

一种面向线性相关冗余优化的源节点选择算法^①

李 珊^{1,2}, 袁 远³, 胡一鹏⁴

¹(湖南大学 计算机与通信学院, 长沙 410082)

²(长沙师范学校 电子信息工程系, 长沙 410100)

³(国防科技大学 计算机学院, 长沙 410073)

⁴(总参, 北京 100081)

摘 要: 基于网络编码的 P2P 流媒体直播系统的优势之一在于多个源节点之间不需要显式的协同调度也能有效地服务于请求节点。但正是由于缺乏协同, 即使编码系数的有限域足够大, 仍然存在线性相关冗余数据, 从而浪费了源节点的带宽。分析了这一问题产生的原因, 并提出采用从 tracker 提供的源节点集合中选择部分节点作为活动源节点来解决该问题。活动源节点最优选择问题可以归约为 0-1 背包问题的变种, 是 NP 难的, 因此我们设计了一个多项式时间的近似算法来逼近最优解。通过形式化证明和模拟, 我们验证了该算法的可行性。数据表明该方法能够进一步提高 P2P 流媒体直播系统的服务质量。

关键词: 网络编码; P2P 直播流媒体; 活动源节点; 背包问题

A Source Peer Selection Algorithm for Linear Dependent Redundancy Optimization

LI Shan^{1,2}, YUAN Yuan³, HU Yi-Peng⁴

¹(School of Computer and Communication, Hunan University, Changsha 410082, China)

²(Department of Electronic and Information Engineering, Changsha Normal College, Changsha 410100, China)

³(School of Computer, National University of Defense Technology, Changsha 410073, China)

⁴(GSD, Beijing 100081, China)

Abstract: One advantage of the network coding based P2P live streaming system is that multiple source peers can serve the requests efficiently without explicit cooperation control. However, the lack of cooperation also brings redundant data due to the linear dependence, even when the Galois field of the coding coefficients is large enough. In this paper, we first analyze the causes of the redundant data. Then, we propose selecting active source peers from the source set given by the tracker to handle this problem. Active source peer selection problem can be regarded as a variant of 0-1 knapsack problem, which is NP hard, such that we design an approximation algorithm to compute the solution. Through formal proofs and simulations, we verify the validity of the algorithm, which can further improve the QoS of the P2P live streaming.

Keywords: network coding; P2P live streaming; active source peer selection; knapsack problem

1 引言

网络编码作为一种向传统存储转发路由方式发起革新的包交换技术, 自 2000 年由香港中文大学的 Ahlswede 等^[1]提出以来, 得到了全球学者的广泛关注。

近年来, 网络编码在充分理论研究的基础上, 逐渐被应用到了 P2P 系统、无线网络和网络安全中, 取得了丰硕的成果。

基于网络编码的 P2P 流媒体系统不同于传统的

① 基金项目: 国家高科技研究发展计划(863)(2009AA012142)

收稿时间: 2010-07-13; 收到修改稿时间: 2010-08-13

P2P 流媒体系统: 传统系统中一般采用拉或推拉结合协议, 媒体文件先被划分成数据段(segment), 每段数据需要由请求节点显式的分配给一个源节点来提供; 而结合网络编码后, 数据段被进一步分解成数据块(block), 段包含数据块的数目被称作“段划分”。而网络编码通常只作用在一个数据段内的所有块上。P2P 系统中通常采用随机线性网络编码(random linear network coding)^[2], 即独立随机选取编码系数对多个数据块求线性组合生成编码块。对于每个数据段, 多个源节点同时向一个请求节点推送该段的编码块。请求节点收集到等于段划分数量的线性无关的编码块后能够通过高斯消去法来解码得到有效的媒体数据, 并发送自己的缓冲位图(buffer map)作为“刹车”消息通知源节点推送下一个数据段的编码块。

结合随机线性网络编码技术的 P2P 流媒体系统是目前国内外学术界和工业界研究的热点之一。特别是加拿大多伦多大学的 Li Baochun 教授从 2005 年以来一直致力于该方向的研究。他们首先提出的随机推协议 R2^[3], 随后实现了原型系统 Lava^[4]来验证新协议对 P2P 系统流服务质量的提高。同时, 文献[5,6]也通过数学建模对基于网络编码的 P2P 流媒体系统的性能进行了理论分析, 证明了采用网络编码的有效性。近期他们还将网络编码模块融入到成熟的商业流媒体系统 UUsee 中^[7], 通过实测数据进一步分析网络编码的优缺点。

