

# 一种保证传感器网络实时服务的中间件机制<sup>\*</sup>

胡侃 刘云生 李坚

(华中科技大学计算机学院 武汉 430074)

**摘要** 传感器网络技术的发展导致了在应用与网络之间建立中间件,根据系统资源及网络能源保证在网络上并发运行的应用的 QoS。首先,本文结合传感器网络的特性及实时应用的特点,定义了一个事件驱动的保证实时 QoS 的传感器网络中间件框架。然后,定义了传感器网络中实时应用的执行模型,提出了一种基于时间戳相关的实现技术,在应用调度中结合应用语义保证实时 QoS、数据流的时间一致性及并发应用执行的逻辑一致性。最后,给出了基于时间戳模型的调度算法。通过性能测试结果分析表明,它能够显著提高系统的运行效率及对高速外部采集数据处理的实时性,特别适应于处理间歇、高速的传感器网络外部采集数据。

**关键词** 中间件,实时服务,数据流,无线传感器网络

## A Middleware for Providing Efficient Real-time Services in Sensor Networks

HU Kan LIU Yun-Sheng LI Jian

(Faculty of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

**Abstract** With the continuing advances in sensor networks and application design, appropriate middleware is needed to provide efficient QoS for coordinate concurrent applications on the sensor networks according the amount of resources and sensor energy. Usually, it is required that an instance of real-time application in sensor networks is periodically activated and must be completed before the next application instance is activated, and this results in data lost while acquiring rate is fast. In the paper, we first propose an event-triggered middleware framework and an executing model of real-time applications in sensor networks. And then, by using the timestamp of collection data to educe the timestamp of coordinate concurrent tasks and integrating application knowledge to tailor the design and implement of the middleware, an implement technique based on timestamp-reliance is presented to providing efficient data services and real-time response while guaranteeing time consistency and correctness of concurrent applications. Finally, we design an effective scheduling algorithms based on the model. Simulation test results show that by using the algorithms, the efficiency of data stream processing and real-time response can be enhanced, and the technique is suitable to process intermittent and high speed acquisition data in sensor networks.

**Keywords** Middleware, Real-time service, Data stream, Wireless sensor networks

无线传感器网络可以广泛地应用于国防军事、环境监测、交通管理、医疗卫生等领域,是近几年来国内外科学工作者的一个热门研究课题<sup>[1]</sup>。传统的传感器网络主要为简单的数据采集应用服务,一个网络通常支持一个应用。随着传感器网络技术的发展,在一个网络上将会有多个应用并发执行,这要求在应用和传感器网络之间建立中间件,根据计算、通信、传感器(CCS)资源和能源,保证系统的 QoS 得到满足<sup>[2,3]</sup>。这几年对中间件的研究已有很多成果<sup>[4~6]</sup>。在实时 QoS 方面,文[7]研制了一个实时网络协作和控制中间件,它在操作系统和应用程序员之间提供了一个称之为微单元的抽象接口,微单元能够完成自复制、迁移、分组操作。文[8]基于 CORBA 中间件,研究了将分布式应用的实时 QoS 自动传递的重量级中间件。文[9]提出了在嵌入式实时系统中的一种轻量级实现机制。

目前,传感器网络中的实时应用通常定义为一组相互通信的任务,应用实例(一组通信任务的一次并发执行过程)周

期性地被激活执行。为保证数据流处理的时间一致性,一个应用实例必须在下一个实例开始前执行完成<sup>[3]</sup>。这样当网络采集速率快,即采集数据速率高于系统处理能力时,采集数据将会被丢失造成网络能源消耗。

本文基于纯传感器网络框架和纯基站框架相结合的方式,给出了一个由事件驱动应用,系统的中间件根据应用需求通过路由协议驱动传感器网络采集数据,并将采集数据分配给相应应用的系统框架;然后将应用定义为由多个数据驱动的独立计算单元组成,这些计算单元之间可以相互传递数据流,从而协作地完成数据融合处理。系统通过将采集数据时间戳与计算单元时间戳相联系,再根据时间戳相关自适应地将一组独立计算单元组成一个语义逻辑完整的进程。这样在保证应用执行正确性的前提下允许多个应用实例并发执行,提高了系统处理高速采集数据流的能力;最后给出了基于时间戳的采集数据流分析处理的系统调度算法,算法中对采集数据的分配使用了复用方式保证多个并发应用的实时性。测

