

## 传感器网络中基于轨迹的多路径容错路由算法

杨伟丰, 彭召意, 孙星明

(株洲工学院 计算机科学与技术系, 湖南 株洲 412008)

(yangwfl1969@163.com)

**摘要:**基于轨迹提出一种新的多路径构造算法,其基本思想是源节点根据目的节点位置信息选择几条合适的转发路径,中间节点根据不同的贪婪转发策略建立相应的动态转发表转发报文。和其他算法相比,新算法具有分布式特性,路径建立只需局部信息即可;算法实现简单,无须大量的计算。理论分析和模拟实验也表明,新算法具有较好的容错性。

**关键词:**传感器网络;路由;多路径;容错

**中图分类号:** TP393.07 **文献标识码:** A

## Trajectory-based multipath tolerant routing for sensor networks

YANG Wei-feng, PENG Zhao-yi, SUN Xing-ming

(Department of Computer Science and Technology, Zhuzhou Institute of Technology, Zhuzhou Hunan 412008, China)

**Abstract:** In order to improve the tolerance of routing algorithms, this paper proposed a new algorithm based on trajectory to construct multipath. Its basic idea is that the source node selects several suitable curves to sink at one time and intermediate nodes build dynamic forwarding tables according to different greedy forwarding strategies. Compared with other algorithms constructing multipath, this new algorithm only needs information in one hop and doesn't need too much calculation, so it is distributed and can be realized easily. Theoretic analysis and simulation results show that it is more tolerant.

**Key words:** sensor networks; routing; multipath; tolerance

### 0 引言

由于无线集成网络传感器(Wireless Integrated Network Sensors)<sup>[1]</sup>具有数据采集、处理及通信的功能,故在环境与军事监控,地震与气候预测,地下、深水以及外层空间探索等许多方面都具有广泛的应用前景。但外界环境的不确定性使得布置成百上千的传感器协同工作成为必要,对由大规模无线集成网络传感器(传感节点)构成的传感器网络(Sensor Network, SN)的研究被认为是本世纪的一项具有重大意义的研究课题<sup>[2-4]</sup>。传感器网络相对于传统网络具有以下特性:1)节点分布极其稠密且数目很大,每个节点均维护全局信息是不可能的;2)节点的能量、存储空间及计算能力等资源非常有限,而且各种资源无法补充;3)网络内节点大部分是静止的,而且恶劣的工作环境使得节点失效的概率很高。因此,传统有线网络中基于链路状态或距离向量的路由算法以及Ad Hoc网络中的常用的DSDV, AODV等路由算法由于几乎没有考虑节点失效的问题因而并不适合于传感器网络,必须针对其特性研究新的路由算法,尤其是具有容错性的路由算法。

为了处理中间节点失效,可以使用应答和重传机制来恢复丢失的数据,但是这将引入大量附加的通信开销和延迟。而多路径(Multipath)路由算法可以明显提高系统的可靠性,具有直观的容错性,有研究人员对其进行了研究,提出了一些多路径构造算法。在Ad Hoc网络中, TORA(Temporally Ordered Routing Algorithm)<sup>[5]</sup>通过维持一个DAG得到多条无

循环路径。DSR(Dynamic Source Routing)<sup>[6]</sup>是每当有需要则采用Flooding的方法临时得到路径,文献[7]则通过减少DSR中flooding的频率来降低维持多条路径产生的开销。但尽管如此,由于能耗的原因,这些算法对于传感器网络并不适合。因而,研究人员又提出了新的容错路由构造算法。文献[8]提出的DD(Directed Diffusion)通过flooding构造一个梯度场,为建立多路径提供了条件。在此基础上,文献[9]根据建立的梯度场来构造多路径,并根据多条路径之间是否相交将Multipath分为两种:Disjoint Multipath和Braided Multipath。文献[10]基于DSR提出了MDR构造多路径,但算法中的路由请求仍是由flooding散布的。采用Flooding的方式进行梯度场的建立和路由请求的散布使得当网络规模很大或目的节点位置或兴趣变化比较频繁时,能量的消耗较大。文献[11]从另一个角度研究降低能耗的方法,提出将要传送的数据等分为 $N$ 份( $N$ 也是建立的路径数目),不是在每条路径上都发送一份完整的数据拷贝,而是只发送完整数据中的一份,目的节点只需要有 $K(K \leq N)$ 份数据片断就可以恢复全部数据。

