

一种对片上网络中 Mesh 结构的改进策略及路由算法

蒋 明 孟铃珊

(西安电子科技大学计算机学院 西安 710071)

摘 要 Mesh 结构以其简单、规整、易于实现与扩展的特性而成为现在广泛应用的片上网络拓扑模型。对 2D-Mesh 结构进行了改进,给出了顶点互连的 Vertices interconnect Mesh(VMesh)结构;同时,提出了在此结构上的无死锁路由算法。最后,通过详细的计算证明,此结构可以减小网络直径和平均延迟,增加吞吐量;并用 gpNoCsim 模拟器对此结构及算法进行了仿真,结果表明,此结构具有较小的平均延迟和平均距离。

关键词 片上网络,拓扑结构,2D-Mesh 结构,VMesh 结构,路由算法

中图分类号 TP393 **文献标识码** A

Improved Structure for Mesh Topology Based on NoC and its Routing Algorithm

JIANG Ming MENG Ling-shan

(College of Computer Science and Technology, Xidian University, Xi'an 710071, China)

Abstract Mesh structure becomes a widely used NoC topology due to its simplicity, regularity, easy to implement and expand. This paper improved the 2D-Mesh structure. VMesh structure which connects every vertex to each other was presented. A deadlock-free routing algorithm based on this structure was proposed too. Finally, we proved that VMesh structure decreases the network diameter and the ideal average communication delay, increases the ideal throughput by detailed calculations. We also emulated the topology and the algorithm with gpNoCsim simulator and our results demonstrated a certain reduction in the average packet delay and routing hops.

Keywords Network on chip, Topology structure, 2D-Mesh structure, VMesh structure, Routing algorithm

1 引言

随着半导体工艺的发展,在同一芯片上集成的处理单元和存储核的数目不断增长,同时,它们之间通信量也大大增加。于是,这些处理单元和存储核之间的通信就成为影响这些系统性能的最大瓶颈。目前,片上多核系统(CMP)主要有 3 种互连方式,即总线、交叉开关、片上网络^[4-8]。

当通信核较少时,总线与交叉开关的互连方式能够满足它们之间的通信需求。但是,随着通信核数量的增加,总线连接方式就不能有效地处理多数据流之间的同步问题,而交叉开关连接方式则会造成硬件负载的指数级增长。片上网络就是一种有效解决多核之间的通信的多核互连方式。

与总线和交叉开关互连方式相比,片上网络具有更好的重用性和可扩展性。目前,片上网络拓扑结构的许多研究都是基于 Mesh 和 Torus 结构的。其中,Mesh 结构的物理实现和路由算法都很简单,所以基于 Mesh 结构的研究最为普遍;且 Mesh 结构已经应用于实践,如 Godson-3^[8]、TILE64 Processor^[9] 都是采用 Mesh 结构连接处理核的。可是,在 Mesh 结构中距离较远的节点之间的通信会有很大的延时,而且增加了网络拥堵的可能性^[10]。文献[11]中,Ogras 提出了一种

改进 NoC 性能的方法,即在 Mesh 结构中通过增加一些较远节点的连接来减少一些节点之间的通信延时。

根据以上思想,本文提出了一种对 Mesh 结构的改进策略,即 VMesh 结构;还提出了一种基于 VMesh 结构的 VXY 路由算法,且此算法不会产生死锁。实验结果证明,VMesh 结构在平均包延时、路由跳数和吞吐量等方面均优于 Mesh 结构。

2 VMesh 拓扑结构

2.1 基本结构

一个 $n \times n$ 的 Mesh 网络包含 $N(n \times n)$ 个节点,每个节点都有对应的坐标 (x, y) ,其中 $0 \leq x \leq n-1$ 且 $0 \leq y \leq n-1$ ^[12]。两个坐标为 (x_i, y_i) 和 (x_j, y_j) 的节点相连当且仅当它们之间的距离为 1,即 $|x_i - x_j| + |y_i - y_j| = 1$ 。图 1 为一个包含 25 个节点的 Mesh 拓扑结构。Mesh 结构没有重叠的连线,所以物理实现较为容易,可是在相距较远的节点间的通信延时比较大。如在图 1(a)中,点 $(0, 0)$ 与 $(4, 4)$ 之间通信需要 8 跳。

在介绍 VMesh 结构之前先定义以下概念。

定义 1 VMesh 结构中, $\{(0, 0), (0, n-1), (n-1, 0), (n-1, n-1)\}$ 为顶点集合,即如果节点坐标属于顶点集合,称其为顶

