

WSANs 中基于六边形分区的多中心编址服务发现算法

杜景林^{1,2} 郑若钦² 谢立¹

(南京大学计算机科学与技术系 南京 210046)¹ (南京信息工程大学电子与信息工程学院 南京 210044)²

摘 要 针对 WSANs 中的服务发现问题,提出了一种基于六边形分区的多中心编址服务发现算法 MASD。算法综合考虑了节点的通信开销和存储负担,设计了一种多中心的编址方案,此方案采用局部计算,拥有计算量小的优点。Sensor 节点可按地址或者采用局部搜索的算法寻找到达附近 actor 节点的下一跳节点。通过实验发现,与现有的 imesh 算法相比,本算法具有更短的搜索距离和更少的通信开销。

关键词 WSANs,服务发现,六边形,编址

中图法分类号 TP393 **文献标识码** A

Multi-centre Addressing Service Discovery Algorithm with Cell-based Network Partition for WSANs

DU Jing-lin^{1,2} ZHENG Ruo-qin² XIE Li¹

(Department of Computer Science and Technology, Nanjing University, Nanjing 210046, China)¹

(School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China)²

Abstract A multi-centre addressing algorithm MASD with cell-based network partition was presented to solve the service discovery problem in WSANs. With the comprehensive consideration of the communication cost and per-node storage load, a multi-centre addressing scheme which uses no global computation and has lower computation was designed. Sensor node can find the next hop to the nearby actor through address or local searching. The simulated experiment results show that the algorithm has shorter search distance and more little communication cost compared to the imesh algorithm.

Keywords WSANs, Service discovery, Hexagonal, Addressing

1 介绍

WSANs 由大量固定的传感器(sensor)和少量的移动执行器(actor)组成,它们通过无线网络互联,用于感知现实世界,处理感知到的数据,当事件发生时进行决策和行动^[1]。由于 actor 节点需要为 sensor 节点提供移动支援服务,因此 sensor 节点在发现事件时需要向 actor 节点发起服务请求。为了减少服务发现的时延和能耗,有必要设计一种有效的服务发现算法,来寻找 sensor 附近的 actor 节点。

为了减少时延和能耗,我们希望找到最近 actor 节点来为 sensor 服务,除了距离因素,服务发现算法还必须考虑网络的限制,比如由于 WSAN 资源的限制,每个节点的存储负载应是相当的,且不能使用全局性的计算^[2]。文献[3]提出了 GLS 算法,GLS 把感知区域按网格划分,并且在网格上构造一棵四叉树。文献[4]中提出了一种地理哈希表 GHT,节点会根据数据的类型把数据转换成独特的位置信息,并根据一种复合的贪婪式路由协议来转发数据到相应位置。文献[5]提出了 GCLP 协议,GCLP 利用内容服务,周期性地向 4 个方向广播自己的位置。由于周期性位置信息的广播,GCLP 会产生大量的消息开销。文献[6]提出了一种基于标签的数据存储

和检索方案,文献[7]提出了一种基于 quorum 的位置服务。

以上算法(除了 GCLP)均存在一些缺点:如需预先知道整个网络的情况,需要频繁的全局性计算,多变的单节点存储负担,存在通信瓶颈等。而文献[2]提出了一种本地的 imesh 算法,此算法克服了以上缺陷。在 imesh 中,节点按 mesh 结构部署,每个 sensor 只能和网格中的一跳邻居通信,actor 节点会向 4 个方向广播自己的位置信息,sensor 节点在服务发现时沿着所在的行和列向 4 个方向发送搜索消息,当消息遇到 mesh 边界时便停止,此节点会记录最近的 actor 节点的位置信息。

另文献[8]中提出了一种用于 WSN 的蜂巢结构,其主要思想是把整个网络按六边形划分进行寻路,但是此算法只适用于固定节点及 1-中心的问题。

本文在文献[8]的基础上提出了多中心编址的服务发现算法 MASD 来与现有算法 imesh 进行比较,通过仿真实验,发现该算法在服务搜索时延和通信开销上是优于 imesh 的。