然而这些相关工作大部分都集中在宏观协议的设计和系统的实现上, 关注的是基于网络编码的 P2P 数据传输不需要显式协同调度, 简化了系统的设计的这一优点, 却没有考虑到系统缺乏对多个源节点的协同调度和有效的资源分配方法, 即使是编码系数取值的有限域足够大, 也会造成大量的线性相关冗余数据, 浪费了部分带宽。本文针对该问题, 提出了一种活动源节点选择方法, 即请求节点从 tracker 获取到源节点信息后不一定需要与其中所有的源节点建立连接, 而只需要选取最合适的部分活动源节点以达到减少线性相关冗余并保证媒体播放质量的目的。活动源节点最优选择问题可以被看成 0-1 背包问题的变种, 是 NP 难的, 因此, 随后本文给出了一个多项式时间的算法来求其近似解。同时通过模拟实验和形式化分析验证了我们活动源节点选择算法的有效性。

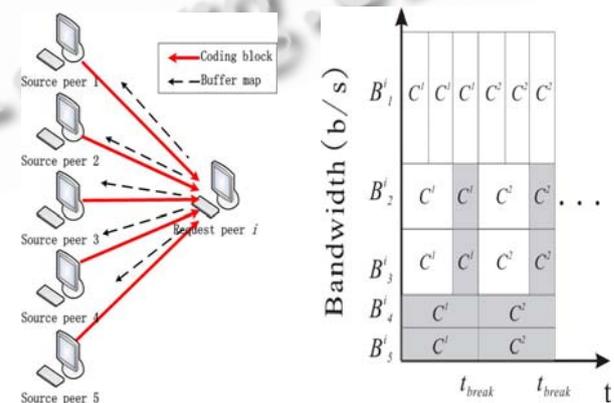
文章的组织如下: 第二节分析造成线性相关冗余数据原因; 第三节提出活动源节点最优选择问题; 第四节给出该问题的近似算法; 第五节对该算法进行性能评价, 结语在第六节。

2 线性相关冗余数据成因分析

在基于网络编码的 P2P 流媒体系统中, 当请求节点新收到的编码块的系数向量与其缓冲中已有的该段数据编码块的系数向量线性相关时, 根据高斯消去法, 这样的编码块不能对解码产生实际效用, 因此称这样的编码块为线性相关冗余数据。

当随机线性网络编码的系数取值有限域足够大时, 由随机选取系数所产生的线性相关冗余数据可以忽略^[2]; 但是由于采用网络编码的 P2P 流媒体系统中数据传输采用推协议^[3]并且缺乏有效的资源分配和协同调度方法同样会导致线性相关冗余。我们用个简单的例子来说明这种情况。为了简化讨论, 我们借鉴文献[8]中系统模型的部分设定: 假设源节点上行带宽和播放速率都是数据块的整数倍, 其单位可表示为(其中 b 表示 block, 即编码块数), 同时忽略传输延迟。

例一. 5 个源节点为请求节点 i 提供媒体数据服务, 他们的上行带宽 $B_1^i = 4b/s$, $B_2^i = B_3^i = 2b/s$, $B_4^i = B_5^i = 1b/s$, 而段划分 m 为 $5b$, C^j 表示数据段 i 的编码块。



(a) 多源服务

(b) 数据传输图

图 1 例示意图

如图 1(a)所示, 请求节点以缓冲位图作为控制消息来通知源节点改变推送的数据段。从图 1(b)可以看出, 当 5 个源节点一起推完个完整的数据段的编码块

(白色部分)后则会改为推送数据段 2, 这样每个数据段将周期性出现 b 的线性相关冗余数据(灰色部分)。由于这个周期也是请求节点发送“刹车”消息时间间隔, 因此我们用 t_{break} 表示。可以发现源节点 4 和 5 的上行带宽始终都处于浪费当中。而如果我们只选取 1 和 4 这两个源节点, 则没有线性相关冗余。这反映了当多个源节点带宽资源与系统中的段划分匹配不好则会导致大量的线性相关冗余。

因此我们可以控制请求节点从源节点集合中有选取合适源节点来建立连接, 达到减少冗余并节约带宽资源的目的。然而, 我们同时还需要满足用户对流媒体服务质量的要求, 即保证启动延迟较小和播放连续。基于网络编码的 P2P 流媒体不同于传统的 P2P 流媒体, 请求节点所获取的某一数据段的线性无关编码块数目在没有超过段划分之前都为不能播放的无效数据。一旦达到则整个数据段都将有效, 能够被播放。因此第一次发送“刹车”消息间隔 t_{break} 必须要小于等于用户容忍的最大启动延迟。同时还应该保证选出的节点的上行带宽之和要大于播放速率, 否则缓冲会有清空的危险。在例一中, 如果用户的最大缓冲容忍延迟为 0.8s, 播放速率为 5b/s, 则选择节点 1、2 和 3 来传输是最优的, 虽然会出现 1b 的冗余。