本文针对传感器网络的特点,基于TBF(Trajectory-Based Forwarding)提出了一种新的多路径构造算法,算法将rumor routing和TBF结合进行兴趣散布,相比于flooding的方法,可以节约相当的能量;算法在源节点处可以方便地选择合适的路径,从而也可以灵活地构造出Disjoint Multipath或者Braided Multipath,实现非常简单;算法的分布式使得每个节点只需本地信息即可方便地构造动态转发表,本文提出了多种贪婪转发策略用以满足不同应用的要求。

收稿日期:2004-08-02;修订日期:2004-10-28 基金项目:国家自然科学基金资助项目(60373062)

作者简介:杨伟丰(1969-),男,湖南益阳人,讲师,硕士研究生,主要研究方向:计算机通信网络、计算机控制技术;彭召意(1968-),男,湖南衡阳人,高级工程师,硕士研究生,主要研究方向:计算机通信网络;孙星明(1963-),男,湖南益阳人,教授,博士生导师,博士,主要研究方向:计算机通信网络、分布式数据库。

## 1 TBF 模型

TBF 是针对大尺度稠密网络而提出的一种路由思想。其基本思想是源节点选定一条转发曲线并将曲线编码到每个报文中,中间节点解码曲线并根据贪婪转发策略独立转发报文,直到目的节点。TBF 所具有的源路由特征使得节点可以根据实际应用选择最合适的路由轨迹,而采用贪婪法选取下一跳节点也使得节点无须维护大量路由信息即可顺利地进行节点的选取和报文的转发,而且 TBF 是分布式的。

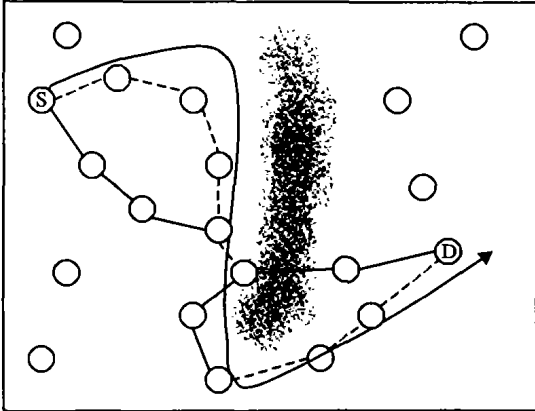


图1 TBF 转发模型

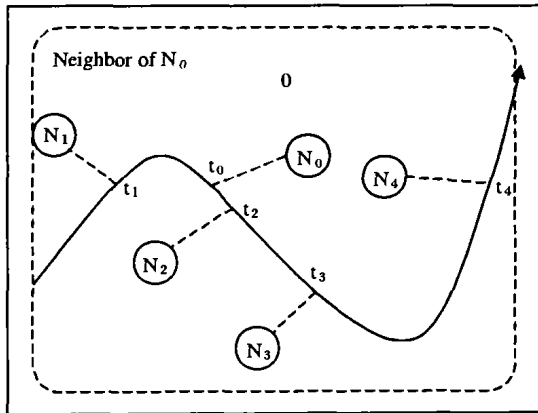


图2 下一跳选取模型

基本模型如图1和图2所示,图1是一般转发模型,图2是具体的下一跳选取模型。源节点和目的节点分别为S和D,  $N_0$  为转发路径上某一节点。图1显示由于障碍物的存在,使得按照传统方法选取的最佳路径无法转发报文到D,但采用TBF,由S选定一条曲线(如正弦曲线),并将曲线编码到报文中,收到报文的中间节点根据解码曲线并根据贪婪转发策略选取下一跳节点则可以到达D。曲线可以采用各种表示形式,但由于参数表达式可以表达非常复杂的轨迹而得到了广泛的应用。而如果中间节点需要改变原曲线轨迹,则需要考虑不同曲线的平滑连接问题,在这方面  $n$  次 B 样条曲线因具有  $C^{n-1}$  阶的自然连接性质而比文献[12]所采用的 Bezier 曲线有优势。