2 理论模型和定义

本文采用文献[2]中的模型,只考虑静态的 WSANs,即 actor 只在收到请求的时候才开始移动,且移动过程中不可为

到稿日期:2013-05-28 返修日期:2013-11-03 本文受国家自然科学基金(60873026,61021062),国家“九七三”重点基础研究发展规划(2006 CB303000,2009CB320705),江苏省科技支撑计划(工业)项目(BE2011195,BE2010178)资助。

杜景林(1974—),男,博士生,副教授,主要研究方向为无线传感器网络,E-mail:jldu@nuist.edu.cn;郑若钦(1989—),男,硕士生,主要研究方向为无线传感器网络;谢立(1942—),男,教授,主要研究方向为分布式系统。

其余的 sensor 服务。同时我们规定 actor 节点移动之后会动态更新当前的位置信息,且只有 actor 通信半径之内的 sensor 节点才会收到此更新。在初始场景中,actor 和 sensor 节点在整个网络中都是随机分布的。

WSANs 可以用图 $G(V, E)$ 表示,其中 V 是所有节点的集合, E 表示可直接连通节点之间的无线链路。我们定义 $|ab|$ 为两个节点 a 和 b 之间的欧氏距离,而 $|S|$ 为集合 S 的大小。下面给出邻居节点的定义。

定义 1(邻居节点) 节点 i 的邻居节点集, SR 表示 sensor 的通信半径

$$Neighbor(i) = \{j | j \in V, |ij| < SR\}$$

在此模型中,actor 是服务提供者,因此被称为 service provides(SPs), sensors 需要发起服务请求,所以被称为 service consumer(SCs),同时用 S_p 和 S_c 分别代表 SP 和 SC 的集合。这里令 $n = |V|$, $v = |S_p|$, 根据定义,有 $v \leq n$, $S_p \cap S_c = \phi$, 且 $S_p \cup S_c \subseteq V$ 。设 $P(a)$ 代表 SC 节点 a 发现的 SP 节点,我们按如下公式定义最近服务选择和附近服务选择。

定义 2(最近服务选择)

$$\forall a \in S_c, |aP(a)| = \min_{p \in S_p} \{|ap|\}$$

定义 3(附近服务选择)

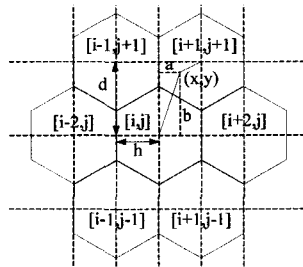
$$\forall a \in S_c, |aP(a)| \leq 2 \min_{p \in S_p} \{|ap|\}$$

此外,每个节点需要知道自身的位置,这是比较容易实现的。

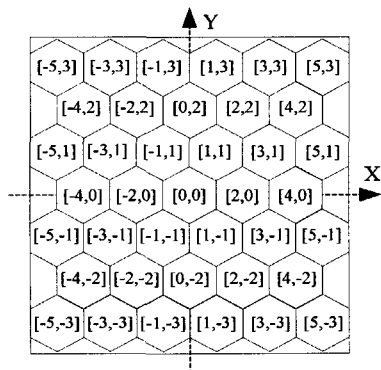
3 算法描述

3.1 蜂巢结构简介

文献[8]中蜂巢结构的构建主要是通过节点的坐标转换和编址来完成的。整个网络按六边形进行分簇,每个六边形称为 hexagonal cell,其中心点落在 (id, jh) 的位置, i 和 j 是整数,此分区单元被命名为 $[i, j]$, $[i, j]$ 称为 cell 坐标。图 1(a)给出了六边形 $[i, j]$ 及其邻居的命名情况和直角坐标的联系,图 1(b)给出了整个蜂巢结构的单元命名情况。



(a) cell 坐标与直角坐标的联系



(b) cell 坐标

图 1

对于随机分布在网络中的每个节点,可以根据文献[8]中的算法 2 把自身的直角坐标转化为对应的 cell 坐标。之后,需要把 cell 坐标 $[i, j]$ 转换成特定的 cell 地址 $[H, I]$, 这个地址可以在数据传输的过程中用来寻址。图 2 说明了 cell 编址的方式,转换规则可以参考表 1。