3 活动源节点最优选择问题

根据上节分析, 活动源节点选择问题可以定义如下:

已知: 请求节点的 N 个源节点的上行带宽集合为 $\{B_1^i, B_2^i, \dots, B_N^i\}$, 播放速率为 R , 用户所能够容忍的最大启动延迟为 d_{max} , 系统段划分为 m 。

目标: 求 N 个变量 $I_1^i, I_2^i, \dots, I_N^i$ ($I_j^i=1$ 表示请求节点选择该源节点为其服务, $I_j^i=0$ 表示不选择该源节点为其服务)的取值使得其能够最小化每个数据段的线性相关冗余数据量 L 同时满足用户的启动延迟要求和播放连续要求。

该问题可以表述成为一个 0-1 整数规划问题:

$$\text{minimize: } L = t_{break} \cdot \sum_{j=1}^N I_j^i B_j^i - m \quad (1)$$

$$\text{s.t. } t_{break} = \min\{t \mid \sum_{j=1}^N [I_j^i B_j^i \cdot t] \geq m\}; \quad (2)$$

$$t_{break} \leq d_{max}; \quad (3)$$

$$\sum_{j=1}^N I_j^i B_j^i \geq m / t_{break} \geq R; \quad (4)$$

$$I_j^i \in \{0,1\} (\forall j \ 1 \leq j \leq N). \quad (5)$$

其中公式(2)表示请求节点收到了 m 个编码块则会立即发“刹车”消息; 公式(3)是启动延迟约束; 公式(4)是连续播放要求: 由公式(2)可以知 $\sum_{j=1}^N I_j^i B_j^i \geq m / t_{break}$ 一定成立, 因此公式(3)和(4)可以合并成为一个约束:

$$t_{break} \leq \min(d_{max}, m / R); \quad (6)$$

定理 1. 活动源节点最优选择问题是 NP 难的。

证明: 活动源节点最优选择问题能够归约到一个背包问题 (knapsack problem): 先去掉约束公式(6), 即将 d_{max} 和 R 分别设置为无穷大和 0, 再将带宽 B_j^i 既看作每件物品的重量又看作该物品的价值, 背包大小设置为 m , 而目标函数设置成: maximize: $-L = m - t_{break} \cdot \sum_{j=1}^N I_j^i B_j^i$, 则整个问题被转换成了一个背包问题, 因此是 NP 难的。

虽然活动源节点最优选择问题可以归约到背包问题, 但是却是一个变种, 与常规的背包问题不同之处在于:

① 活动源节点最优选择问题需要找到一个允许超过背包负载最少的解, 而原背包问题则是希望找到一个不能超过背包负载价值最大的解。

② 活动源节点最优选择问题中要么不选取该物品, 要么选取该物品的份量为 t_{break} , 即所有被选出的物品都具有同样的份量 t_{break} , 同时这个份量为小于等于 $\min(d_{max}, m / R)$ 的实数。

因此需要进一步讨论能否修改已有背包问题算法来解决活动源节点最优选择问题。

4 近似算法

首先我们需要考虑的是, 在选出活动源节点后, 如何计算由它们产生的线性相关冗余数据量, 以作为判断活动源节点选择好坏的标准。不难观察到请求节点发送“刹车”消息, 一定是在至少有一个活动源节点刚好完成某个编码块的传输时。据此, 我们设计了算法 Redundant_cal(A, m)来计算冗余, 其中 A 为记录活动源节点上行带宽的数组, m 为段划分, 则有以下冗余计算算法:

 Redundant_cal(A, m)

```

1.  $t_{min} = m / \sum A[j]$ ;  $L = \infty$ ;
2. for  $j = 1$  to  $|A|$  do
3.    $t_{break} = \lceil t_{min} \cdot A[j] \rceil / A[j]$ ;
4.   if  $\sum \lfloor A[i] \cdot t_{break} \rfloor == m$  then
5.      $L = t_{break} \cdot \sum A[j] - m$ ;
6.     break;
7.   end if
8. end for
9. return  $L, t_{break}$ ;