当前节点如何选取下一跳节点是 TBF 的另一关键之处(如图2所示)。根据不同的贪婪选取策略,将得到不同的下一跳节点。本文在2.3节中详细描述了儿种可选的下一跳选择策略。文章针对图2使用如下几个术语:  $SET_N$ ,  $t_N$ ,  $\Delta t$ ,  $\delta_N$ ,  $Dist_N$ 。其中:下标表示节点标识,  $SET_N$  表示  $N$  的下一跳候选集,一般将节点的直接邻居集合作为候选集,如  $SET_N$  为  $\{N_1, N_2, N_3, N_4\}$ ;  $t_0$  指的是  $N_0$  在曲线上的投影点所对应的参数值;  $\Delta t$  指的是参数  $t$  的步长,由于曲线上有无穷多个点,只能计算一定数目的点(如参数为  $t + n\Delta t$  的点)代表整个曲线;  $\delta_N$  表示  $N$  所对

应的参数  $t$  与  $SET_N$  中各点对应的  $t$  之间的差值;  $Dist_N$  表示  $N$  和其投影点的距离。 $SET$  的构造方式可能导致选取错误的下一跳节点,因此,本文在构造候选集时,不仅要求下一跳节点在当前节点的通信范围内,而且要求其  $Dist$  小于某个阈值(将根据网络节点密度进行调整)。

## 2 基于 TBF 的多路径构造算法

TBF 是一种基于位置的路由算法,节点如何确定其自身的位置信息是首先要解决的问题。

### 2.1 位置信息

卫星定位技术的发展,使得地球上任意一点只要有一个简单的接收装置即可接收卫星传回的该点的经纬度信息。这也使得 SN 中传感节点可以通过全球定位系统(Global Positioning System, GPS)<sup>[13]</sup> 得到自身精确的位置信息。对低能量节点组成的网络, Nath 和 Niculescu 提出了 LPS(Local Position System)<sup>[14]</sup>, 这种相对位置方法用于节点确定其位置,这也适用于传感器网络。为了简化算法描述,本文采用 GPS 定位方法确定节点位置。

### 2.2 B 样条曲线

$N$  次 B 样条曲线是计算机图形学中常用的一种性质良好的参数曲线,本算法选取二次 B 样条曲线来确定报文轨迹。曲线定义式和计算公式如下所示,其中  $F_{l,n}(t)$  被称为基函数,是决定曲线性质的主要部分。

$$P_{0,2}(t) = \sum_{l=0}^2 V_l F_{l,2}(t) \quad (1)$$

$$F_{l,n}(t) = \frac{1}{n!} \sum_{j=0}^{n-l} (-1)^j C_{n+1}^j (t+n-1-j)^n, \quad 0 \leq t \leq 1, l = 0, 1, \dots, n \quad (2)$$

$$P_{0,2}(t) = [t^2 \quad t \quad 1] \frac{1}{2} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 2 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \end{bmatrix}, \quad 0 \leq t \leq 1 \quad (3)$$

从式(3)可以看出,只要选定型值点  $V_{0-2}$ , 就可以完全确定对应段的样条曲线,而且可以通过选择合适的型值点使得源节点和目的节点分别成为曲线的起点和终点。对二次 B 样条曲线而言,选定3个型值点可以完全确定一条曲线。设定网络源节点为  $P_{0,2}(0)$ , 目的节点为  $P_{0,2}(1)$ , 则可以根据如下公式得出合适的  $V_0$  和  $V_2$ :

$$\begin{cases} t=0, P_{0,2}(0) = (V_0 + V_1)/2 \\ t=1, P_{0,2}(1) = (V_1 + V_2)/2 \end{cases} \Rightarrow \begin{cases} V_0 = 2P_{0,2}(0) - V_1 \\ V_2 = 2P_{0,2}(1) - V_1 \end{cases}$$

关于多段曲线的连接性问题,根据曲线定义可以很容易推知相邻二次 B 样条曲线在终点(起点)处是自然连接好的,而且具有  $C^1$  阶连续。如:给定四个型值点  $V_{0-3}$  可以确定2段二次 B 样条曲线,  $V_{0-2}$  确定的曲线终点为  $(V_1 + V_2)/2$ , 而  $V_{1-3}$  确定的曲线起点为  $(V_1 + V_2)/2$ , 显然曲线已经是连接的。通过求导数也可以推导出二次 B 样条曲线具有  $C^1$  连续性。

### 2.3 贪婪转发策略

为了保证报文的转发是沿着选定的曲线前进,必须针对当前节点  $I$ , 解决如何在  $SET_I$  集中选取下一跳的问题。本文提出了如下几种选取策略,而最终实现算法应根据应用的不同而选取最合适的一种(是从  $SET$  中选取):