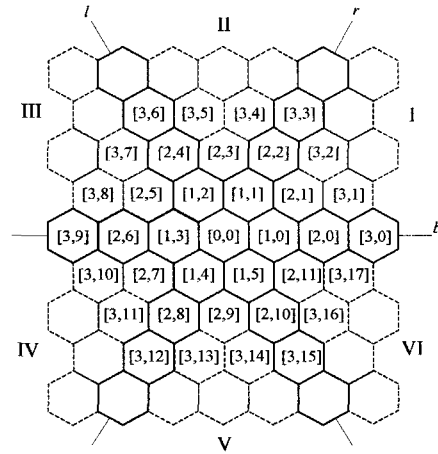


图 2 cell 的编址方式

表 1 坐标转换成地址的规则

象限	判断条件	转换结果
1	$i > j, j \geq 0$	$[i, j] \Rightarrow [(i+j)/2, j]$
2	$ i \leq j , j > 0$	$[i, j] \Rightarrow [j, 2H - (i+j)/2]$
3	$ i \geq j , j > 0, i < 0$	$[i, j] \Rightarrow [(i + j)/2, 3H-j]$
4	$ i > j , j \leq 0, i < 0$	$[i, j] \Rightarrow [(i + j)/2, 5H+ j]$
5	$ i \leq j , j < 0$	$[i, j] \Rightarrow [j, 2H + (i+j)/2]$
6	$i > j, i > 0, j < 0$	$[i, j] \Rightarrow [(i + j)/2, 6H- j]$

当整个网络的节点编址完成之后,可利用文献[8]中的算法 3 找到去往中心的下一跳 cell。有了之前的编址,寻路算法是非常简单的,且仅靠局部算法即可实现。文献[8]中考虑的是 1-中心问题,我们在此基础上提出了自己的算法。

3.2 基于多中心分区编址的服务发现

sensor 发现事件时要向 actor 汇报,此时 actor 就相当于蜂巢结构的中心节点,但是文献[8]的方案只适用于 1-中心的情况,而本文的服务发现模型中有多个 actor 节点,所以必须要扩展之前的方案。为此,我们从减少通信开销和存储负担的角度考虑,提出了一种多中心的分区编址方案。此方案在 cell 坐标的基础上,把整个网络分为 n 个区域(n 是 actor 节点的数量),每个区域以 actor 为中心。每次 actor 移动过之后会在其通信半径内进行广播,通知其通信区域内的 sensor 节点自身的位置,然后可以根据相对坐标对 sensor 进行编址。这种限定范围的广播的通信代价是比较小的。

3.2.1 actor 通信半径内 sensor 的服务发现

为了减少多中心编址的重叠性和 actor 节点更新位置信息的复杂性,每次 actor 节点移动之后,只对 actor 通信半径 AR 之内的 sensor 节点进行编址。这些 sensor 节点可称为被 actor 节点覆盖的节点,它们会收到来自 actor 节点的位置消息,而 actor 节点也只需广播一跳的位置信息即可,所以通信开销很小,编址算法如下:

算法 1

- Input: $[i_s, j_s]$; sensor 的 cell 坐标, $[i_a, j_a]$; actor 的 cell 坐标
- Output: $[H, I]$; sensor 的分区地址

3. $i = i_s - i_a$
4. $j = j_s - j_a // i, j$ 是相对坐标值
5. transform $[i, j]$ to $[H, I]$ // 见表 1, H 代表距离 actor 的跳数, I 是与象限有关的量

经过算法 1 的计算,可以得到每个 sensor 节点的分区地址,之后 actor 通信半径内的所有 sensor 节点就可以快速寻找到向 actor 节点发送数据的路径,如此便节省了服务搜索的时间和通信开销,图 3 展示了分区编址的效果。如果一个 sensor 处在多个 actor 的通信半径之内,它也可以根据多个地址 $[H, I]$ 中的 H (即跳数)来判断哪个 actor 节点离自己最近,选择跳数少的向其发送服务请求,如果距离 actor 的跳数相同,则可根据具体的位置信息进行进一步的比较来选择出较近的 actor 节点向其发送服务请求。