<sup>\*</sup> 本课题得到国家自然科学基金项目(60073045)、博士点基金及国防预研基金(00J15.3.3.JW0529)资助。胡侃 博士生,主要研究多元实时数据集成、移动数据管理及实时数据库;刘云生 教授,博士生导师,主要研究方向为现代(实时、主动、内存、移动等非传统)数据库理论与技术及其集成实现、数据库与信息系统开发、实时数据工程、软件方法学与工程技术;李坚 讲师,主要研究方向为实时数据库、仿真实验。

试分析表明,该方法在能源有限的传感器网络中能获得较好的实时 QoS 并提高了处理高速外部数据流的能力。

## 1 系统框架

### 1.1 基于传感器网络的框架

在这种实现方法中,应用对感知数据的查询由基站直接压入传感器网络中,网络通过路由协议将一簇传感器(一组空间相邻的监测目标对象的传感器)聚集查询结果传给基站,然后基站将数据分配给应用。为了解决多个并发应用同时查询导致传感器网络成为瓶颈资源问题,开发了查询聚集优化中间件,将多个并发查询合并为一个或几个查询<sup>[10]</sup>。这种框架的主要思想是通过应用来控制传感器工作模式,传感器按照应用需求采集数据,这样节约传感器能源,延长传感器网络的生命期。但是,由于通信时延,并不能很好地保证应用的实时要求。

### 1.2 基于基站的框架

在这种方法中,传感器网络不断地将聚集查询结果以数据流形式传给基站,基站将数据流保存在一种称之为概要数据结构<sup>[11]</sup>中保存起来,应用通过管理该数据结构的 DSMS 提供的 DML 使用采集数据。尽管这种框架能获得较好的实时性,但由于传感器网络不受应用控制,对于并发应用的查询需求系统无法按照传感器网络的位置特征进行聚集优化,且发回基站的数据大量丢失而造成传感器能源消耗。

### 1.3 基于事件驱动的系统框架

为保证系统的实时 QoS,我们给出一种事件驱动的基于感知查询数据时间戳的系统框架。如图 1 所示。

在实时应用环境中,大量的传感器分布在感知区域中监测感知对象的状态变化。为了节约能源,大部分传感器节点通常处以一种低能耗的睡眠状态,只有少数几个“哨兵”节点正常工作。一旦“哨兵”发现感知对象状态变化事件,它们将事件发送给系统事件管理器,事件管理器根据被这些事件激活的应用的数据需求,使用一种使得所有传感器节点能源消耗最小化的贪心策略<sup>[12]</sup>,唤醒一个或多个睡眠节点簇,这些节点簇使用传感器网络查询聚集算法,在一定的时间段内周期性的将结果通过簇的头节点传送到基站。同时“哨兵”节点将事件报告给应用管理器。应用管理器接受到事件报告,将根据事件类型驱动一个或多个应用在一个时间段内周期性的执行,对采集数据进行分析的处理,将结果外化给观察者。

的采集数据和导出数据,要保证一个应用实例是对一组时间一致的数据进行处理。

图 1 所给出的系统框架,由事件来控制传感器网络对数据的采集和应用的执行。这样不仅考虑了能源有限问题,也使得应用的执行模式与通常实时操作系统的事件驱动方式相同。

下面我们给出应用及传感器网络的模型。

## 2 应用及传感器网络模型

### 2.1 传感器网络模型

传感器网络定义为一个图,其中每个结点代表一个传感器,结点之间的边代表相连的两个传感器能够互相通信。整个网络监测一组变量的值(如温度,湿度等),令  $X = \{X_1, X_2, \dots, X_n\}$  是传感器网络监测的  $N$  个变量。传感器网络的工作模式是事件驱动的,即按照模式  $\{ON(\text{event 表达式}) \rightarrow SELECT X^+ FROM \text{sensors WHERE period} = t1, \text{duration} = t2\}$  唤醒一个或多个簇执行聚集查询。它表示当有事件发生时,  $X$  的一个子集  $X^+$  中每个变量的值将被传感器网络在一个时间段  $t2$  内以周期  $t1$  周期性的传递到基站,  $\forall x_i \in X^+$  是带有时间戳的数据流,该时间戳是监测该变量的一簇传感器聚集一个数据值时每个传感器时间的平均值,反映了监测对象在某时刻的状态值。

