一种基于蚁群优化的 WSN 拥塞控制算法*

余小华[1]，黄灿辉[2]
（1. 华南理工大学广州学院 计算机工程系，广州 510800；2. 华南师范大学增城学院，广州 511363）

摘 要：针对无线传感器网络中由于拥塞引起的丢包和能量过度消耗等问题，提出了一种基于蚁群优化的拥塞控制算法以减轻 WSN 中的拥塞和改进网络性能。该算法充分考虑了给定时刻 WSN 的拥塞状况，分成三个阶段在源节点和 sink 节点间寻找一条最佳的路径，并及时地消除拥塞。仿真实验结果表明，该算法在网络吞吐量、丢包率、时延和能耗方面具有较好的综合网络性能。

关键词：无线传感器网络；蚁群优化；拥塞控制；丢包率；能耗
中图分类号：TP393；TP301.6 **文献标志码**：A **文章编号**：1001-3695(2012)04-1525-04
doi：10.3969/j.issn.1001-3695.2012.04.090

Congestion control algorithm based on ant colony optimization in wireless sensor networks

YU Xiao-hua[1], HUANG Can-hui[2]
(1. *Dept. of Computer Engineering, Guangzhou College of South China University of Technology, Guangzhou* 510800, *China*; 2. *Zengcheng College of South China Normal University, Guangzhou* 511363, *China*)

Abstract: In view of packet loss and excessive energy consumption due to congestion in wireless sensor networks, this paper proposed a congestion control algorithm based on ant colony optimization to reduce congestion scale and improve network performance in WSN. The algorithm was divided into three stages in the given time, and fully considering the congestion status of WSN from source node to sink. The algorithm could timely eliminate congestion and find a best path. Simulation results show that the algorithm can achieve good network performance about network throughput, packet loss, delay and energy consumption.

Key words: wireless sensor networks; ant colony optimization; congestion control; packet loss ratio; energy consumption

无线传感器网络是一种具有很好应用前景的技术，可以被广泛运用于军事、农业、工业、商业等领域。尤其是一些人类难以涉及或危险的领域，比如能够帮助人类执行诸如环境监测、动植物监测、健康护理、交通控制、灾难拯救等任务[1]。WSN 一般包括一个或多个基站（sink 节点）和大量小型的传感器节点，这些节点能够采集、处理和转发数据[1,2]。节点具有体积小、能耗低等特点，但也具有能量有限、计算能力有限、带宽有限以及容易受故障影响等缺点[3,4]，因此能量和安全管理等成为 WSN 的研究热点。除此之外，还有其他很重要的研究热点，如拥塞，目前大量针对 WSN 拥塞的研究工作已经展开，这将在下文进行介绍。当传感器节点检测到周围物理现象出现异常时，它们将向 sink 节点发送数据，这时候可能出现拥塞。一旦出现拥塞，WSN 将会出现吞吐量降低、丢包频繁、传输率降低、出错率高、能耗增加等现象，进而影响到 WSN 的正常运转。因此，必须要有有效的方法来处理 WSN 的拥塞，以使得 WSN 能在各种情况下都能满足运转的要求。

WSN 中的拥塞可以分为节点拥塞和链路拥塞两种情况[5]。节点拥塞指的是由于节点溢出造成的，由于大量的节点向同一节点发送数据而该节点又不能及时接收、处理和转发数据而造成丢包和重传，进一步导致额外的能耗；链路拥塞是指多个节点共享同一无线信道向同一节点发送数据可能导致的冲突。这两种拥塞都会增加数据的服务时间并减少总的吞吐量。拥塞控制要求高能效以提高服务质量（QoS），并提供加权吞吐量和公平性以保证传感器节点能够适应不同的传感器，并能适应于不同的优先级水平。

本文借助 PCCP[4] 中的有关定义，给出基于蚁群优化的拥塞控制算法的描述。该算法分为三个阶段：a）初始化，这是算法的主要阶段，可以唤醒其他功能；b）蚁群优化，这是算法的核心部分，主要是在节点和 sink 节点上产生蚂蚁代理，并启动消除拥塞过程；c）拥塞消除，借助 PCCP 拥塞消除方法并对其进行改进。

1 相关研究

自从提出蚁群优化算法后，其已经被运用在好几个研究领域，主要用来对网络进行优化。其中基于蚁群优化的网络协议或算法广泛使用在有线和无线网络的路由协议中。拥塞控制是 WSN 中最重要的网络控制活动之一，包括预防、发现和消除网络拥塞。