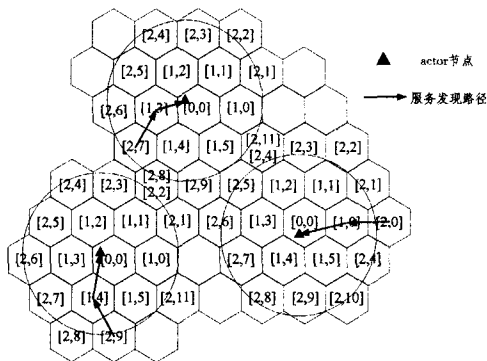


图3 分区编址示意图

3.2.2 actor 通信半径外 sensor 的服务发现

由于存在部分 sensor 节点在 actor 节点的通信半径之外,因此需要靠局部搜索来找出其附近 actor 节点或者是知道 actor 节点位置信息的 cell。我们的方案是从节点自身出发沿着 6 个方向寻找知道 actor 节点位置的 cell (即 actor 通信半径的边缘 cell),具体搜索路径如图 4 所示,规定每个 cell 需要存有自己邻居 cell 的信息,此信息包括是否被 actor 覆盖,以方便其余 sensor 节点的搜索。算法 2 描述了局部搜索算法。

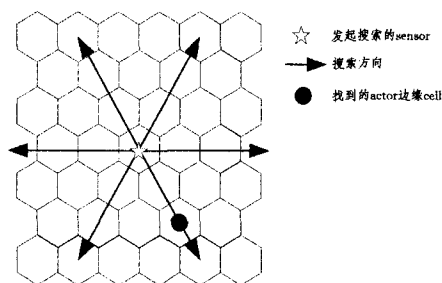


图4 服务搜索示意图

算法2 局部搜索算法

1. search in the assigned cell C_{sensor} // 搜索发起节点所在的 cell
2. if (a is found) $\forall a | a \in S_p$
3. stop
4. end if
5. search $C_i | C_i \in Neighbor(C_{sensor})$
6. if (a is found)
7. stop
8. else
9. $\forall C_n(i) | C_n(i)$ is the i th next cell and has the same direction of C_i

// 从 6 个邻居 cell 出发沿着 6 个方向搜索

10. for($i=0, i++$, until a is found)
11. search $C_n(i)$
12. end for
13. end if

采用算法 2 搜索,每个方向都有可能发现 actor 节点,如此各方向发现 actor 节点之后,搜索便停止。值得注意的是,由于每个节点被覆盖的概率是相同的,因此每个 sensor 的存储负担也是相当的(主要是邻居信息和地址信息),这样有利于平衡各节点的能耗。

3.2.3 搜索扩展

这种沿 6 个方向的搜索在一般情况下是可以找到被 actor 覆盖的 cell 的。但是如果出现某些特殊情况,比如某一区域的 actor 节点分布特别稀疏,有可能会使算法 2 在搜索的过程中漏掉一些附近的 actor 节点,因此算法中定义了一个扩展规则。

扩展规则:当 sensor 沿着 6 个方向搜索过 NH 跳 cell 的时候,如果还没有发现附近的 actor 节点,那么就当前节点为起点,各向顺时针和逆时针偏移 120° 发起新的搜索。设 actor 的通信半径为 AR ,每个六边形分区的边长为 r ,则 $NH =$

$$\min |AE| = \frac{\sqrt{3}}{2} \left(\frac{3AR}{\sqrt{3} \times r} + 2 \right) \text{hop}.$$

下面给出 NH 的证明:

这里直接考虑最坏的情况,即如图 5 所示,两个相邻的搜索路径正好错过了夹在中间的 actor 节点 C 。可以看出 actor 的通信半径上有 $\frac{AR}{\sqrt{3} \times r}$ 跳 cell,再加上每一跳搜索都会考虑邻居 cell,所以垂线 BC 的最短长度为 $\left(\frac{AR}{\sqrt{3} \times r} + 1 \right) \text{hop}$ 。