到稿日期:2011-08-15 返修日期:2011-12-14 本文受国家自然科学基金(60872041, 61072066), 国家科技部重大专项(2011ZX03005-002)资助。

蒋 明 男,副教授,主要研究方向为计算机应用、片上网络;孟铃珊 男,硕士生,主要研究方向为计算机应用、片上网络, E-mail: mls19861225@163.com。

点。

定义 2 各顶点之间的连接称为长连接。

定义 3 把一个 $n \times n$ 的 VMesh 拓扑网络分成 4 个部分, 即 A1, A2, A3, A4, 每个部分都包含一个顶点。每部分的划分如下:

$$A1: 0 \leq x \leq (n-1)/2, 0 \leq y \leq (n-1)/2$$

$$A2: (n-1)/2 < x < n, 0 \leq y \leq (n-1)/2$$

$$A3: (n-1)/2 < x < n, (n-1)/2 < y < n$$

$$A4: 0 \leq x \leq (n-1)/2, (n-1)/2 < y < n$$

定义 4 VMesh 结构中, 两节点 S 与 D 之间的跳数, 用 $L(S, D)$ 表示。假如, V 表示包含节点 S 部分的顶点, T 表示包含节点 D 部分的顶点。如果长连接能够减少 S 与 D 节点之间的跳数, 则 $L(S, D) = |x_V - x_S| + |y_V - y_S| + |x_T - x_D| + |y_T - y_D| + m$, 否则 $L(S, D) = |x_S - x_D| + |y_S - y_D|$ 。

其中, m 为长连接的跳数, $m = 1$ 。VMesh 结构是在 Mesh 结构的基础之上, 增加了 6 条使顶点互连的长连接来减小它们之间通信的延时。图 1(b) 为包含 25 个节点的 VMesh 结构。

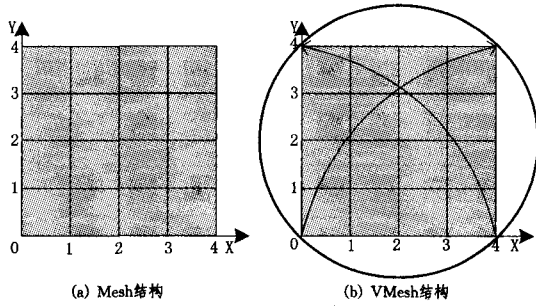


图 1 Mesh 和 VMesh 结构

2.2 VMesh 与 Mesh 的对比

网络直径、理想平均延迟和理想吞吐量是衡量网络拓扑优劣的 3 个主要参数。下面给出 VMesh 与 Mesh 结构的对比。

(a) 网络直径

网络直径是指网络中任意两个节点距离的最大值^[13]。通常, 为了提高网络的传输速度与质量, 需要减小网络直径的大小^[14]。容易看出, 对于一个 $n \times n$ 的 Mesh 网络, 其网络直径应为网络中两个对角顶点间的距离, 大小为 $2(n-1)$ ^[15]; 而 VMesh 结构的网络直径可以达到 $n-1$ 。所以 VMesh 结构的网络直径优于 Mesh 结构的网络直径, 下面给出证明。

设 d 为 VMesh 结构的网络直径, 则可分两种情况分析。即当 n 为奇数时, d 等于坐标 $(0, 0)$ 与 $((n-1)/2, (n-1)/2)$ 之间的跳数; 当 n 为偶数时, d 等于坐标 $(n/2-1, n/2-1)$ 与 $(n-1, n-1)$ 之间的跳数。两种情况下 d 的大小均为 $n-1$ 。图 2 示出 n 分别为奇数和偶数情况下的网络直径(图中长连接没有标出)。