1) 选取沿曲线前进最多的节点。策略就是选择  $\delta_i$  最大的节点。通常这可以得到最小的延时,在对数据实时处理要求

比较严格的应用中可以采用,比如:在对地震等灾害监测应用中,要求监测数据应尽快到达控制中心以便采取措施,这时应使数据转发经过尽可能短的路径也就是跳数尽可能少。

2) 选取沿曲线前进最小的节点。本策略就是选择  $\delta_i$  最小的节点。通常这种方式可以获得比较接近设定曲线外形的转发轨迹。这种方式可以在一些对转发轨迹要求严格的应用中使用,比如:在军事应用中,为了避免危险区域可以设置特定的曲线,要求实际转发轨迹应该严格靠近设定的曲线。

3) 选取距离曲线最近的节点。本策略就是当前节点  $I$  选择  $Dist$  最小的节点。

4) 随机选取一个节点。本策略就是当前节点  $I$  从上面三个节点中随机选择一个,通常是在各指标之间取得一种折中。

5) 策略 1)、2) 和 3) 之间进行折中。本策略就是当前节点  $I$  将其候选集合内各节点的  $Dist$  和  $\delta_i$  加以综合考虑。根据应

用的不同而给  $\alpha$  和  $\beta$  指定不同的值,调整  $Dist$  和  $\delta_i$  的权重,根据公式  $Value = \alpha Dist + \beta \delta_i (0 \leq \alpha, \beta \leq 1)$  计算的值选取最合适的节点。

6) 选取剩余能量最多的节点。本策略就是选择剩余能量最多的节点,通常这种方式可以较好地平衡能耗。

需要注意的一点是,无论采用何种选取策略都应使得下一跳节点沿着曲线前进的方向进行转发,也就是说要保证下一跳节点  $N$  的  $t_N$  要大于当前节点  $I$  的  $t_I$ 。

## 2.4 多路径构造算法

多路径构造算法的具体实现由如下几个阶段组成:邻居感知阶段 (Neighbor-Percept Stage, NPS)、兴趣散布阶段 (Interest-Distribution Stage, IDS) 和路径构造阶段 (Path-Construction Stage, PCS)。

各个阶段中用到的报文格式如表 1 所示。

表 1 算法中出现的报文

报文名称	报文包含的域	报文长度/Byte	出现阶段
TBP	PT=1, ID <sub>I</sub> , POS <sub>I</sub> , E <sub>I</sub>	0.5+1.5+4+3=9	NPS
DAP	PT=2, TTL, DT, ID <sub>I</sub> , UH	0.5+1+0.5+1.5+1.5=5	IDS
IAP	PT=3, TTL, DT, POS <sub>I</sub> , ID <sub>I</sub>	0.5+1+0.5+4+1.5=6.5	IDS
PCP	PT=4, MT, V <sub>i</sub> , ID <sub>S</sub> , ID <sub>D</sub> , TTL, NH, t	0.5×2+3×4+3×1.5+1+2=20.5	PCS

表 1 提到的报文将在算法实现的具体阶段中加以说明,报文中的各数据域含义说明如下:S 和 D 表示源和目的;PT (Packet Type) 表示报文类型;ID<sub>I</sub> 表示任意一个节点  $I$  的标识;POS<sub>I</sub> 表示节点  $I$  通过 GPS 确定的地理位置;E<sub>I</sub> 表示节点  $I$  目前的剩余能量(以节点尚可转发的报文数代表);TTL 表示报文的生存时间,以跳数(hops)衡量;DT 表示需要的或者搜集到的数据类型;MT 表示路径编号;DAP, ID 表示收集到数据的节点标识;UH(up-hop)表示当前节点的上一跳节点标识;NH(next-hop)表示当前节点的下一跳节点标识;IAP 中的 ID 和 POS 分别表示需要数据的节点标识和地理位置,V<sub>i</sub> 表示确定曲线的型值点,t 表示转发报文的节点在曲线上的投影。

### 2.4.1 邻居感知阶段

在本算法中,每个节点只需知道 1hop 内的信息,而无需全局信息。为了获得局部拓扑信息,当节点被散布后,每个节点发送拓扑建立报文(Topology Building Packets, TBP, 其类型标识为 1),报文内容如表 1 所示。需要说明的是 TBP 报文决不会被转发,而且每个节点将在初始被散布后的一段合适的时间内发送 3 次 TBP 以保证其被邻节点收到。节点收到 TP 为 1 的报文,则将报文中的 ID, POS, E 等内容提取出来,并查找由三元组构成的邻居表(Neighbors Table, NT),如果 NT 中还没有该 ID,则将其填入 NT 中。构成 NT 的三元组由 ID, POS 和 E 组成,节点的每个邻节点将占据 NT 中的一行。