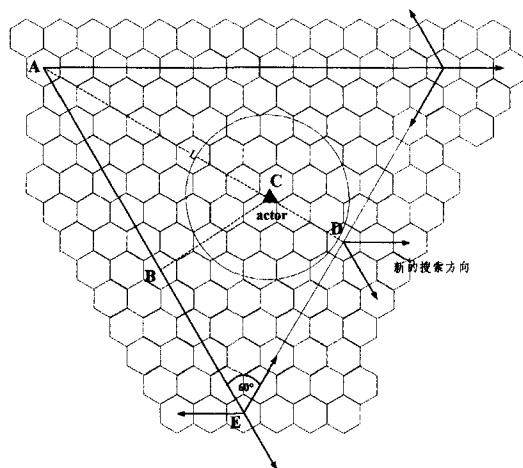


图5 扩展搜索示意图

在直角三角形 ABC 中,斜边 AC 的最短长度约为 $2\min |BC|$,因此 $\min |AD| = \min |AC| + \frac{AR}{\sqrt{3} \times r} \text{hop}$ 。在直角三角形 ADE 中,根据直角三角形各边关系,可以推出斜边 $\min |AE| = \frac{\sqrt{3}}{2} \left(\frac{3AR}{\sqrt{3} \times r} + 2 \right) \text{hop}$ 。因为 A 是搜索的起点, E 是当前搜索到的位置,所以搜索过的跳数 $NH = \min |AE| = \frac{\sqrt{3}}{2} \left(\frac{3AR}{\sqrt{3} \times r} + 2 \right) \text{hop}$,证毕。

扩展搜索的终止条件:当两个扩展搜索的路径相遇时,终止当前搜索。

重复之前的搜索:扩展搜索终止之后,每个搜索方向上将在偏移 120° 之后继续新的搜索,可以看出此时的搜索方向和原搜索方向相同。当搜索跳数再次达到 NH 时重复以上步骤。

这种扩展性搜索可以增加找到最近 actor 的几率,对于网络规模大且某一区域 actor 节点分布过于稀疏的情况有效。而通常可以在 NH 跳之内就搜索到附近的 actor 节点,这里所谓的附近 actor 节点是指至多第二近的 actor 节点。

4 算法仿真与性能分析

本文利用 MATLAB 对 MASD 算法进行了仿真,同时加入 imesh 算法进行比较。实验中统计 sensor 搜索到 actor 节点时经过的距离,搜索距离这一性能参数表明了 sensor 节点进行服务发现的搜索时间,搜索距离越短,则找到 actor 所需的时间也相应越少;通信开销主要包括构建网络结构时的信息更新和服务发现时转发的搜索消息,我们统一使用消息包的数量来衡量。

我们的实验参数如表 2 所列。在实验中,sensor 和 actor 节点在 $1000m \times 1000m$ 的区域内随机分布。actor 节点一开始保持静止,移动之后更新位置信息,同时在整个网络中随机产生 50 个事件,然后由 sensor 发起服务发现。

表 2 算法实验参数

参数	取值
网络区域	$1000m \times 1000m$
sensor 节点个数	1000
actor 节点个数	10, 15, 20, 25, 30
sensor 节点通信半径	50m
actor 节点通信半径	100m

由于 imesh 算法是沿着水平和垂直方向搜索的,因此即使是最理想情况下,其搜索距离也肯定是超过沿着六边形方向的搜索方式,且由于六边形分区方案只需要本地计算,因此相对于 imesh 也不会额外增加通信开销。图 6 给出了不同 actor 节点下平均每个 sensor 搜索到附近 actor 节点所经过的距离。从图中可以看出 MASD 算法相较于 imesh 有一定优势,这符合之前的分析。

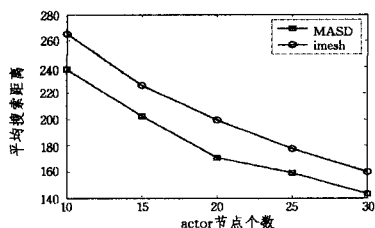


图 6 平均 sensor 节点搜索距离图

图 7 给出了不同 actor 节点下更新 actor 位置所发送的消息包总数,从图中可以看出 MASD 算法在构建多中心分区地址时所需发送的数据包远小于 imesh 中广播 actor 节点位置信息所转发的数据包。由于 imesh 算法中 actor 节点需要沿着 4 个方向向全网络广播自己的位置,因此其在建立 mesh 结构时通信开销会远大于 MASD 算法中仅局部更新 actor 位置信息的做法。