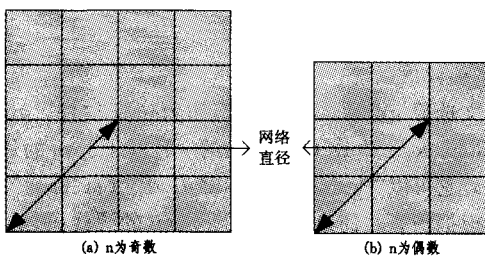


图 2 5×5 和 4×4 VMesh 结构的网络直径

(b) 理想平均延迟

理想平均延迟是指在网络没有拥堵的情况下, 所有节点的平均延迟的平均值。式(1)^[16]给出了计算理想平均延迟的方法。

$$T = H \times T_r + D/V + L/b \quad (1)$$

式中, H 是指拓扑网络中从源节点到目的节点的平均跳数, T_r 是指路由开关的延迟, D 是指源节点到目的节点的平均距离。通常情况下, H 等于 D , V 是指线上传输速率, L 是指包的大小, b 是指带宽。对于大多数的拓扑网络, 计算平均延迟主要是计算 H 和 D 。所以要确定理想平均延迟的优势只需比较 Mesh 结构与 VMesh 结构的平均距离 D 的大小。其中, Mesh 结构的平均距离 D 等于 $2(n^2-1)/3n$ ^[17], 当 n 为 4 时, D_{Mesh} 为 2.5。VMesh 结构比 Mesh 结构多了 6 条长连接, 所以它的理想平均延迟比 Mesh 结构的小。下面举例说明一个包含 16 个节点的 VMesh 结构(见图 1(b))。因为 VMesh 结构为对称结构, 所以所有节点可以被归为 3 类。对于每类节点只需取一个代表节点, 计算它到其它节点的距离, 最后再乘以该类节点的总数。表 1 计算了每类中一个节点到其他节点的距离。其中, d 为源节点到目的节点的跳数。

表 1 每类节点到其他节点所需跳数

源节点	(0,0)	(1,0)	(1,1)
$d=1$	(1,0)(0,1)(0,3) (3,0)(3,3)	(0,0)(1,1)(2,0)	(1,0)(0,1) (1,2)(2,1)
$d=2$	(2,0)(0,2)(1,1) (3,1)(1,3)(3,2) (2,3)	(0,1)(2,1)(1,2) (3,0)(0,3)	(0,0)(2,2)(2,0) (0,2)(3,1)(1,3)
$d=3$	(2,1)(1,2)(2,2)	(1,3)(3,1)(2,2) (3,3)(0,2)	(3,0)(0,3) (2,3)(3,2)
$d=4$	无	(2,3)(3,2)	(3,3)

每类节点到其他节点的总距离计算如下:

$$D(0,0) = 1 \times 5 + 2 \times 7 + 3 \times 3 + 4 \times 0 = 28$$

$$D(1,0) = 1 \times 3 + 2 \times 5 + 3 \times 5 + 4 \times 2 = 36$$

$$D(1,1) = 1 \times 4 + 2 \times 6 + 3 \times 4 + 4 \times 1 = 32$$

如图 1(b) 所示, 包含 16 个节点的 VMesh 结构的 3 类节点的个数分别为 4、8 和 4。则

$$D_{VMesh} = \frac{D(0,0) \times 4 + D(1,0) \times 8 + D(1,1) \times 4}{16 \times 16} = 2.0625$$

由于 $D_{VMesh} < D_{Mesh}$, 由式(1)可知, VMesh 结构的理想平均延迟小于 Mesh 结构的平均延迟。

(c) 理想吞吐量

理想吞吐量是指在理想的流控制和路由机制下, 单位时间内能成功地传送数据的最大量。式(2)^[16]给出了计算理想吞吐量的方法。

$$TP \leq 2b \times \frac{B_c}{N} \quad (2)$$

式中, b 为每条信道的带宽, B_c 为把单工网络分成两个相等的部分所需的切线数, N 为拓扑结构包含的节点数。

根据式(2), 对于两个网络在 N 与 b 相同的情况下, TP 与 B_c 成正比。对于一个 $n \times n$ 的 Mesh 结构, 把它分成两个相等的部分需要 n 条切线, 而 Mesh 结构均为双工网络, 所以 B_c 为 $2n$ 。而对于一个 $n \times n$ 的 VMesh 网络, 需要 $n+4$ 条切线完成相同的工作, 所以 B_c 等于 $2(n+4)$ 。则 Mesh 结构与 VMesh 结构的理想吞吐量分别为: $TP_{Mesh} \leq 4b/n$, $TP_{VMesh} \leq 4b$

$(n+4)/n^2$ 。显然,VMesh结构的理想吞吐量略大于Mesh结构的理想吞吐量。

3 VXY路由算法及无死锁性

XY路由算法是Mesh结构的基本路由算法。其基本思想是:先沿X方向路由到目的节点所在的列,再沿Y方向路由到目的节点。令当前节点的坐标为 (x_c, y_c) ,目的节点坐标为 (x_d, y_d) 。X+、X-分别表示X轴的正反两个方向;Y+、Y-分别表示Y轴的正、反两个方向。则XY路由算法描述如下:

<pre> if($x_c \neq x_d$) { if($x_c < x_d$) 沿X+方向路由; else 沿X-方向路由; } else if($y_c < y_d$) 沿Y+方向路由; else if($y_c > y_d$) 沿Y-方向路由; else 到达目的节点; (a) XY路由算法 </pre>	<pre> if(当前节点不是顶点) 采用XY路由算法 else if($L0 < L1$) { if($x_c = x_d, y_c \neq y_d$) 沿左或右长连接路由 else if($x_c \neq x_d, y_c \neq y_d$) 沿对角长连接路由 else if($x_c \neq x_d, y_c = y_d$) 沿上或下长连接路由 } else 采用XY路由算法 (b) VXY路由算法 </pre>
---	--

图3 XY路由算法和VXY路由算法

XY路由算法属于一种最短路径算法,而对于VMesh结构,并不适合采用这类算法。因为无论VMesh结构的节点数为多少,其顶点个数恒为4。若采用最短路径路由,随着节点数目的增加,在顶点处一定会造成阻塞。所以VXY路由算法采用的是一种非最短路径路由策略。其基本思想是:当包到达节点时,首先检查当前节点是不是顶点,如果不是,则采用XY路由算法;否则检查 $L0$ 是否小于 $L1$,若小于,表明长连接是较短的路径,则沿长连接方向路由;否则,采用XY路由算法。图3(b)为VXY路由算法的描述。其中,当前节点的坐标为 (x_c, y_c) ,目的节点坐标为 (x_d, y_d) ,源节点坐标为 (x_s, y_s) ,目的节点所在部分顶点坐标为 (x_t, y_t) 。 $L0, L1$ 分别代表通过和不通过长连接的路径距离。

VXY算法的关键在于 $L0, L1$ 的计算和长连接的选取。首先, $L0$ 和 $L1$ 的计算可以通过以下公式得到。

$$L0 = |x_t - x_d| + |y_t - y_d| + m \quad (3)$$

$$L1 = |x_c - x_d| + |y_c - y_d| \quad (4)$$

式中, $L0$ 的计算属于定义4中的特殊情况,即源节点已是顶点。然后是长连接的选取。观察可知,每个顶点都有3条与其他顶点相连接的长连接。由于VMesh结构具有对称性,因此这里只需选一个顶点为源节点。此处选 $(0,0)$ 为源节点进行说明。若 $x_c = x_t, y_c \neq y_t$,则数据包沿左或右长连接路由;若 $x_c \neq x_t, y_c \neq y_t$,则沿对角方向长连接路由;若 $x_c \neq x_t, y_c = y_t$,则沿上或下长连接路由;否则,目标节点位于本部分,采用XY路由算法。

VXY算法是一种基于虫孔交换机制的路由算法,而虫孔路由机制在传输过程中不释放节点资源,所以很容易形成死锁。文献[18]中提出了一种Turn Model模型,其基本思想是分析数据包在网络传输中存在的转向来消除网络传输中的环路,从而避免死锁。VXY算法主要通过限制4种转向来避免死锁,如图4所示,其中虚线为被限制的转向。

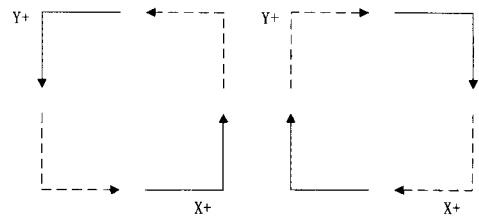


图4 VMesh结构中被限制的4个转向

在VMesh结构中共存在8种环,如图5所示。因为VXY算法限制了NE、SW、NW、SE4种转向,所以图5(a)~5(f)的环被破坏,这4种情况不会产生死锁。在图5(c)情况下,数据包遍历同行(或列)节点,根据VXY算法,在形成环之前数据包必已到达目的节点,所以,此种情况也不会产生死锁。图5(d)与此种情况相同。图5(g)、5(h)的情况也不会出现,根据VXY算法,若算法选择对角长连接,则说明目的节点在对角部分,所以不会再执行其他长连接。综上所述,VXY算法是无死锁的。

