Systems. Marina Del Rey, California, USA. 2002. 31–42.
- 4 李欠欠. 无线传感器网络中基于能量有效的按需缓存策略研究[硕士学位论文]. 杭州: 中国计量学院, 2014.
  - 5 Xu JL, Hu QL, Lee DL, *et al.* SAIU: An efficient cache replacement policy for wireless on-demand broadcasts. Proc. the 9th International Conference on Information and Knowledge Management. McLean, VA, USA. 2000. 46–53.
  - 6 Xu JL, Hu QL, Lee WC, *et al.* Performance evaluation of an optimal cache replacement policy for wireless data dissemination. IEEE Trans. on Knowledge and Data Engineering, 2004, 16(1): 125–139. [doi: [10.1109/TKDE.2004.1264827](https://doi.org/10.1109/TKDE.2004.1264827)]
  - 7 Chen H, Xiao Y, Shen XM. Update-based cache access and replacement in wireless data access. IEEE Trans. on Mobile Computing, 2006, 5(12): 1734–1748. [doi: [10.1109/TMC.2006.188](https://doi.org/10.1109/TMC.2006.188)]
  - 8 Akon M, Islam MT, Shen XM, *et al.* OUR: Optimal update-based replacement policy for cache in wireless data access networks with optimal effective hits and bandwidth requirements. Wireless Communication and Mobile Computing, 2013, 13(15): 1337–1352.
  - 9 程莉, 刘建毅, 王枫. 计算机网络. 北京: 科学出版社, 2012.
  - 10 赵国栋, 顾峰. 对Cache命中率优化的探讨. 宁夏师范学院学报, 2007, 28(6): 54–57.
  - 11 席红旗. 计算机高速缓冲存储器(Cache)命中率的分析. 河南教育学院学报(自然科学版), 2012, 21(3): 31–32.
  - 12 Min R, Bhardwaj M, Cho SH, *et al.* Low-power wireless sensor networks. Proc. 14th International Conference on VLSI Design. Bangalore, India. 2001. 205–210.
  - 13 柯志亨, 程荣祥, 邓德隽. NS2仿真实验——多媒体和无线网络通信. 北京: 电子工业出版社, 2009.
  - 14 Pant S, Chauhan N, Kumar P. Effective cache based policies in wireless sensor networks: A survey. International Journal of Computer Applications, 2010, 11(10): 17–21. [doi: [10.5120/ijca](https://doi.org/10.5120/ijca)]
  - 15 Srivastava JR, Sudarshan TSB. Energy-efficient cache node placement using genetic algorithm in wireless sensor networks. Soft Computing, 2015, 19(11): 3145–3158. [doi: [10.1007/s00500-014-1473-8](https://doi.org/10.1007/s00500-014-1473-8)]