### 2.4.2 兴趣散布阶段

本算法结合 rumor routing 和 TBF 进行兴趣散布,而不是采用 flooding 的方法。其基本思想可以描述如下:1) S 和 D 分别发送各自的代理报文 DAP (Data Agent Packets) 和 IAP (Interest Agent Packets),代理报文携带数据、兴趣等相关信息;2) S 和 D 随机独立选取路径转发代理报文;3) 如果两种代理报文在某个节点相遇,节点将发出相应的报文反向沿着 DAP 所经过的路径到 S,告知有关 D 的兴趣等信息。而收到 DAP 和 IAP 的中间节点将分别记录如下内容:DAP, IDS, DAP, DT, DAP, UH 和 IAP, POSD, IAP, IDD 和 IAP, DT。如果某个节点收到了 DAP 和 IAP 并发现 DAP, DT 和 IAP, DT 相

同,则通过记录下来的 DAP 内的信息反向转发告诉 S 有关 D 的信息。

通过这个阶段,S 就可以得知 D 的位置和所需数据。显然,本阶段的关键是保证 DAP 和 IAP 的转发路线有足够高的相交概率。为了保证相交,本文通过让 DAP 和 IAP 各自通过 5 条相互独立随机选取的路径转发。文献[15]通过理论分析和模拟实验表明:在选取多条(>4)轨迹的情况下可以保证足够高(99.7%)的相交概率。

但是具体采用 TBF 还是 flooding 进行兴趣散布,应该根据网络规模进行选择,在算法实现复杂度和算法开销这两个方面进行折中。如果网络规模较小,采用实现简单的 flooding,其能耗不会很大;而如果网络规模较大而且兴趣变化比较频繁,采用 TBF 进行兴趣散布将会节省很多的能量。

### 2.4.3 路径构造阶段

经过上述两阶段,S 将明确 D 的位置信息和感兴趣的数据类型。随后节点即可基于 TBF 采用 B 样条曲线进行转发。在此阶段节点  $I$  对报文的处理,可以如下描述(以图 2 为例)。需要说明的是,本阶段将采用 PCP (Path-Construction Packets) 建立多条路径,而具体数据转发将有一个专门的阶段处理。收到 PCP 的节点将根据选取策略在 SET 中选取下一跳节点填充数据转发表(Data Forwarding Table, DFT),随后的数据转发就可以按照 DFT 进行转发,DFT 结构如下:

(DT, MT, ID<sub>S</sub>, ID<sub>D</sub>, NH, UH, FP)。

其中 FP (Forwarded Packets) 表示已经转发的由 S 到 D 且类型为 DT 的数据量。根据 FP 可以衡量当前节点在该次转发中已经消耗的能量,使得 UH 从各节点能耗平衡的观点判断是否需要改变其下一跳节点。

本文通过以下两种方法构造多条路径。

方法一 采用不同型值点构造由不同的 B 样条组成的多路径。其基本思想是,源节点为每条 B 样条曲线选择合适的型值点,中间节点采用相同的贪婪转发策略构造 DFT,本文选择 2.3 中提出的策略 3) 作为贪婪转发策略。本方法将很容易构造 Disjoint Multipath,也就是可以使路径完全不相交。

方法二 采用相同型值点而不同贪婪转发策略构造不同的B样条曲线组成多路径。其基本思想是,源节点选择相同的型值点,但是选用不同的贪婪转发策略构造DFT,本文选择2.3中提出的策略1)~3)分别作为不同路径的贪婪转发策略。本方法构造的多路径可能是Braided Multipath,也可能是Disjoint Multipath,这与网络内的节点密度等实际情况有关。

假设网络内任一节点标识为 $N$ ,算法实现可简单描述如下:

```

If  $ID_N = PCP.ID_D$  Then
    PCS is ended;
Else
    If  $((PCP.TTL = PCP.TTL - 1) = 0 \vee N \neq PCP.NH)$  Then
        PCP is discarded;
    Else
        If there isn't an item in DFT to suit PCP.DT, PCP.IDs and PCP.IDD Then
            N selects next-hop node in the SETi according to NT and selecting strategy;
            If N is the same one which is selected as next-hop node Then
                PCP is discarded;
            Else
                N adds a new item to DFT according to PCP;
                N replaces PCP.NH using DFT.NH to rebuild new packets;
                N adjusts DFT.E;
                N broadcasts the new packets to all its neighbors;
        Else
            PCP is discarded;
    Endif;
Endif;
Endif;