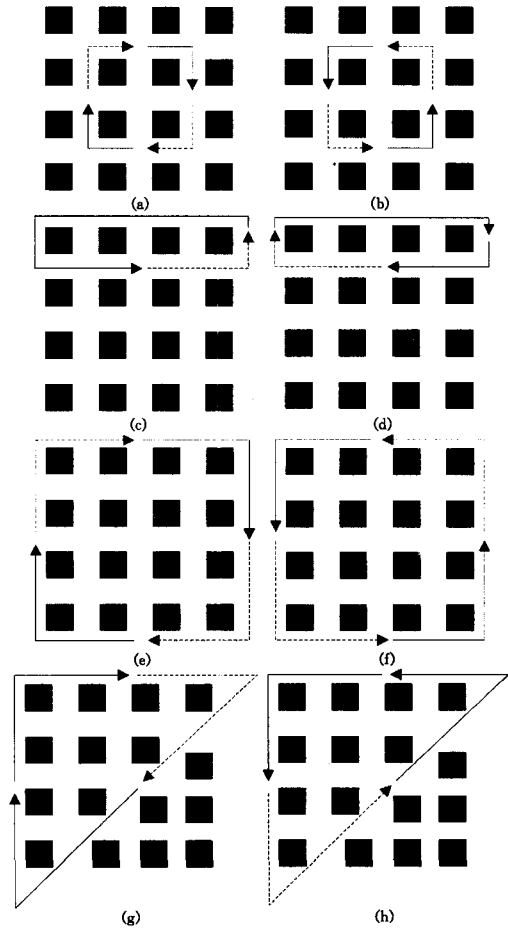


图5 VMesh结构中的8种环

4 仿真与性能分析

4.1 实验评估系统

本文选择gpNoCsim模拟器作为仿真软件。gpNoCsim是由Bangladesh University开发的一种基于Java的片上网络模拟器。它可以Mesh、Torus和Butterfly Fat Tree等拓扑结构进行模拟仿真。通过修改gpNoCsim的源代码,对VMesh结构和VXY算法进行了仿真,还对相同规模下的Mesh结构和VMesh结构的平均延迟及平均跳数进行了对比。模拟器

的配置如表 2 所列。

表 2 模拟器配置

参数	值
AVG_INTER_ARRIVAL	150
AVG_MESSAGE_LENGTH	200
FLIT_LENGTH	64
NUMBER_OF_IP_NODE	16, 36, 64, 100
CURRENT_VC_COUNT	4
NUM_CYCLE	100000
NUM_FLIT_PER_BUFFER	2

4.2 性能分析

本文对 Mesh 和 VMesh 结构在网络规模为 16、36、64、100 个节点的情况下分别进行了仿真,结果显示,VMesh 结构的链路和缓冲区的利用率都比 Mesh 结构的小。这说明 VMesh 结构降低了网络发生拥堵的概率。

平均延迟和平均跳数是片上网络的性能参数中非常重要的两个。图 6 和图 7 列出了 Mesh 和 VMesh 结构在不同规模下平均延迟和平均跳数的对比。实验结果显示,在不同网络规模下,VMesh 结构在平均延迟和平均跳数上均小于 Mesh 结构,使网络性能得到了一定的提升。当网络节点数为 36 时,VMesh 结构的平均延迟和平均跳数要比 Mesh 结构的分别低 3.40% 和低 5.10%。

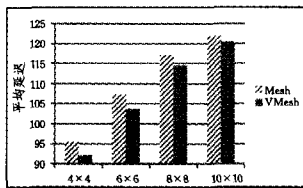


图 6 平均延迟对比

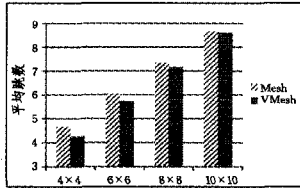


图 7 平均跳数对比

随着网络规模的增大,VMesh 结构较 Mesh 结构的优势逐渐减少。源节点和目的节点都是随机产生的,多数数据包都通过 XY 算法路由到目的节点。当顶点数固定为 4 时,网络规模越大,顶点所占的比例就越小,发挥的作用也就越小。但是当数据包到达顶点并通过长连接路由时,路由跳数可以明显减少,VMesh 结构的优势得以显现。

结束语 本文提出了一种 Mesh 结构的改进策略,即 VMesh 结构,并提出了 VXY 算法。通过计算网络直径、理想平均延迟和理想吞吐量,肯定了 VMesh 结构的优势。同时,仿真结果也证明 VMesh 结构确实比 Mesh 结构有更好的性能。未来将对 VXY 算法进行改进,并对 VMesh 结构的容错进行深入研究。