本文中,我们对传感器网络做了下列假定:(1)所有传感器具有统一的时钟;(2)传感器网络支持基于时间戳的路由模式,一簇传感器聚集的数据值带有一个时间戳;(3)应用管理器可以看作是一个具有全局基站视图的中间虚拟机。

### 2.2 应用模型

应用的执行是事件驱动的,即按照模式  $\{ON(\text{event 表达式}) \rightarrow ACTIVE A WITH \text{duration} = t\}$  控制应用执行。它表示当 event 表达式为 TRUE 值时,应用管理器将应用 A 在时间段内  $T$  内置为激活状态,任务调度器再根据传感器网络传入基站的数据流周期地驱动处于激活状态的 A 执行。每次执行过程称为一个应用实例。

一个应用定义为一个无环有向图  $G = (V, E)$ 。  $V$  是结点的集合,每个结点代表组成应用的一个结点函数,一个结点函数的一次执行(结点实例)将一组带有时间戳的数据转换为另一组带有时间戳的导出数据。一个结点函数定义为三元组  $F = (T_1, P, T_0)$ 。  $T_1 = (I_1, I_2, \dots, I_n)$  称为输入模版,它的一个具体取值  $D_1 = (d_1, d_2, \dots, d_n)$  称为一个输入模版实例,每个  $d_i$  都是带有时间戳的数据。在一个应用被激活的时间段内,一个结点的输入模版中一旦产生一个实例,将触发结点函数的程序体  $P$  执行,它完成了数据转换  $P(T_1) \rightarrow T_0$ 。  $T_0 = (O_1, O_2, \dots, O_m)$  称为导出模版,它的一个具体取值  $D_0 = (v_1, v_2, \dots, v_m)$  称为一个导出实例。每条有向边  $(U, V, \{O_i \rightarrow I_j\})$  表示从结点  $U$  的导出模版到结点  $V$  的输入模版的数据流,  $\{O_i \rightarrow I_j\}$  表示具体数据流值。

按照上述定义,一个应用的所有结点被分为三个层次:采集层,处理层,外化层。  $G$  中所有只有出边没有进边的结点函数组成应用的采集层,采集层结点的输入模版中数据来自于传感器网络监测变量  $X = \{X_1, X_2, \dots, X_n\}$ 。所有只有进边没有出边的结点函数组成应用的外化层,外化层结点的导出模版中数据输出给系统观察者。所有既有进边又有出边的结点函数组成应用的处理层,处理层结点完成采集数据的分析处理,它将一种语义抽象层的采集数据转换成另一个抽象层的

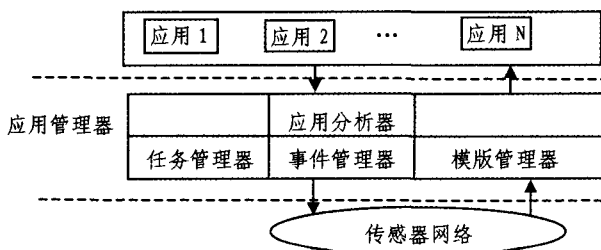


图 1 基于事件的系统框架

应用程序事先由程序员通过应用管理器提供的接口定义。应用分析器对应用定义进行分析,根据基站分布及应用函数处理的采集数据类型将应用的子函数程序分布在不同的机器中。事件管理器负责管理在事件发生时激活应用。任务管理器负责对一个激活的应用,当其需要处理的采集数据到达时实时的调度执行。模版管理器管理应用执行过程中处理