蚁群优化起源于蚂蚁为觅寻食物而发现最佳路径。首先蚂蚁在旅行过程中随机选择路径以到达食物所在地，并在沿途

收稿日期：2011-07-31；**修回日期**：2011-09-05 **基金项目**：广东省科技计划项目（2007b010400068）；省部产学研结合项目（2011B090400085）

作者简介：余小华（1976-），男，江西永新人，博士研究生，主要研究方向为计算机网络、无线网络和网络安全（yu_x_h@21cn.com）；黄灿辉（1974-），女，讲师，主要研究方向为无线多媒体传感器网络.

释放信息素以标志该条路径。较短的路径意味着有更多的蚂蚁释放了信息素从而鼓励更多的蚂蚁沿着同一条路径去搬运食物。在文献[6]中提出了 AntNet 协议,由类似于蚂蚁的代理来选择可靠的最佳路径。当路径完全建立之后蚂蚁才完成信息素释放的工作。另外,该方案中使用本地启发式信息在每个时间间隔内合并网络流量。AntNet 比一般的基于蚁群优化技术表现出色,原因在于它有一个测量源速率累积的选项,能够在遇到拥塞的时候执行。PCCP 是减缓拥塞最流行的协议之一,它使用了跨层优化并在 MAC 层和网络层之间引入了一个调度器。调度器基于拥塞程度对临时队列的数据包进行动态调度。另外,PCCP 也考虑到节点的优先级,并基于加权吞吐量和公平性的概念,所以节点可以根据它们的优先级和重要性获取不同的带宽,并添加到数据包的有关字段里面。文献[7]采取了一种非常消极的态度,即一旦在某条线路上检测到拥塞就转换到另外一条路径传输数据,这样频繁地转换会降低网络吞吐量和增加重传的次数。而在 CCACO 中,首先考虑的是怎样解决拥塞,然后再考虑其他选择。

2 网络模型和多路径路由

采用以 sink 节点为根的树型无线传感器网络结构,大量的小型传感器节点均匀分布在网络中,所有的传感器节点具有同样的传输和感知范围。另外,这些传感器节点具有有限的处理能力、功率、内存和能量。然而 sink 节点具有强大的资源可以执行任何一个任务。假定不同节点传输的数据包大小是一样的,节点的缓存大小是节点能够保存最多数据包的大小,在每个节点的 MAC 层上执行 CSMA/CA 协议。

路由协议采用多路径数据转发,因此,每个源节点需要使用一种可靠的多路径路由算法来建立多条路径到达目的地。选择建立六条可相互替换的路径,称为 P_1,P_2,P_3,P_4,P_5 和 P_6,如图 1 所示。源节点同时使用 P_1 和 P_2 来分流流量负载,当其中一条出现故障时可以切换到其他路径,这样减少了路由重建的开销。

3 基于蚁群优化的拥塞控制(CCACO)算法

基于以下几个因素的考虑,提出了基于蚁群优化的 WSN 拥塞控制算法,以提高 WSN 的吞吐量,降低丢包率、时延和能耗,延长 WSN 的生命周期。

a)在无线传感器网络中,传感器节点由于其重要性、功能性、参与程度及位置不同,通常具有不同的优先权,因此拥塞控制协议需要加权公平性,以使得 sink 节点能够从加权公平性方式运行的传感器节点上获取一定的吞吐量。

b)参与数据通信的节点数在某个给定的瞬间可能会发生变化,因此由 ACO 预先确定的路径也许不是非常合适的。由拥塞控制带来的改进也许会导致之前被声明为在最小网络流量和高峰时刻最有效的路径,成为一条单路径路由。

c)蚁群优化路由协议需要支持服务质量 QoS,即吞吐量、时延和能效等。

d)考虑到传感器节点能量有限,要在每个节点上维持路由表会增加额外的开销。在 CCACO 算法里面能够减少这些开销。

CCACO 利用基于 ACO 方法的信息素来检测路径,无论在距离还是能耗方面都要求是最佳的。但是随着信息素浓度升高,该路径将被使用得更多,最后可能导致拥塞的出现。CCACO 基于数据到达时间和服务时间(即穿越该节点的时间)在节点上检测拥塞,假如该时间超过某个设定好的阈值,则表示出现了拥塞并在节点上进行标记,通知下面的子节点。CCACO 是通过提炼 ACO 和 PCCP 两种方法的优势而提出来的,怎样融合两种工作到 CCACO 里面是一项非常具有挑战性的工作,充分发挥出 ACO 的本质属性和 PCCP 的消除拥塞方法。