参考文献

[1] 欧阳一鸣,董少周,梁华国. 基于 2D-Mesh 的 NoC 路由算法设

计与仿真[J]. 计算机工程,2009,35(22):227-229

[2] 段新明,杨恩鲁,杨梅. 基于 PRDT 的 16 节点 NoC 路由算法[J]. 计算机工程,2007,33(9):12-14

[3] 师帅,窦文华,鲁佳. 基于二维 Mesh 结构片上网络的路由策略方法研究[J]. 计算机工程与科学,2011,33(5):32-37

[4] Olukotun K, Hammond L, Laudon J. Chip Multiprocessor Architecture Techniques to Improve Throughput and Latency[M]. 2009

[5] Kahle J A, Day M N, Johns C R, et al. Introduction to the Cell multiprocessor[J]. IEEE,2005,40(4):589-604

[6] Hammond L, Hubbert B A, Siu M, et al. The Stanford Hydra CMP[J]. IEEE,2000,20(2):71-84

[7] Wang wei, Tang Zhi-zhong, Qiao Lin. Summary of the Inter-connection in Chip Multiprocessor[J]. Computer Science, 2008, 35(9):7-9

[8] Wang Huan-dong, Gao Xiang, Chen Yun-ji, et al. Interconnection of Godsom-3 Multi-Core Processor[J]. Journal of Computer Research and Development, 2008, 45(12):2001-2010

[9] Tilea Company. TILE64 Multicore PCIe card connectivity solutions[M]. The world of electronic components, 2008:31-32

[10] Villanueva J C, Flich J, Duato J, et al. A Performance Evaluation of 2D-Mesh, Ring, and Crossbar Interconnects for Chip Multiprocessors[M]. ACM, 2009

[11] Ogras U Y, Marculescu R. "It's a Small World After All"; NoC Performance Optimization Via Long-Range Link Insertion[J]. IEEE, 2006, 14(7):693-706

[12] Daneshtalab M, Ebrahimi M, Mohammadi S, et al. Low-distance path-based multicase routing algorithm for network-on-chips[J]. The Institute of Engineering and Technology[J]. 2009, 3(5):432-442

[13] Shen Z. Average diameter of network structures and its estimation[C]//Proc. of the 1998 ACM Symp. on Applied Computing. 1998:593-597

[14] Zhu Xiao-jing, Hu Wei-wu, Ma ke, et al. Xmesh: A Mesh-Like Topology for Network on Chip[J]. Journal of Software, 2007, 18(9):2194-2204

[15] Chen Guo-liang, Wu Jun-min, Zhang feng, et al. Parallel Computer Architecture[M]. 2002:158-159

[16] Dally J, Towles B. Principles and Practices of Interconnection Network[M]. Morgan Kaufmann Publishers, 2003

[17] Dong Ying-fei, Wang Ding-xing, Zheng Wei-min. Exact Computation of the Mean Minimal Path Length of N-Mesh and N-Torus[J]. Chinese Computers, 1997, 20(4):376-380

[18] Glass C J, Ni L M. The Turn Model For Adaptive Routing[C]//ACM. 1992:278-287

(上接第 29 页)

[8] He T, Huang C, Blum B M, et al. Range-free localization schemes for large scale sensor networks[C]//Proc 9th Annual Int'l Conf on Mobile Computing and Networking (MobiCom). San Diego, CA, 2003:81-95

[9] Gupta R, Das S R. Tracking moving targets in a smart network [C]//The VTC Fall 2003 Symposium. Oct. 2003

[10] Zhong Zi-guo, Zhu Ting, Wang Dan, et al. Tracking with Unreliable Node Sequences[C]//28th IEEE Conference on Computer

Communications. Rio de Janeiro, Brazil, Apr. 2009

[11] Wan Jiang-wen, Guo Xiao-lei, Yu Ning, et al. Multi-Hop Localization Algorithm Based on Grid-scanning for Wireless Sensor Networks[J]. Sensors, 2011, 11:3908-3938

[12] Sheu J-P, Chen Pei-chun, Hsu C-S. A Distributed Localization Scheme for Wireless Sensor Networks with Improved Grid-scan and Vector-Based Refinement[J]. IEEE Transactions on Mobile Computing, 2008, 9(7):1110-1123

[13] 陈敏. OPNET 网络仿真[M]. 北京:清华大学出版, 2004