```

经过PCS后,几条路径将被建立,而随后的数据转发可以采用不同的方式。如果节点的能量充足可以在每条路径上都发送一份数据的完整拷贝,如果节点能量已经较低,也可以根据文献[10]中的方法将数据分为 $N$ 份(路径有 $N$ 条),每条路径上发送一份,只要有 $K(K \leq N)$ 份到达即可完整地恢复全部数据。

### 3 理论分析和模拟比较

为了分析基于TBF的多路径路由算法的性能,本文根据如下定义的几个衡量尺度对算法进行了理论分析和模拟实验比较。

#### 3.1 衡量尺度

本文采用以下指标来衡量本算法的性能:

时间复杂度(Time Complexity, TC) 算法在选定参与转发报文的节点时所耗费时间的量级。

消息复杂度(Message Complexity, MC) 算法在选定参与转发报文的节点所发送分组数的量级。

消息域(Message Field, MF) 算法在选定参与转发报文的节点时所必须知道的拓扑图的信息范围。

容错度(Tolerance, Tol) 容错度是当网络内有节点失效时可用路径的平均值。具体而言就是,在节点失效的情况下,如果某条路径仍然可用,则该路径的值为1,否则为0。

#### 3.2 理论分析

本文从TC,MC和MF这三个方面,在理论上将基于TBF的多路径算法与SN中两种具有代表性的路由算法DD和LEACH进行了比较。具体内容列在表2中。

对于DD而言,算法初始将建立一个梯度场进而可以构造多路径,随后数据即可自动到达Sink节点,由于其TC,MC是只根据邻居节点洪泛的局部信息而确定的,因而二者均为

$O(n)$ ;同样,LEACH算法是一种动态的成簇算法,可以具有一定的容错性(即便是簇头失效也只影响一轮内的数据转发而已)。LEACH算法在每轮中,只需执行一次簇头选取动作,其TC也为 $O(n)$ 。文献[16]已经证明:在异步无线Ad Hoc网络中,若其unit-disk图(当且仅当两点之间的距离不大于1时,才有一条边的图)为环,则分布式地计算簇头的算法所需要传输的信息均不小于 $O(n \log n)$ ,所以LEACH的MC为 $O(n \log n)$ ,因此可以认为传感器网络是一种节点移动速度为0的特殊无线网络。而对本算法来说,其复杂度主要由PCS决定,根据算法的伪代码描述可以知道,其TC和MC不会高于 $O(n)$ ,而算法的分布式特性使得其MF也只需一跳信息即可。

表2 不同算法在TC,MC,MF尺度上的比较结果

比较尺度	DD	LEACH	TBF的B样条实现
TC	$O(n)$	$O(n)$	$\leq O(n)$
MC	$O(n)$	$O(n \log n)$	$\leq O(n)$
MF	1 hop	1 hop	1 hop