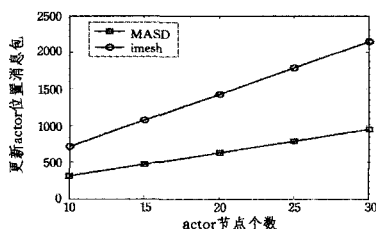


图 7 更新 actor 位置使用的消息包总数图

图 8 给出了不同 actor 节点下平均每个 sensor 节点在搜索到 actor 节点时所转发的消息包总数,可以看出由于 imesh 算法在任何情况下都需要由 sensor 发起服务搜索,因此其需要的转发消息包多于 MASD,而 MASD 算法中只是 actor 通信半径之外的 sensor 节点才需要发起服务搜索,因此效率很高;且随着 actor 节点数的增加,有更多的 sensor 节点被覆盖到,此时 MASD 算法的优势会更加明显。

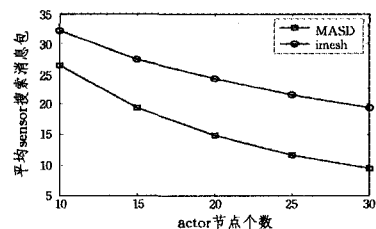


图 8 平均各 sensor 节点搜索产生的数据包图

结束语 本文针对 WSNs 中的服务发现问题,提出了一种基于六边形分区的多中心编址服务发现算法。算法考虑了 WSN 网络本身的一些资源限制,从减少计算量和通信开销的角度出发设计了一种多中心的编址方案来配合 sensor 节点进行服务发现。理论分析和仿真实验结果表明,相对于 imesh 算法,MASD 算法在搜索时间上拥有更好的性能,可以更加快速准确地找出附近的 actor 节点,而且在通信开销上也有比较大的优势,这样有利于延长网络的寿命。

参考文献

- [1] Mezei I, Lukic M, Veljko M, et al. Auctions and iMesh based task assignment in wireless sensor and actuator networks[J]. Computer Communication, 2012, 36(9): 979-987
- [2] Li Xu, Santoro N, Stojmenovic I. Localized distance-sensitive service discovery in wireless sensor and actor networks[J]. IEEE Transactions on Computers, 2009, 58(9): 1275-1288
- [3] Li J, Jannotti J, Couto D S J D, et al. A scalable location service for geographic ad hoc routing[C]// Proceedings of the ACM MobiCom, 2000: 120-130
- [4] Ratnasamy S, Karp B, Yin L, et al. GHT: A geographic hash table for data-centric storage[C]// Proceedings of the International Workshop on Wireless Sensor Networks and Applications (WSNA), 2002: 78-87
- [5] Tchakarov J B, Vaidya N H. Efficient content location in wireless ad hoc networks[C]// Proceedings of the IEEE International Conference on Mobile Data Management (MDM), 2004: 74-85
- [6] Fang Q, Gao J, Guibas L J. Landmark-based information storage and retrieval in sensor networks[C]// Proceedings of the IEEE

- [7] Stojmenovic I, Liu D, Jia X. A scalable quorum based location service in ad hoc and sensor networks[J]. International Journal of Communication Networks and Distributed Systems, 2007, 1(1):71-94
- [8] Erman A T. Multi-Sink Mobile Wireless Sensor Networks; Dissemination Protocols, Design and Evaluation[D]. Turkey; the University of Twente, 2011
- [9] Gungor V C, Akan O B, Akyildiz I F. A real-time and reliable transport (RT) 2 protocol for wireless sensor and actor networks[J]. IEEE/ACM Trans. on Networking, 2008, 16(2):359-370
- [10] 江海峰, 钱建生, 孙彦景, 等. WSN 中基于能量代价的能量优化路由算法[J]. 计算机科学, 2012, 39(1):73-76, 84
- [11] 王江涛, 陈志刚, 邓晓衡. WSN 中基于可信核心树的路由算法研究[J]. 计算机科学, 2011, 38(12):36-42
- [12] Mezei I, Malbasa V, Stojmenovic I. Robot to robot; communication aspects of coordination in robot wireless networks[J]. IEEE Robotics and Automation Magazine, 2010, 17(4):63-69
- [13] Selvaradjou K, Siva Ram Murthy C. Maximizing lifetime of mobile actors in wireless sensor and actor networks[J]. Ad Hoc & Sensor, Wireless Networks, 2010, 34(9):179-202
- [14] Bechkit W, Koudil M, Challal Y, et al. A New Weighted Shortest Path Tree for Convergecast Traffic Routing in WSN[C]//Computers and Communications (ISCC). 2012:187-192
- [15] Mian A N, Baldoni R, Beraldi R. A survey of service discovery protocols in multihop mobile ad hoc networks[J]. IEEE Pervasive Computing, 2009, 8(1):66-74