3.1 定义

下面就算法中涉及到的一些术语和定义作说明,有些是采纳 PCCP 里面的工作。

1)优先级 不同节点担任不同的角色,sink 节点要从节点上获取不同的信息。用 $SP(i)$ 表示节点 i 产生的数据优先级。

2)智能拥塞检测(intelligent congestion detection,ICD) 使用 ICD 测量拥塞程度,数据到达时间 t_a^i 和数据服务时间 t_s^i(从到达 MAC 层到数据成功传输出去的时间,如图 2 所示)用来确定 WSN 中的拥塞程度,节点 i 上的拥塞程度表示为

$$d(i) = \frac{t_s^i}{t_a^i} \tag{1}$$

当 $d(i)$ 的值大于 1 时,表明节点 i 出现了拥塞。拥塞程度代表了 WSN 的当前运行状态,可以将 $d(i)$ 通知给子节点以调整它们的发送速率。正如 PCCP 中所描述的那样,本文也使用指数加权移动平均公式来估计 t_a^i 的值。在任意时刻,当有 N 个数据包到达节点 i,t_a^i 更新为

$$t_a^i = (1 - w_a) \times t_a^i + w_a \times T_N / N \tag{2}$$

其中,$0 < w_a < 1$ 是一个常量,T_N 是测量出的时间间隔。类似地,当每次数据包转发时 t_s^i 更新为

$$t_s^i = (1 - w_s) \times t_s^i + w_s \times T' \tag{3}$$

三个参数 $d(i)$、t_s^i 和 t_a^i 以及对应的更新技术和公式都来自于 PCCP。

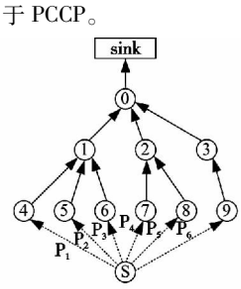

图1 网络模型和多路径路由

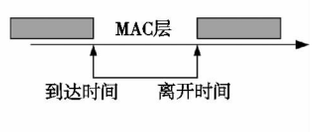

图2 数据包服务时间t_s^i

3.2 CCACO 算法

通过前面的论述,采用如图 1 所示的拓扑结构,算法给出采用模块结构并进行详细的描述,一些思想来自于 PCCP 和蚁群优化。首先,提出了初始化函数,该函数是算法的主要部分,同时会触发其他功能函数。CCACO 也参考了文献[8]中蚁群优化对一般优化问题的解决方法。

3.2.1 *初始阶段*

初始化函数是用来在指定的仿真时间内形成一个随机网络拓扑结构,主要是根据区域的大小节点数、优先权以及分配给节点的其他特点来进行构建,并生成节点的坐标位置,给每个节点设置随机的能量值,各邻接边也赋予随机的带宽、延迟、抖动和信息素。

a)找出需要发送数据到 sink 的所有节点并保存到数组 $T[\max]$ 中(T 表示发送);

b)假如时间少于网络预定义时间,对于数组 $T[\max]$ 中每个 $T[i]$

进行步骤 c)的操作,否则跳到步骤 d);

c)启动蚁群优化过程 ACO($T[i]$),详情如 3.2.2 节所示,回到步骤 b);

d)将节点 i 归入到不同路径上;

e)直到所有需要发送数据的节点归结完毕,否则返回步骤 a)。

3.2.2　蚁群优化阶段

1)蚁群优化过程 ACO($T[i]$)

a) 过程开始;

b) 启动节点上的调度;

c) 在节点 $T(i)$ 上产生蚂蚁代理,调用过程 GAnodes($T(i)$);

d) 在 sink 节点上产生蚂蚁代理,调用过程 GAsink(sink);

e) 若有中断产生结束调度,否则回到步骤 b);

f) 过程结束。

2)GAnodes($T(i)$)过程

a) 过程开始;

b) 假设能量未耗尽,调用 ant()生成新的蚂蚁代理(即 new ant()),否则结束;

c) 过程结束。

3)GAsink(sink)过程

a) 过程开始;

b) 假设能量未耗尽,调用 ant()生成新的蚂蚁代理(即 new ant()),否则跳到步骤 d);