采集数据。如一个结点将监测区域中不同区域的温度值转换成全局温度值。一个应用是由一组数据驱动的独立执行的结点函数协作实现的。事件发生时,系统将应用激活一个时间段,而应用的每个结点函数是由输入模版实例数据驱动并发调度独立执行的,它们的执行并不是由应用显式地直接控制的。在应用被激活的时间段内,由于传感器网络周期性的将采集数据传入基站,多个协作结点函数可以被多次驱动并发执行,在同一个时刻,系统中存在着多个并发执行的结点函数,而每个并发执行的结点函数可能对应于同一个应用的不同实例。图2是应用的描述图。

在上述应用模型中,我们做了如下假定:(1)在一个时间段内,可以有多个不同的应用被激活。一个被激活应用在激活期内多次并发执行。(2)在一个时间段内,同一个应用的结点函数被数据周期地驱动且与其它结点并发调度执行。(3)传感器网络采集数据周期基本相同。

### 2.3 需解决的问题

基于上述模型,存在着下列几个问题需要解决:

(1)数据时间一致性问题。一个应用是对多个传感器网络监测变量进行分析处理的过程,这些变量是网络中不同的簇聚集的带有时间戳的数据。受网络通信时延及无线传感器网络通信断接的影响,它们发送回基站的时间是不定的。系统应保证一个应用实例(应用的一次执行)及组成该应用的所有协作结点函数实例是对监测对象在同一个采集周期的采集数据(或导出数据)进行处理的过程。

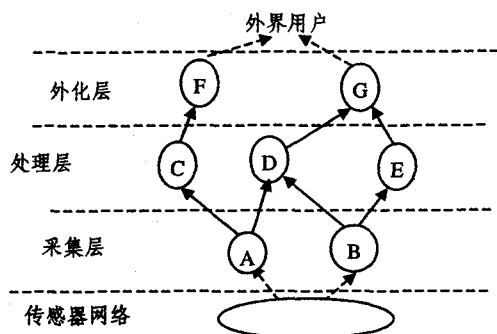


图2 应用描述图

(2)应用执行的语义逻辑一致性问题。组成应用的每个结点函数在应用被激活的时间段内可以有多个实例,每个结点实例是独立的由数据驱动的。传感器网络的断接性及通信时延不确定,这使得一个应用实例只有部分结点函数运行,而其它的函数结点没有执行,如图2中只有A、C、F执行,B结点由于得不到该时刻的数据使得D、E、G均不执行。若允许多个应用实例并发执行,则系统中某时刻可能并发存在多个结点函数实例,它们在语义逻辑上属于同一个应用的不同实例,即属于同一个应用对监测对象不同周期时间状态的不同的执行过程。由于数据独立驱动的结点调度方式不同于通常的由应用直接控制结点执行的调度方式(由进程直接控制线程)。因而,系统应根据应用语义能够自适应地正确识别出一组独立的协作结点实例(看作线程)是一个应用实例(进程)的子线程,即系统能够自动地将一组并发执行的结点线程(它们应是对监测对象在同一个采集周期内的处理执行过程)识别为一个进程。另外系统还应保证应用执行过程中的外化数据时间次序是按照被处理的采集数据时间次序输出的,不能出现后采集的数据处理结果被先外化现象。

(3)采集数据的复用问题。由于系统允许多个不同应用并发地被激活,传感器网络监测变量集X中的数据有可能被多个不同应用共享使用。因此,系统应使得被网络传递到基站的数据能复式地分配到多个当前被激活应用的采集层结点的输入模版实例中,以提高并发应用的实时性。

下面我们给出解决这几个问题的具体方法。

## 3 基于时间的实现技术

为了说明实现机制,我们先给出时间相关的定义,令 $t()$ 是求时间戳的函数。

定义1 令 $D=\{d_1, d_2, \dots, d_n\}$ 是由 $N$ 个元素组成的集合。对于一个给定的阈值 $V$ (小于传感器网络的采集周期),若 $\forall d_i, d_j \in D (i \neq j, 1 \leq i, j \leq n)$ 有 $|t(d_i) - t(d_j)| < V$ 成立,则称元素 $d_i$ 与 $d_j$ 是时间相关的,且 $D$ 是一个时间相关集。