#### 3.3 模拟实验

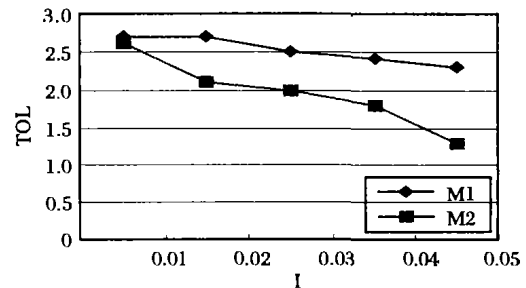


图3 固定 $N$ 和 $R$ 而 $I$ 型失效中节点比例改变

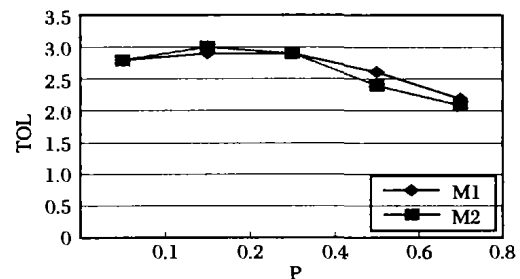


图4 固定 $N$ 和 $R$ 而 $P$ 型失效中节点比例改变

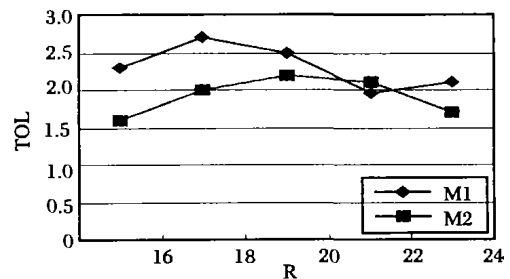


图5 固定 $N$ 和 $I$ 型节点失效比例而 $R$ 变化

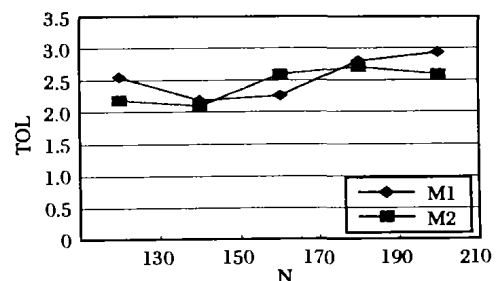


图6 固定 $R$ 和 $P$ 型节点失效比例而 $N$ 变化  
本文在模拟实验中从Tol指标上衡量了算法的性能。模

拟实验是在一个给定范围的区域( $150 \times 150$ )内,随机散布  $N$  个节点,每个节点有一定的通信半径  $R$ 。通过分别改变  $N$  和  $R$  来比较在不同的失效模型下,不同多路径构造算法的性能。根据文献[9],节点的失效模型可分为两类:I 型失效和 P 型失效。I 型失效指的是某些节点随机独立失效,而 P 型失效指的是在某个半径范围内的节点整体失效。

图3~图6是分别表示了网络内节点数、节点传输半径、I 型失效的节点数、P 型失效的节点数改变时,对算法容错度的影响。图中 M1 和 M2 分别表示不同型值点构造多路径和不同策略构造多路径的算法。

从图中可见,方法一对于 I 型失效的容忍度比方法二好,尤其是在失效节点比例较高时,这是因为方法一构造的多路径相互之间几乎没有影响,而两种方法对 P 型失效的容忍度相差不大。当  $R$  变化或  $N$  变化时,相比之下,方法一要稍好于方法二。但方法一在具有较好性能的同时也需要更多的开销,因而需要在性能提高和附带开销这两方面进行有效的折中和平衡。

#### 参考文献:

- [1] ACRE J, CLARE L. An integrated architecture for cooperative sensing networks[J]. IEEE Computer, 2000, 33(5): 106 - 108.
- [2] ESTRIN D, GOVINDAN R, HEIDEMAN J, et al. Next century challenges: Scalable Coordination in Sensor Networks[A]. Proc of the 5th annual ACM/IEEE international conference on mobile computing and networking[C]. Seattle, Washington, 1999. 263 - 270.
- [3] 李建中, 李金宝, 石胜飞. 传感器网络及其数据管理的概念、问题与进展[J]. 软件学报, 2003, 14(10): 1717 - 1727.
- [4] 任丰原, 黄海宁, 林闯. 无线传感器网络[J]. 软件学报, 2003, 14(7): 1282 - 1290.
- [5] PARK VD, CORSON MS. Temporally - ordered routing algorithm (TORA) version 1: Functional specification[S]. draft-ietf-manet-tora-spec-01.txt, 1998.
- [6] BROCH J, JOHNSON DB, MALTZ DA. The dynamic source routing protocol for mobile ad hoc networks[Z]. IETF Internet Draft,

1998.