c) 假如存在拥塞,更新子节点 Update_child($t_s(T(i))$,$t_a(T(i))$,$SP(T(i))$,$O(T(i))$);

d) 过程结束。

4)new ant()过程

a) 初始化 ant();

b) 更新节点的存储,即缓冲区大小;

c) 当未达到最终状态时,更新本地蚁群路由表,否则跳到 j);

d) 仿真一个有限状态机,并根据当前状态确定下一状态和概率参数;

e) 假如监测到拥塞,将数据包中拥塞标记设为 1;

f) 假如拥塞标记为 1,则执行拥塞删除过程 Congestion_remove($T[i]$,$t_s(p_i)$,$t_a(p_i)$,$SP(p_i)$,$O(p_i)$),信息素进行挥发;

g) 转移到下一条路径,并在访问的路径上放置信息素;

h) 更新蚁群路由表;

i) 更新蚂蚁内部状态;

j) 对方案进行评价;

k) 蚂蚁结束活动;

l) 过程结束。

作为算法的一部分,这个过程将一直执行,直到预设的仿真终止条件结束为止。普通节点或 sink 上蚂蚁代理的产生和移动是同时启动的。节点上生成蚂蚁代理的过程就是产生一个虚拟的蚂蚁发送到网络中一个给定的节点,并将该蚂蚁作为一个参数。同时也在 sink 节点上启动另外一个 new ant(),一直循环下去直到最终状态。随着本地蚁群路由表的创建,某些限制如节点数、优先级等用来计算蚂蚁跳到下一个状态的概率。

完成一跳之后,一个预定义的信息素值被放置在路径上。然而,在此之前,蚂蚁要求检测将要离开的节点是否出现了拥塞。如果出现拥塞,则调用拥塞删除过程 Congestion_remove(),该过程将在 3.2.3 节中进行分析。在拥塞删除的同时适量的信息素也被挥发了。之后,对蚁群路由表和蚂蚁内部状态进行更新。最后,通过对拥塞和非拥塞路径的比较,对该方案进行评价,过程结束,蚂蚁也消亡了。

从 sink 节点产生蚂蚁代理的过程就意义而言与普通节点稍有不同,即它是拥塞的通告人。正如前面所述,返回的蚂蚁代理要检查每个节点的内部拥塞标志位。识别拥塞后,子节点的拥塞状态将根据新的到达服务时间和父节点传给它的优先级进行更新。

3.2.3　拥塞消除阶段

```
Congestion_remove(T[i], t_s^pi, t_a^pi, SP(p_i), O(p_i), r_svc^i)
{
  d'(p_i) = 1, O'(p_i) = 0, r_svc^i = r_0;
  GetSvcRate(t_s^pi, t_a^pi, SP(p_i), O(p_i), r_svc^i, SP(i));
  d(p_i) = t_s^pi / t_a^pi;
  tatal_rate = 1/t_s^pi;
  if(O(p_i) = O'(p_i))
  {
  if(d(p_i) < d'(p_i))
       r_svc^i = r_svc^i / d(p_i);
    else
       r_svc^i = total_rate * SP(i)/SP(p_i);
  }
  else if(O(p_i) > O'(p_i))
       r_svc^i = total_rate * SP(i);
  else   r_svc^i = r_svc^i / d(p_i);
  d'(p_i) = d(p_i), O'(p_i) = O(p_i);
  r_svc^i = min(r_svc^i, 1/t_s^i);
  return(r_svc^i * h);
  GetSrcRate(r_svc^i);
  r_src^i = r_svc^i * SP(i)/SP(p_i);
  return r_src^i;
}
```

函数 Congestion_remove( )是参考文献[4,5]并根据具体的需要进行修改,采用六个参数传递。每个节点 i 仅有一个父节点 p_i。节点 i 通过节点 p_i 转发过来的数据包获取关于节点 p_i 的拥塞信息,包括 $t_s^{p_i}$,$t_a^{p_i}$,$SP(p_i)$,$O(p_i)$ 和 r_{svc}^i。节点 i 通过 GetSvcRate( )分别更新其本地调度率和源速率。初始调度速率设置为一个比较小的数值 r_0。$O(i)$ 表示节点 i 的后代节点数量。