### 3.1 基于时间技术的基本思想

基于时间技术的主要思路是对系统中每个结点函数的一次执行(结点实例)由系统自动地赋予一个时间戳,该时间戳与被该结点实例分析处理的采集数据集的时间戳相互一致,且所产生的导出数据具有与该结点实例相同的时间戳。我们按照均值公式定义结点实例的时间戳:令 $P_i$ 代表结点 $P$ 的某个实例,且 $P_i$ 是由输入模版实例 $D_i=(d_1, d_2, \dots, d_n)$ 驱动执行的,则 $t(P_i)=1/n \sum_{i=1}^n t(d_i)$ ,且 $P_i$ 产生的导出模版实例中每个数据的时间戳均为 $t(P_i)$ 。这样,我们通过将数据时间转换成被执行的结点函数的时间,从而将应用的语义与系统中调度执行的结点实例的时间戳相联系,使得系统能够根据结点实例的时间戳,自动地将一组独立调度执行的结点实例依据它们的时间戳相关组成为应用的一次执行(一个应用实例)。因为从应用语义逻辑来看,这组时间相关的结点实例是对监测对象在同一个时刻(准确地讲,是传感器网络在同一个采集周期内)的状态进行分析处理的一个完整过程。定义1给出的时间相关概念实质上是传感器网络中一个周期时间段转换为系统中的一个时刻,这种处理方式与传感器网络中由传感器周期地采集数据(每个簇在一个周期聚集数据的时间戳不完全相等),系统的应用周期地处理这些数据的工作模式相适应的。传统的实时系统中,为了保证时间一致性,要求一个应用的每次执行一定要在下一个周期之前实时完成。由于应用的复杂性,要求调整传感器的数据采集周期以适应应用的复杂性要求,降低了系统的实时QoS。基于时间相关的技术,允许应用的多次周期执行并发执行,系统根据网络数据采集周期自动地将系统中处理同一个周期的采集数据的结点函数实例组成一个完整的应用实例。这样由应用去适应实际的传感器监测网络,而不是由传感器网络适应应用,提高了系统的实时QoS。

在设计实现中,应用管理器遵照下面规则解决上述一致性问题:

(1)时间一致性及数据复用。系统保证当前被激活应用的模版中每个输入模版实例中数据集是一个时间相关集合,一个结点函数是由一组时间相关的采集数据驱动的。当有传感器网络监测变量的数据值到达时,模版管理器查询所有当前被激活的每个应用的采集层结点的输入模版(考虑了采集数据复用),按照时间相关限制将数据插入相应的模版实例中,若形成一个完整的时间相关的模版实例,则以该模版实例为参数转任务管理器调度相应的结点函数执行。当一个结点

函数实例执行完时,若该结点是外化层结点,则将导出模版输出给外界;否则,任务管理器将其导出模版实例传递给模版管理器,模版管理器按照边的导向将数据按照时间相关限制插入对应结点函数的输入模版实例中,若形成一个完整的模版实例,则转任务调度器。

(2)语义逻辑一致性。为了保证应用实例并发执行的正确性及同一个应用的多个并发实例执行的有序性(不出现优先级倒置)。系统自动地将所有时间戳相关的结点实例作为一个进程处理,同一个进程的所有结点实例作为该进程的子线程处理。对结点实例的调度,采用传统的基于服务器的实时调度框架。同一个结点的所有实例在一个服务器(占有一定比例的虚拟处理机)上实时调度,每个结点配置一个等待队列,只有在前一个实例执行完后,再从队列中选取一个优先级最高的等待实例执行。结点实例的优先级是按照结点实例的时间戳,较早时间戳的实例具有较高的优先级来选定的。系统将当前并发执行的所有时间相关的结点实例组成一个应用实例,当一个新的采集周期数据到达并驱动对应于该周期的应用实例的第一个采集层结点实例执行时,系统创建一个新的应用实例进程,并将该结点实例作为应用实例的一个线程。对其它的所有处理该周期内采集数据的结点实例,系统通过时间相关自动地把它们组成该应用实例的线程。这样系统不仅能够将一组独立并发执行的结点实例按照网络采集周期自动地组成对应于不同周期的应用实例,而且保证了先一个周期的应用实例先结束不出现优先级倒置现象。图3描述了系统自动的将各个周期的结点实例按照时间相关划分为不同的应用实例。Y轴的 $S_i$ 是结点函数,由于传感器网络采集数据的周期性,因此每个数据驱动的结点函数实例按照其时间戳可以看作是一个周期执行序列(X轴所示)。若传感器网络的断接性使得某个周期的部分数据未发送到基站(如图2中B结点未被驱动),我们将对应该周期未被驱动的结点实例看作是一个空实例,而对应于该周期的应用实例将最终因超截止期而终止。基于时间技术的另一个隐含优点是在网络发生断接时系统不会将不同采集周期的数据由一个应用实例处理,从而保证了应用外化给系统观察者信息的准确性。