(上接第 23 页)

体系结构的可靠性。GeNIe 是由匹兹堡决策系统实验室研发的图形化开发工具, 支持贝叶斯网络、影响图等计算。计算得出方案 A 体系可靠的概率为 25.6%, 方案 B 体系可靠的概率为 19.0%。

由于方案 B 缺乏软硬件和相应机制的支持, 导致整个体系结构的成熟度、容错性较差, 虽然增加了一个决策者辅助决策, 但也弥补不了整体体系结构可靠性的降低。因此, 选择方案 A 作为可行方案。

结束语 扩展影响图能够充分描述人类的推理模式, 综合运用先验概率和后验概率, 既可避免先验概率带来的主观偏见, 又可避免后验概率带来的噪声影响。它采用因果关系表达影响图中的概率依赖关系, 采用定义关系对应 AHP 方法中的权重关系, 具有影响图与 AHP 方法的综合优势。

通过在影响图基础上引入新的节点与关系, 并在构建过程中运用防御图的思想, 使扩展影响图不仅较好解决了传统体系结构分析中的薄弱环节, 而且使整个分析流程更加直观和紧密。本文验证了扩展影响图对体系结构可靠性分析的可行性, 由于可靠性是评价一般体系结构方案的重要内容, 因此该方法可扩展到体系结构其它方面的分析。

参 考 文 献

- [1] Johnson P, Lagerström R, Närman P, et al. Enterprise Architecture Analysis with Extended Influence Diagrams[J]. Information Systems Frontiers, 2007, 9(2/3):163-180
- [2] Närman P, Schönherr M, Johnson P, et al. Using Enterprise Architecture Models for System Quality Analysis[C]//2008 12th International IEEE Enterprise Distributed Object Computing Conference. 2008:14-23
- [3] Johnson P, Lagerström R, Närman P, et al. Extended Influence Diagrams for System Quality Analysis[J]. Journal of Software, 2007, 3(2):30-42
- [4] Lagerström R. Analyzing System Maintainability Using Enterprise Architecture Models[J]. Journal of Enterprise Architecture, 2007, 3:33-42
- [5] Lagerström R, Johnson P, Närman P. Extended Influence Diagram Generation[C]//Proceedings of the Interoperability for Enterprise Software and Applications Conference. 2007
- [6] Sommestad T, Ekstedt M, Johnson P. Combining defense graphs and enterprise architecture models for security analysis[C]//12th International IEEE Enterprise Distributed Object Computing Conference. 2012
- [7] Sommestad T, Ekstedt M, Johnson P. Cyber Security Risks Assessment with Bayesian Defense Graphs and Architectural Models[C]//Proceedings of the 42nd Hawaii International Conference on System Sciences. 2009
- [8] Franke U, Sommestad T, Ekstedt M, et al. Defense Graphs and Enterprise Architecture for Information Assurance Analysis[C]//26th Army Conference. Orlando, 2008
- [9] Druzzdel M J, van der Gaag L C. Building Probabilistic Networks: "Where Do the Numbers Come From?" Guest Editors' Introduction[J]. IEEE Transactions On Knowledge and Data Engineering, 2000, 12(4):481-486
- [10] Druzzdel M J, van der Gaag L C. Elicitation of Probabilities for Belief Networks: Combining Qualitative and Quantitative Information[C]//UAI'95 Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. USA, 1995:141-148
- [11] 郭浩, 刘俊先. 基于体系结构的 C4ISR 系统可靠性指标体系[J]. 火力与指挥控制, 2010, 35(7):36-42
- [12] 吴兴波. 对武器装备可靠性指标体系的探讨[J]. 火炮发射与控制学报, 1997(4):53-56
- [13] 任建军, 张喜斌, 张恒喜. 基于 AHP 的可靠性指标确定方法研究[J]. 装备指挥技术学院学报, 2004(1):59-63
- [14] 李根成, 姜同敏. 空空导弹可靠性指标体系研究[J]. 中国惯性技术学报, 2006(8):88-92
- [15] 赵建立, 高会生, 赵生岗. 贝叶斯网络在可靠性评估中的应用[J]. 电力科学与工程, 2008, 24(2):51-53