- [7] Nasipuri A, DAS SR. On-Demand Multipath Routing for Mobile Ad Hoc Networks[A]. 8th International Conference on Computer Communications and Networks (IC3N 99)[C], 1999.
- [8] INTANAGONWIWAT C, GOVINDAN R, ESTRIN D. Directed diffusion: A scalable and robust communication paradigm for sensor networks[A]. Proc Sixth Annual International Conference on Mobile Computing and Network[C], 2000.
- [9] GANESAN D, GOVINDAN R, SHENKER S, et al. Highly-resilient, energy-efficient multipath routing in wireless sensor networks [J]. Mobile Computing and Communications Review, 2001, 4(5).
- [10] DULMAN S, NIEBERG T, WU J, et al. Trade-Off between Traffic Overhead and Reliability in Multipath Routing for Wireless Sensor Networks[Z]. WCNC Workshop, New Orleans, Louisiana, USA, 2003.
- [11] DULMAN S, WU J, HAVINGA P. An energy efficient multipath routing algorithm for wireless sensor networks[A]. IEEE International Symposium on Autonomous Decentralized Systems (ISADS 2003) [C]. Pisa, Italy, 2003.
- [12] YUKSEL M, PRADHAN R, KALYANARAMAN S. Trajectory - Based Forwarding Mechanisms for Ad-Hoc Sensor Networks[A]. IEEE 2nd Upstate New York Workshop on Sensor Networks[C]. Syracuse, NY, 2003.
- [13] PARKINSON B, SPILKER Jr JJ. Global Positioning System: Theory and Application[A]. Progress in Astronautics and Aeronautics[C], 1996, 163.
- [14] NICULESCU D, NATH B. Localized positioning in ad hoc network [A]. Sensor Network Protocols and Applications[C], 2003.
- [15] BRAGINSKY D, ESTRIN D. Rumor routing algorithm for sensor networks[A]. Proceedings of the 1st ACM international workshop on wireless sensor networks and applications[C], 2002.
- [16] WAN P-J, ALZOUBI KM, FRIEDER O. Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks[A]. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies[C]. 2002, 3. 1597 - 1604.

(上接第 500 页)

此外,破坏生成树的结构,使整个链路不仅在物理上,也在逻辑上连通。借鉴路由器中的方法,每个包都可以独立的选择自己的最佳路径,如果最短路径发生堵塞时,连接网段的桥能够自动选取相对最佳的路径供数据的传递。如果能够更加智能化,就可以实现长信息和短信息分别选路。

从长远的目光来看,随着以太网技术的不断成熟,人们为适应现场总线需要对以太网也作出各种改进,特别是 Profibus、Devicenet、Controlnet 和 Lonworks 等公司都在研究通过一种称作管道(tunnel)的简单传递机构,使用 Ethernet 网络传送报文。与此同时,美国电气工程师协会(IEEE)正着手制定现场装置与 Ethernet 通信的新标准。该标准能够使网络直接“看到”对象(object)。这些工作都为 Ethernet 进入工业自动化化的现场级打下了基础<sup>[4]</sup>。

在解决了网络的不确定性(Determinism)、环境适应能力、抗干扰能力以及自身的供电能力等问题之后,把以太网运用于工业总线中正成为当前一个比较热门的研究话题。

综上所述,我们的最终目标是在有成本限制的情况下,提高分布式系统互连的可伸缩性、递增可扩充性、可分割性以及可靠性与可修复性。

#### 参考文献:

- [1] TANENBAUM AS. 分布式操作系统[M]. 陆丽娜, 伍卫国, 刘

隆国, 等译. 北京: 电子工业出版社, 1999.

- [2] GALLI DL. 分布式操作系统——原理与实践[M]. 徐良贤, 唐英, 毛家菊, 等译. 北京: 机械工业出版社, 2002. 14 - 15.
- [3] CENA G, VALENZANO A. FastCAN: A High-Performance Enhanced CAN-Like Network[J]. IEEE Transactions on Industrial Electronics, 2000, 47(4): 951 - 963.
- [4] 周明. 现场总线控制[M]. 北京: 中国电力出版社, 2002.
- [5] ISO 11898, Road Vehicles-Interchange of Digital Information-Controller Area Network(CAN) for High-Speed Communication(First Edition)[S], 1993.
- [6] YOUNUS B, AHMAD B, BASHIR O, et al. A network protocol for distributed embedded systems[A]. Students Conference, ISCON'02 [C]. IEEE, 2002, 1. 149 - 153.
- [7] 张满怀. 嵌入式 TCP/IP 协议研究与实现[D]. 广东工业大学, 2001.
- [8] IEEE Std 802.1D-1990, IEEE standards for local and metropolitan area networks: media access control(MAC) bridges[S], 1990.
- [9] COMER DE. 用 TCP/IP 进行网际互联——第一卷: 原理、协议与结构[M]. 林瑶, 蒋慧, 杜蔚轩, 等译. 北京: 电子工业出版社, 2001. 233 - 234.
- [10] DUATO J, YALAMANCHILI S, NI L. 并行计算机互连网络技术[M]. 谢伦国, 张民选, 窦强, 等译. 北京: 电子工业出版社, 2004. 342 - 343.