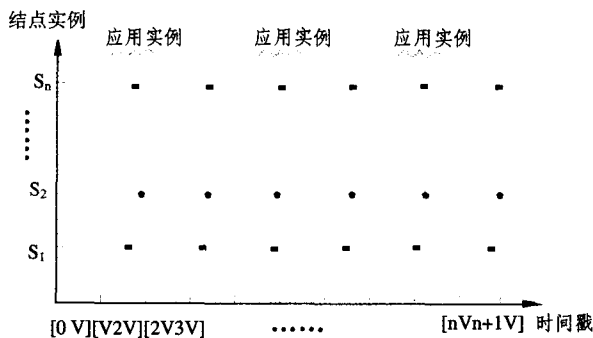


图3 时间相关的应用实例控制区域

### 3.2 算法描述

#### (1)模版管理器

模版管理器主要由输入模版匹配函数 INSERT-TEMPLAT()及导出数据传递函数 TRANSFER-DATA()组成。设L是当前被事件激活的所有应用队列。X是传感器网络传递到基站的监测变量集,  $\forall x_i \in X$  定义为一个二元组  $(V, t)$ , V是  $x_i$  值, t是  $x_i$  的时间戳。Y是应用实例的一个结点实例执行完时产生的导出模版实例,每个元素也是二元组。

算法描述如下:

```

INSERT-TEMPLAT()
{ FOR L 中的每个应用 Li DO
  { FOR Li 的所有采集层模版 DO
    { 将 X 的每个元素插入到一个对应的时间相关的模版实例中;
      IF 没有时间相关实例
        THEN 产生一个新的实例;
      IF 产生一个完整的模版实例
        THEN 计算模版实例元素的时间戳均值;
           将该模版实例及其时间戳按照时间戳优先插入对应
           结点的作业等待队列;
    };
  };
};

TRANSFER-DATA()
{ IF Y 是由一个外化层结点实例产生的导出实例
  THEN 调用外化函数外化结果;
  ELSE { 将 Y 的每个元素按照边的导向插入到对应结点的时间
    相关的输入模版实例中;
    IF 没有时间相关实例
      THEN 产生一个新的实例;
    IF 产生一个完整的模版实例
      THEN 计算模版实例元素的时间戳均值;
         将该模版实例及其时间戳按照时间戳优先插入对应
         结点的作业等待队列;
    };
};

```

#### (2)任务管理器

任务管理器主要由结点调度函数 SCHEDULE-VERTEX()和结点实例完成函数 FINISH-VERTEX()组成。每个结点在一个服务器(占有一定CPU比例的虚拟机)上使用基于服务器的实时调度算法调度。

```

SCHEDULE-VERTEX()
{ 从等待队列中选队头元素;
  创建一个结点实例;
  按照时间相关将该实例插入到对应的应用实例中作为其线程;
  IF 没有时间相关的应用实例
    THEN 创建一个新的应用实例并启动应用实例定时器;
  转实时调度;
};

```

```

FINISH-VERTEX()
{ IF 是一个外化层结点实例结束
  THEN { 检查父应用实例的所有儿子结点实例是否结束;
    IF 都结束
      THEN 结束该应用实例;EXIT();
      ELSE 在该应用实例中标记该结点实例结束 5
    };
  ELSE 在其父应用实例中标记该结点实例结束;
  调用 TRANSFER-DATA();
};