```

图 2 冗余计算算法

从图 2 易知, 该算法的时间开销为。

接下来就是考虑如何找出活动源节点集合, 使它们的带宽资源能够大致与段划分以及播放要求形成较好的匹配。由于关键参数的大小未知, 因此不能通过将每个节点的带宽乘以然后取整后作为物品的重量来借用 0-1 背包问题的现有算法。更合理的方法是调整的背包容量并不改变物品的重量, 然后通过多次迭代来找出近似最优解。由于该问题中物品重量为带宽, 单位为, 因此背包容量也应该设置为, 而不能直接用 m 表示。根据公式(6), 我们能够计算出背包的最小容量应该为, 可以将其作为迭代的起点, 然后设置一个最大迭代次数来求解。用 Knapsack(T, B, C) 表示 0-1 背包问题容量为 C 时的动态规划算法, 其中表示动态规划过程中基于容量为的背包和物品所能够得到最大价值。 B 表示记录所有源节点带宽的数组。我们还用表示在第次迭代中是否选取节点作为活动源节点, 和分别记录第次迭代中的冗余量和“刹车”间隔。以下为活动源节点选择算法伪代码:

 Active_selection(I, B, m, R, d_{max}, MAX)

```

1. Initialize  $I[i, k] = 0$ ;  $T[i, 0] = 0$ ;  $T[0, c] = 0$ ;  $L[k] = 0$ ;
    $t[k] = 0$ ;  $idx = 0$ ;
2.  $C_{min} = \lceil m / \min(d_{max}, m / R) \rceil$ ;

```

```

3. Knapsack( $T, B, C_{min} + MAX$ );
4. for  $k = C_{min}$  to  $C_{min} + MAX$  do
5.    $j = k$ ;
6.   for  $i = 1$  to  $N$  do
7.     if  $T[i, j] > T[i-1, j]$  then
8.        $I[i, k] \leftarrow 1$ ;
9.        $j \leftarrow j - B[i]$ ;
10.    end if
11.  end for
12.  Use  $I[i, k]$  and  $B$  to derive the active source set  $A$ 
13.  ( $L[k], t[k] = \text{Redundant\_cal}(A, m)$ );
14.  if  $L[k] = 0$  then
15.     $idx = k$ ;
16.    break;
17.  end if
18. end for
19. if  $idx = 0$  then
20.    $idx = \min(L)$ ; //Find the subscribe of the minimum
   value in array  $L$ .
21. end if
22. return  $I[:, idx], L[idx], t[idx]$ ;

```

图 3 活动源节点最优选择算法

该算法的时间开销为 $O(N \cdot (C_{min} + MAX))$, 因此整个算法的时间开销为问题规模的多项式, 能较高效的完成活动源节点的选择。

5 性能评价

这一节, 我们先用一个案例对该活动源节点最优选择算法进行评测, 然后将该算法加入我们自己用 C++ 实现的 P2P 流媒体模拟器中, 与 R2[3] 的随机推策略以及所有 tracker 提供的源节点都服务同一数据段的 Nave 策略进行比较。选择的性能指标有线性相关冗余数据量、启动延迟。

我们根据现实中流媒体系统实际情况来设计案例: 媒体文件大小为 200MB, 码率为 100KBps。数据块大小设为 1KB, 则一共有 200000 个数据块。请求节点加入系统时, tracker 为请求节点提供了 20 个源

节点，不同源节点为该请求节点提供的上行带宽服从 20KBps 到 80KBps 的均匀分布。用均匀分布随机数生成器，我们产生的 20 个节点带宽如下：

31 60 40 74 27 79 52 62 80 37
45 48 66 69 26 31 42 23 51 40

我们先固定启动延迟为 1.5s，调整段划分以 50b 为间隔从 100b 增加到 300b，则有活动源节点选择结果及其相应的线性相关冗余和启动延迟如表 1 所示：

表 1 不同段划分下算法结果

Segment size(b)	Selected bandwidth of active peers (b/s)	Redundant(b/s)	Initial delay(s)
100	60, 40	0	1
150	60, 40	0	1.5
200	31, 40, 80	0.075	1.325
250	40, 62, 80	0.25	1.375
300	60, 40, 52, 42	0.4231	1.4038