GetSvcRate( )可以分为四种情况:

a)当子节点空闲时,节点 p_i 的间隔到达时间 $t_a^{p_i}$ 将增长,节点 p_i 的拥塞程度 $d(p_i)$ 将下降,节点 i 为了提高链路利用率,可以根据 $d(p_i)$ 提高它的数据发送率。

b)当新节点变得活跃了,数据包间隔到达时间 $t_a^{p_i}$ 将会下降,然后拥塞程度 $d(p_i)$ 将增长,这时节点 i 需要减少它的数据发送率。考虑到节点 i 的后代也都是节点 p_i 的后代,因为节点 i 是节点 p_i 的子节点,节点 i 为了保证公平性和高链路利用率,将它的数据发送率设置为允许的最大速率。新的数据发送率主要依赖于节点 i 和父节点 p_i 的优先级。

c)当后代节点数量 $O(p_i)$ 保持不变但某些节点没有足够的流量,拥塞程度 $d(p_i)$ 将变小。节点 p_i 为了提高链路利用率,根据 $d(p_i)$ 扩大数据发送率。

d)$O(p_i)$ 保持不变但是一些具有小流量的节点将产生更多的流量。这时,拥塞 $d(p_i)$ 将增加并可能大于 1,节点 i 将它的数据发送率设置为 b)中可允许的数据发送率,这是为了在维持高链路利用率的时候减轻拥塞。

h 是一个接近于 1 的比较小的参数,用来维持一个小的队列长度和高的吞吐量。

4 仿真实验及性能评价

4.1 仿真环境与参数

利用 NS2[9] 的数据包级的流量仿真对提出的方案的有效性进行了评价。NS2 是一个面向对象的网络仿真器,所有的仿真都由离散事件驱动,对于无线网络的模拟具有非常好的灵活性。仿真参数设置如下:网络区域范围为 1000 m ×600 m,200 个传感器节点均匀分布在该区域。Sink 节点位于坐标(1000,300)。每个传感器节点的传输和感知范围分别是 100 m 和 50 m。每个数据包的有效负载是 128 Byte。链路的数据速率是 1 Mbps。应用层类型采用 CBR,随着缓冲区的大小而发生变化。随机选择 10 个节点为数据源节点并执行仿真 360 s。源数据包的优先级设置为 1、2、3 三个级别。本仿真实验实现了 CCACO 算法,并与 PCCP 和 AntNet [6] 在吞吐量、丢包率、时延、能耗等性能指标方面进行了比较。

4.2 网络吞吐量

网络吞吐量是指单位时间内通过网络的数据量。在仿真实验中统计一段时间内 sink 节点上接收到的 CBR 数据包的个数除以统计时间,得到与网络吞吐量相当的一个性能指标。假定所有传感器节点产生数据的优先级都是相同的,每个节点的缓冲区大小都为 80 个数据包大小。在计算中,不考虑任何拥塞控制数据包,数值越高则性能越佳。如图 3 所示,用 Mbps 来表示网络吞吐量,CCACO 算法由于采用了蚁群优化的方法,故所获取的网络吞吐量明显优于 PCCP 和 AntNet,CCACO 和 PCCP 都采用了源数据包优先级的概念,吞吐量也优于 AntNet。

4.3 丢包率

丢包率指一般测试中丢失数据包数量占所发送数据包的比率。在仿真实验中,统计一段时间内各个传感器节点的应用层发出的 CBR 包个数和 sink 节点接收到 CBR 包的个数。用总的发送数量减去总的接收后除以总的发送数量就可以得到丢包率。假定所有 10 个节点具有不同的优先级,缓冲大小从 10 个数据包到 100 个数据包。图 4 给出了传感器节点在不同缓冲区大小的情况下的丢包率。随着缓冲区的增大,对 AntNet、PCCP 和 CCACO 的丢包率进行了比较。CCACO 由于采用了蚁群优化和源数据包优先级的技术,它的丢包率明显小于 AntNet 和 PCCP。

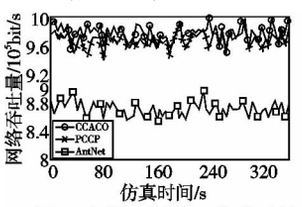

图3 网络吞吐量vs仿真时间

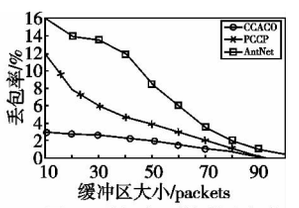

图4 丢包率vs缓冲区大小

4.4 平均端到端时延

平均端到端时延是指数据成功从源节点发到 sink 节点的时间。在仿真过程中选取一个离 sink 节点较远的传感器节点,并统计其产生数据包传输到 sink 节点的时延。如图 5 所示,在拥塞较轻的情况下,时延比较小,但是随着节点缓冲区增大时(大于 80 个数据包时),拥塞情况变得非常明显。这是由于缓冲区大时,缓存队列增加,从而产生了较大的排队时延。但 CCACO 算法的时延小于 PCCP 和 AntNet。

4.5 平均能耗

平均能耗是在网络生存周期内测量所有节点初始能量和最后能量差的平均值,设用 E_a 来表示[10,11],周期是从仿真实验开始到有节点消耗掉能量。E_i 表示节点的初始能量,E_f 表示节点的最终能量,N 为仿真中的节点个数,则 E_a 可表示为

$$E_a = \frac{\sum_{k=1}^{n} E_{ik} - E_{fk}}{N} \tag{4}$$

节点的数目从 20 一直增加到 200,每次增加 20 个。图 6 显示了三种方案随着节点的增多,能耗也随之增大。在 AntNet 执行时,能耗明显比另外两种方案要多出不少。CCACO 和 PCCP 的能耗非常接近,主要是因为都对源数据包采用优先级的标记,还因为 CCACO 采用了蚁群优化方法,所以进一步降低了能耗,并表现出了较好的能效。

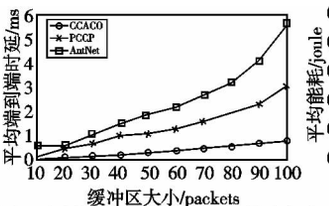

图5 平均端到端时延vs缓冲区大小

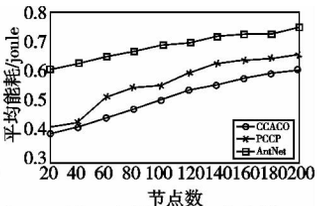

图6 平均能耗vs节点数

5 结束语

本文提出的基于蚁群优化的 WSN 拥塞控制算法通过结合蚁群优化和源数据包优先级加载等方法,在 PCCP 和 ACO 的基础上进行了结合、优化,能够明显地提高网络吞吐量、减少丢包率以及时延和能耗。接下来将从理论上展开对工作的研究,分析在无线传感器网络中的准确性和拥塞算法的公平性。

参考文献:

[1] AKYILDIZ I F, SU W, SANKARASUBRAMANIAM Y, *et al*. A survey on sensor networks [J]. IEEE Communication Magazine, 2002, 40(8): 102-114.

[2] 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.

[3] 刘拥民,蒋新华,年晓红. 无线传感器网络拥塞控制研究[J]. 计算机应用研究,2008,25(2):565-568,571.

[4] WANG C, LI B, SOHRABY K, *et al*, Upstream congestion control in wireless sensor networks through cross-layer optimization [J]. IEEE Journal on Selected Areas in Communications, 2007, 25(4): 786-795.

[5] DHURANDHER S K, MISRA S, MITTAL H, *et al*. Using ant-based agents for congestion control in Ad hoc wireless sensor networks [J]. Cluster Computing, 2011, 14(1): 41-53.

[6] DORIGO M, CARO G D. AntNet: distributed stigmergetic control for communications networks [J]. Journal of Artificial Intelligence Research, 1998, 9(1): 317-365.

[7] LIU Yuan, MA Zheng-xin, CAO Zhi-gang. A mitigating stagnation-based ant colony optimization routing algorithm [C]//Proc of IEEE International Symposium on Communications and Information Technology. 2005: 36-39.

[8] DORIGO M, COLORNI A, MANIEZZO V, The ant system: optimization by a colony of cooperating agents [J]. IEEE Trans on System, Man, and Cybernetics-Part B, 1996, 26(1): 29-41.

[9] The network simulator NS-2.34 [EB/OL]. http://www.isi.edu.

[10] XIE Hui, ZHANG Zhi-gang, NIE Feng. A novel routing protocol in wireless sensor networks based on ant colony optimization [J]. International Journal of Intelligent Information Technology Application, 2010, 3(1): 1-5.

[11] CORREIA S L O B, CELESTINO J, CHERKAOUI O. Mobility-aware ant colony optimization routing for vehicular Ad hoc networks [C]// Proc of IEEE Wireless Communications and Networking Conference. 2011: 1125-1130.