```

当一个应用实例的定时器到时,终止该应用及其子结点实例。

#### (3)事件管理器

事件管理器主要由事件处理函数 EVENT-P()组成。

```

EVENT-P()
{ FOR 所有被该事件驱动的应用 DO
  { 根据应用输入层模版需求及网络拓扑构造簇;
    将存在相交的簇查询优化合并;
    根据网络中传感器的能耗情况选取最优的传感器作为簇头节点;
    激活簇并将查询压入网内
  };
};

```

## 4 性能评测

### 4.1 相关模型对比

数据流程图程序模型由于其结构简单、语义清晰而被广泛地应用在实时监控领域。传统的纯数据流程图的调度,因为应用在简单的实时环境中没有考虑并发调度问题。当采集层结点的数据都到达时,系统按照下列步骤调度程序执行:①调度当前所有被数据激活的结点函数同步执行;②按照边传递所有的数据;③重复上述过程直到程序结束。在简单的实时应用环境中,可以认为采集层接受的采集数据是与外部对象时间一致的,系统能够保证简单的应用程序在一个周期内完成。在复杂的传感器网络环境中,应用程序需要的采集层

数据往往被传入不同的基站,由于无线通信的断接性及通信时延的不定性,不能保证采集层数据是与外部对象时间一致的。并且由于应用的复杂性,系统应允许同一个应用的不同周期执行是并发的。因此纯数据流程图模型不能很好的适应复杂传感器网络环境。

本文提出的系统框架综合了基于传感器网络框架及基于基站的框架。尽管传感器网络与应用是独立并行工作的,但它们都是由事件驱动的,系统根据事件以及被事件激活的应用对采集数据的需求来优化地控制传感器工作,传感器一旦工作将独立于应用执行。这样有效地控制了传感器能源消耗。应用程序的调度执行采用了一种基于时间相关的并发数据流程图模型。系统将采集数据时间与程序执行相结合,从而将应用语义与调度相结合。系统根据应用语义自动地将异步并发执行的结点函数组成一个应用的动态执行过程,保证了应用处理数据的外部时间一致性。通过时间优先级调度保证应用并发执行的优先级次序。这种机制适宜事件驱动的实时应用环境,与通常的事件驱动实时操作系统能有效地结合。

#### 4.2 模拟测试

基于事件的系统框架实质上是一种根据应用需求由应用控制传感器网络的方式,这种方式与只使用电压控制对传感器网络生命期的对比测试在文[3]中已有描述,在此不再赘述。我们在由国家自然科学基金资助的自行研制的嵌入式实时操作系统上分别使用纯数据流调度与基于时间相关的数据流调度完成了实时性能测试,PC机带有计时器,计时精度可以达到1 $\mu$ s。测试应用例程为两个采集层结点从外部读入字符数据;转换结点来自来自串口的数据进行一定的变换,结果发送给分配结点;分配结点将数据分两路输出,一路给图形显示结点,一路给文本显示结点。我们从两个方面测试系统的实时性,一是在系统任务繁重(调整应用的执行周期)下对外部数据的实时响应,测试指标为应用的超截止期比率MR=超截止期应用实例数量/应用实例总数\*100%;二是在外部数据高于系统数据处理时间下系统对高速数据的接受能力,测试指标为外化率OR=外化数据数量/采集数据数量\*100%。表1是主要测试参数。图4、图5是测试结果。

表1 模拟参数

Parameter	Value	Parameter	Value
Think time	5~10s	Schedule policy of CPU	HPF
Number of nodes	6	Replacement policy of memory	LRU
Number of sensors	2	Average communication time	316 $\mu$ s
Number of fixed hosts	8	Average executing time	8720 $\mu$ s
Type of task	Soft real-time	Data send period	5~50ms