接下来我们固定段划分为 200b，调整最大可容忍启动延迟从 0.5s 增加到 2s，其相应的结果如表 2 所示：

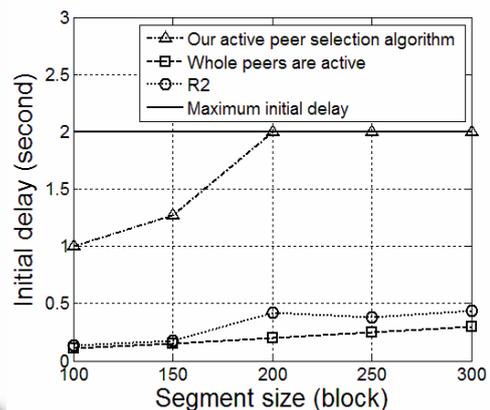
表 2 不同启动延迟下算法结果

Maximum Initial delay (b)	Selected bandwidth of active peers (b/s)	Redundant (b/s)	Initial delay (s)
0.5	31,60,74,27,79,62,80	1.3375	0.4875
1.0	31,27,62,80	0	1
1.5	31, 40, 80	0.075	1.325
2.0	60, 40	0	2
2.5	60, 40	0	2

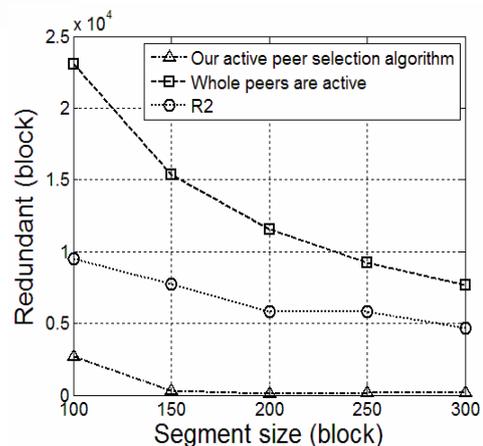
从上面两个表可以看出，我们的算法能够有效的优化每段内的线性相关冗余，并且保证实际启动延迟不高于用户的最大容忍启动限度。

为了进一步说明我们算法的可行性及优点，我们将该活动源节点算法、R2 策略以及 Nave 策略都在自己用 C++ 写的 P2P 流媒体模拟器中进行了实现。在 R2 策略中，源节点会随机的选取数据段来为请求

节点服务，而在 Na?ve 策略中，所有的源节点都将同时为请求节点提供同一数据段。模拟实验中，系统规模为 10000 个节点，其它设定同案例。如图 4(a) 所示，随着段划分的增大，我们的算法启动延迟会增加，虽然高于其他两个算法，但是却始终保证在用户的容忍范围内。这是由于我们的算法能够精确的选取满足服务要求的源节点来建立连接，以启动延迟为代价减少带宽资源的浪费。图 4(b) 表示的是案例中的该部媒体节目全部播放完成后平均每个节点的冗余数据量的统计。很明显，我们算法的线性相关冗余量要远小于其他两个算法，这样使得系统中的带宽资源在保证用户服务质量的同时，能够为更多的请求节点提供数据，有利于提高系统的可扩展性。



(a) 平均启动延迟



(b) 线性相关冗余

图 4 不同段划分下的算法比

6 结语

基于网络编码的 P2P 流媒体直播系统是近年来网络流媒体研究的新趋势。在这篇文章中,我们针对其中由于缺乏合理的资源分配策略而导致多源服务产生大量线性相关冗余数据这一个问题,提出了一个活动源节点最优选择问题,并给出了一个多项式时间近似算法。模拟实验表明该算法较 R2 等其他策略具有更少的线性相关冗余同时满足服务要求。我们下一步将在这个算法基础研究出合理的协同调度策略,用较少的控制开销更好的发挥网络编码在 P2P 系统中的作用。

参考文献

- 1 Ahlswede R, Cai N, Li SY, Yeung RW. Network information flow. *IEEE Transactions on Information Theory*, 2000,46(4): 1204—1216.
- 2 Ho T, Koetter R, Medard M, Karger D, Effros M. The benefits of coding over routing in a randomized setting. *Proc. of the IEEE International Symposium on Information Theory*. 2003: 442.
- 3 Wang M, Li B. R2: Random push with random network coding in live peer-to-peer streaming. *Journal on Selected Areas in Communications*, 2007,25(9):1655—1666.
- 4 Wang M, Li B. Lava: A reality check of network coding in peer-to-peer live streaming. *Proc. of IEEE International Conference on Computer Communications (INFOCOM)*. 2007:1082—1090.
- 5 Feng C, Li B. On large-scale peer-to-peer streaming systems with network coding. *Proc. of ACM Multimedia*. 2008:267—278.
- 6 Niu D, Li B. On the resilience-complexity tradeoff of network coding in dynamic p2p networks. *Proc. of IEEE International Workshop on Quality of Service (IWQoS)*, 2007:38—46.
- 7 Liu Z, Wu C, Li B, Zhao S. UUsee: Large-scale operational on-demand streaming with random network coding. *Proc. of IEEE International Conference on Computer Communications (INFOCOM)*. 2010.
- 8 Xu D, Hefeeda M, Hambrusch S, Bhargava B. On peer-to-peer media streaming. *Proceedings of International Conference on Distributed Computing Systems (ICDCS'02)*. Wien, Austria, Jul. 2002:343—355.
- 9 Chen HY, Sezaki K, Deng P, So HC. An improved DV-Hop localization algorithm for wireless sensor networks *Industrial Electronics and Applications*, 2008. ICIEA 2008. 3rd IEEE Conference on 3-5 June 2008:1557—1561.
- 10 Rudafshani, Datta S. Localization in Wireless Sensor Networks, *IPSN*, April 2007.
- 11 李道远,王义发,黎洪生.基于改进 DV-Hop 算法的无线传感器网络节点定位. *武汉理工大学学报(信息与管理工程版)*,2007,29(4):50—52.
- 12 刘林,范平志.一种降低定位误差的无线传感器网络节点定位改进算法. *电路与系统学报*,2007,12(2):50—55.
- 13 姜山,李建波.一种改进的 DV-HOP 传感器网络定位算法. *计算机工程与应用*,2007,43(34):141—143.
- 14 刘艳文,王福豹,段渭军,于超.基于 DV-Hop 定位算法和 RSSI 测距技术的定位系统. *计算机应用*,2007,27(3):516—518.
- 15 赵红,孙光,秦姣华,唐飞岳.夹角修正的 dv-hop 传感器网络节点定位研究. *计算机工程与应用*,2009,45(13):141—143.100—102.
- 16 Liu Y, Qian ZH, Liu D, Zhong H. A DV-Hop Positioning Algorithm for Wireless Sensor Network Based on Detection Probability, *ncm, 2009 Fifth International Joint Conference on INC, IMS and IDC*,2009.453-456.
- 17 张佳,吴延海,石峰,耿方.基于 DV-HOP 的无线传感器网络定位算法. *计算机应用*,2010,30(2):323—326.

(上接第 53 页)

- 7 Niculescu D, Nath B. DV based positioning in ad hoc networks. *Journal of Telecommunication Systems*, 2003, 22(1/4):267—280.
- 8 Shang Y, Ruml W, Zhang Y, Fromherz MPJ. Localization from mere connectivity. *Proc. of the 4th ACM Int'l Symp. On Mobile Ad Hoc Networking & Computing*. Annapolis: ACM Press, 2003.201—212.
- 9 Chen HY, Sezaki K, Deng P, So HC. An improved DV-Hop localization algorithm for wireless sensor networks *Industrial Electronics and Applications*, 2008. ICIEA 2008. 3rd IEEE Conference on 3-5 June 2008:1557—1561.
- 10 Rudafshani, Datta S. Localization in Wireless Sensor Networks, *IPSN*, April 2007.
- 11 李道远,王义发,黎洪生.基于改进 DV-Hop 算法的无线传感器网络节点定位. *武汉理工大学学报(信息与管理工程版)*,2007,29(4):50—52.
- 12 刘林,范平志.一种降低定位误差的无线传感器网络节点定位改进算法. *电路与系统学报*,2007,12(2):50—55.
- 13 姜山,李建波.一种改进的 DV-HOP 传感器网络定位算法. *计算机工程与应用*,2007,43(34):141—143.
- 14 刘艳文,王福豹,段渭军,于超.基于 DV-Hop 定位算法和 RSSI 测距技术的定位系统. *计算机应用*,2007,27(3):516—518.
- 15 赵红,孙光,秦姣华,唐飞岳.夹角修正的 dv-hop 传感器网络节点定位研究. *计算机工程与应用*,2009,45(13):141—143.100—102.
- 16 Liu Y, Qian ZH, Liu D, Zhong H. A DV-Hop Positioning Algorithm for Wireless Sensor Network Based on Detection Probability, *ncm, 2009 Fifth International Joint Conference on INC, IMS and IDC*,2009.453-456.
- 17 张佳,吴延海,石峰,耿方.基于 DV-HOP 的无线传感器网络定位算法. *计算机应用*,2010,30(2):323—326.