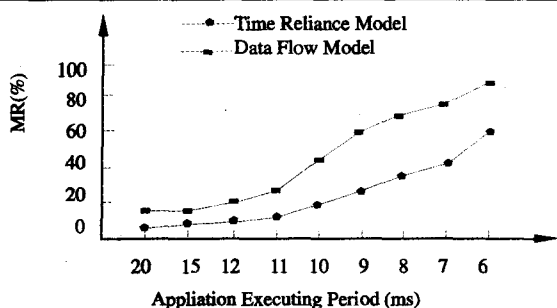


图4 应用实例的不同周期对MR的影响

在第一种测试情况下,在应用的执行周期大于11ms时,纯数据流调度与基于时间的数据流调度的MR值相差不大。

因为这时应用实例的执行基本上是串行的。在执行周期小于11ms时,纯数据流调度的MR值急剧增加,尽管从理论上讲在11ms~9ms之间纯数据流不应出现这样的情况,但测试结果表明,操作系统内部实现对系统资源的占用导致MR值的急剧变化。

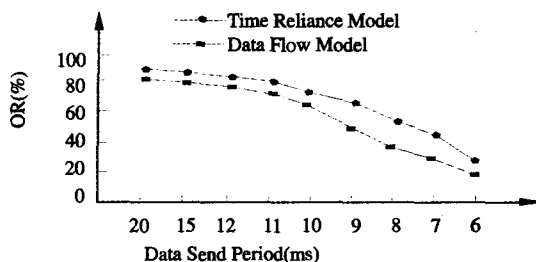


图5 采集数据速度对OR的影响

在第二种测试情况下,基于时间的数据流调度的外化率要高于纯数据流调度,其外化速度在慢速采集时有一定缺点(实现机制引入了一部分处理时间)但高速时要比纯数据流快。但在6ms后,基于时间的调度方法不仅出现了超截止期增加,而且外化率也下降,这是由于系统资源有限造成的,通过换一台配置更高的计算机发现这种现象有所改变。

结束语 传感器网络技术的发展导致了保证并发应用QoS的中间件研究。本文结合传感器网络的特性及实时应用的特点,定义了一个保证实时QoS的传感器网络中间件框架;提出了基于该框架的应用执行模型;描述了一种基于时间相关的数据流处理实现技术;对算法进行了测试,证明其能够显著提高系统的运行效率及对高速外部采集数据处理的实时性。该技术特别适应于处理间歇、高速的传感器网络外部采集数据。

#### 参考文献

- 1 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. 软件学报,2003,14(10):1717~1727
- 2 Romer K, Kasten O, Mattern F. Middleware challenges for wireless sensor networks. ACM SIGMOBILE Mobile Communication and Communication Review, 2002,6(2)
- 3 Yu Y, Krishnamachari B, Prasanna V K. Issues in designing middleware for wireless sensor networks. IEEE Network Magazine, 2003
- 4 Yu X, Niyogi K, Mehrotra S, et al. Adaptive middleware for distributed sensor networks. IEEE Distributed Systems Online, May 2003
- 5 Murphy A, Heinzelman W. MiLAN: Middleware linking applications and networks:[Tech Rep]. TR-795. University of Rochester, 2002
- 6 Cougar Project. [Online]. Available: <http://www.cs.cornell.edu/database/cougar>
- 7 A Network Virtual Machine for Real-Time Coordination Services. [Online]. Available: <http://www.cs.virginia.edu/nest>
- 8 Pyarali I, Schmidt D C, Cytron R. Techniques for enhancing real-time corba quality of service. In: IEEE Proceedings, May 2003
- 9 ORBACUS website. [Online]. Available: <http://www.orbacus.com>
- 10 Yu W, Le T N, Xuan D, et al. Query aggregation for providing efficient data services in sensor networks. In: IEEE International Conference on Mobile Ad-hoc and Sensor Networks, 2004
- 11 Babcock B, Babu S, Datar M, et al. Models and issues in data streams. In: Pappa L. ed. Proc. of the 21<sup>st</sup> ACM SIGACT-SIGMOD-SIGART Symp on Principles of Database Systems. Madison; ACM Press, 2002. 1~16
- 12 Yu Y, Prasanna V K. Energy-balanced task allocation for collaborative processing in wireless sensor networks. Accepted by MONET special issue on Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks