

# 一种基于权重的动态分簇算法

张卫刚<sup>1</sup>, 何庆<sup>1</sup>, 陈浩亮<sup>2</sup>, 许骏<sup>1</sup>

(1. 华南师范大学计算机学院, 广东 广州 510631; 2. 广州无线电研究所, 广东 广州 510500)

[摘要] 基于分簇算法,提出了一种基于权重的动态分簇算法(WDCA)。该算法综合考虑了节点与其邻居节点的相对速度,节点到其邻居节点的平均距离,节点的能量以及节点的邻居数等因素来选择簇头;同时取消了一般加权分簇算法中簇成员到簇头只有一跳的限制,而是根据簇内成员数动态调整。模拟结果表明,与经典的加权分簇算法(WCA)相比,该算法的簇头稳定性、网络的负载均衡都有很大提高。

[关键词] Ad Hoc 网络; 分簇算法; 动态; 权重

[中图分类号] TP393 [文献标识码] A [文章编号] 1000-9965(2009)01-0035-05

## A weighted dynamic clustering algorithm

ZHANG Wei-gang<sup>1</sup>, HE Qing<sup>1</sup>, CHEN Hao-liang<sup>2</sup>, XU Jun<sup>1</sup>

(1. College of Computer, South China Normal University, Guangzhou 510631, China;

2. Guangzhou Radio Research Institute, Guangzhou 510500, China)

[Abstract] A weighted dynamic clustering algorithm(WDCA) is proposed which takes into consideration the relative velocity, average distance, battery power and neighbors of mobile nodes to selecting cluster head. The restriction in many weighted clustering algorithms, that the cluster members to their cluster head not exceed one hop is dropped, but dynamic adjustment it depends on the number of cluster member. The simulation results show that the stability of cluster heads, network load balancing have greatly improved compared to the typical algorithm weighted clustering algorithm (WCA).

[Key words] ad hoc networks; clustering algorithm; dynamic; weight

在一般的Ad Hoc网络中,所有节点的地位是平等的,原则上不存在网络瓶颈节点,网络比较健壮。但随着网络规模的增大,维护网络的控制开销迅速增加,并且随着共享相同网络资源的用户数量的不断增加,每个用户获得的吞吐量将急剧下降。因此,它主要适应于中小型Ad Hoc网络。

分簇是解决Ad Hoc网络可扩展性的有效方法。

分簇Ad Hoc网络中不同层次的节点分配不同功能,其中簇头节点负责簇的形成与维护,簇头的集合被称为支配集(dominate set)。如果网络中的簇成员节点到其簇头的跳数不超过 $d$ 跳,则这些簇头组成的集合称为 $d$ -跳支配集。由<sup>[1]</sup>可知,构建一个最小 $d$ -跳支配集是NP-完全问题,因此该问题基于启发式的解存在。

[收稿日期] 2008-09-14

[基金项目] 广东省科技攻关计划项目(2006B15001004);广州市科研条件建设项目(穗科条2006-21)

[作者简介] 张卫刚(1983-),男,硕士研究生,研究方向:无线自组路由协议与算法

通讯作者:许骏(1959-),男,教授,博士生导师

Ad Hoc 网络中节点是任意移动的,网络的拓扑结构随时可能发生改变,簇的重新选择是不可避免的,因此如何选择适合的簇头,高效的进行簇的维护便成为分簇算法研究的重点. 到目前为止已经提出了许多分簇算法,其中一些算法规定簇内节点到簇头不能超过一跳,我们称为单跳的分簇算法(参见文献[2-6]等);而另一些算法是簇内节点到簇头不能超过限定的跳数,我们称这类算法为多跳的分簇算法(参见文献[1,7]等). 本文在综合分析这些算法的基础上,提出了一种新型的基于权重的动态分簇算法.

## 1 相关工作

近年来,国内外研究人员对 Ad Hoc 网络分簇算法进行了大量的研究,提出了许多分簇算法. 我们可以把这些算法分为两类:单跳的分簇算法和多跳的分簇算.

### 1.1 单跳的分簇算法

单跳的分簇算法,即簇内节点到簇头最多一跳. 最小 ID 分簇算法<sup>[3,5]</sup>是一种简单的分簇算法. 该算法为每个移动节点分配唯一的 ID,在相邻节点中,具有最小 ID 的节点作为簇头,簇头的一跳邻居节点成为该簇头所在簇的成员节点. 该算法实现简单、收敛较快,维护簇头花费的开销较小;但它倾向于选择 ID 较小的簇头,这将导致这些节点因为担任簇头职责的频率较高而消耗更多的电池能量,从而缩短了整个网络出现分割的时间.

在最高节点度算法<sup>[4]</sup>中,每个节点通过交互控制消息来获得邻居节点数以及邻居节点的邻居节点数,该节点和其一跳邻居节点中具有最大节点度的节点被选为簇头. 该算法的优点在于网络中簇的数目较少,从而减少了分组投递时延. 其缺点是当节点的移动性较强时,簇会变得不稳定,簇头更新频率很高.

最低移动性分簇算法<sup>[6]</sup>则是根据节点的移动性来选举簇头. 节点的移动性越高,其分配的权重越低. 该算法选择邻居节点中具有最高权重的节点作为簇头,当权重相同时,则选举 ID 最小的作为簇头. 当移动性较强时,该算法可以明显地减少簇头更新的频率. 它的缺点在于当节点权重相近时簇头更换频繁,并且没有考虑系统的负载均衡和节点的

能量损耗.

加权的分簇算法(WCA, weighted clustering algorithm)<sup>[2]</sup>是由 Chatterjee M 等提出的. WCA 算法综合考虑了节点的移动性、实际处理能力、传输功率以及电池能量耗费等因素对网络簇结构稳定性的影响,并对各种决定性参数变量赋以不同的对应权重,从而根据不同的环境适应需求,实现一种优化的选举簇头、划分网络节点形成簇的策略.

WCA 算法有很强的适应性,可以分配不同的权重来适应不同的应用环境. 但该算法在选择簇头时,并未考虑负载均衡,并且当节点权重相近时,簇头更换频繁.

总的来说,这类算法节点到簇头最多一跳,如果网络规模较大将产生很多的簇,仍然存在和平面路由协议类似的问题;而且当网络节点分布不均匀时,会造成有些簇成员过多,而另一些簇成员很少的负载均衡问题.

### 1.2 多跳的分簇算法

由于单跳分簇算法存在的问题,对于较大的网络,我们就应该允许在簇内存在多跳. 一般做法是预先设定一个阈值  $d$ ,规定簇内节点到簇头不超过  $d$  跳.

Ghosh R K 等人提出了一种密集簇网(DCG)协议<sup>[7]</sup>,该协议以 K-树核方式构建簇,簇与簇的边缘节点为网关节点(通常有多个网关节点). 一个簇的规模如果超出某个限定的阈值,就产生一个子簇头,它负责簇内的路由来减轻簇头的负担.

文献[1]提出了一种把网络分成多个  $d$  跳域集(即簇)的方法. 首先,算法为每个节点设置一个 WINNER 值,每个节点的 WINNER 值为节点的 ID;然后进行两轮泛洪(flooding)操作,再根据节点所获得的 WINNER 值来选择簇头. 在该方法中,网络节点倾向于选择做过簇头的节点为新的簇头,因而很容易产生簇头节点瓶颈问题;该算法的簇维护开销也很大.

以上多跳分簇算法虽然采用了多跳思想,但都没有考虑节点的移动性、实际处理能力、传输功率、剩余能量对分簇的影响,并且簇的维护开销很大.

## 2 WDCA 算法

### 2.1 预备知识

(1)将 Ad Hoc 网络抽象为一个连通的无向图

$G=(V,E)$ ,其中 $V$ 表示无线网络的节点集, $E$ 表示节点之间的双向链路集合;假设节点数为 $n$ .

(2)下面给出3个假设:

假设1:同文献[8],假设每个节点都能够在限定时间内收到其一跳邻居发送的信息.

假设2:假设每个节点都装有GPS系统,节点的位置信息可以通过GPS系统获得.

假设3:假设簇头消耗的能量远远大于普通节点.

## 2.2 簇头选择与簇的形成

算法初始化与簇的形成包含以下步骤:

第1步:算法初始化时,每个节点都处于未决定状态,通过周期性的交互Hello消息(初始化时权值为未分配标志),获得邻居节点的信息,使每个节点 $i$ 都可以确定各自的邻居节点数以及其邻居节点的速度.节点 $i$ 的邻居节点数

$d_i = \sum_{i' \in V, i' \neq i} \{dist(i, i') < tx_{range}\}$ ,其中 $tx_{range}$ 是节点 $i$ 的传输范围.

第2步:计算节点度数与簇的最佳度数 $\delta$ 之差: $\Delta_i = |d_i - \delta|$ ,其中 $\delta$ 的值预先设定.

第3步:对每个节点 $i$ ,计算其到邻居节点的平均距离: $\bar{D}_i = \sum_{i' \in N(i)} \frac{1}{d_i} \sqrt{(x_i - x_{i'})^2 + (y_i - y_{i'})^2}$ .

第4步:计算每个节点 $i$ 与所有邻居节点的平均相对速度大小: $\bar{V}_i = \sum_{i' \in N(i)} \frac{1}{d_i} \sqrt{(v_{xi} - v_{xi'})^2 + (v_{yi} - v_{yi'})^2}$ .其中 $(v_{xi}, v_{yi})$ 是节点 $i$ 的当前速度, $(v_{xi'}, v_{yi'})$ 是节点 $i$ 的邻居 $i'$ 当前的速度.

第5步:计算节点 $i$ 做簇头的累积时间 $P_i$ ,由于我们假设簇头消耗的能量远远大于普通节点,因此 $P_i$ 可以代表已经消耗的节点的能量.如果 $P_i > P_{max}$ ,则 $P_i = 2 \times P_{max}$ .其中 $P_{max}$ 是预先设定的值,表明该节点所剩的能量已经非常有限.

第6步:计算节点 $i$ 的综合权重 $W_i = w_1 \Delta_i + w_2 \bar{D}_i + w_3 \bar{V}_i + w_4 P_i$ ,其中: $w_1 + w_2 + w_3 + w_4 = 1$ .每个节点都将权值加入Hello消息中,广播给各自的邻居节点.

第7步:如果节点 $i$ 的权值 $W_i$ 比所有的邻居节点大,则该节点成为簇头,该节点所有的未选择簇头的一跳邻居节点加入该簇.其所有两跳的未选择簇头的节点检测到该簇的簇成员数小于 $\delta$ ,则向该簇

头节点发送加入簇请求消息,簇头接收到该消息后,如果是下面两种情况的一种,则发送确认消息给请求节点,使其加入该簇,否则不发送确认消息,直到簇成员数达到 $\delta$ 或者跳数达到 $d$ .

① 请求节点到簇头的跳数不大于该簇当前的最大跳数且该簇的簇成员数小于 $\delta$ .

② 请求节点到簇头的跳数为 $h+1$ ,且 $h+1 \leq d$ ,该簇的簇成员数小于 $\delta$ .其中 $h$ 是簇当前的最大跳数.

如果某节点的权值虽然不是最大的,但自己的权值比所有的未加入簇的一跳邻居节点的权值大,并且自己未能加入已加入簇的邻居所属的簇,则该节点成为簇头,该节点所有的未选择簇头的一跳邻居节点加入该簇.

第8步:重复第2到第7步,直到所有节点或者成为簇头或者成为簇成员.

## 2.3 簇的维护

在Ad Hoc网络中,由于网络节点的频繁移动,网络的拓扑结构在不断地变化.随之而来的是簇结构的改变,每次簇结构的改变都会增加系统的开销,降低系统的性能.该算法的簇维护策略,就是为了尽量保持簇结构的稳定性而制定的.主要分为以下几种情况来讨论:

(1)簇头的改变

① 当两个簇头节点A、B(权值分别为 $W_A, W_B$ )成为一跳邻居节点时,如果 $W_A > W_B$ ,则簇头B状态不改变,节点A放弃自己的簇头地位进入未分配状态,其簇成员也进入未分配状态,重新选簇.

② 当簇头的电池能量耗尽或发生故障时,其簇内成员进入未分配状态重新选簇.

③ 当一个簇的簇头离开本簇后,该簇头及其簇成员进入未分配状态,重新选簇.

④ 当自己邻居节点所在的簇或者簇成员数达到 $\delta$ 或者自己到该簇簇头的跳数大于 $d$ 时(即该节点不能加入任一邻居所在的簇),自己成为簇头.

⑤ 当一个簇内簇成员A的权值大于某一个预设的值 $x$ ,小于该簇的簇头B的权值的50%,且二者为邻居节点时,A成为该簇新的簇头节点,B成为该簇的簇成员.如果该簇的某个簇成员发现自己到簇头的跳数大于 $d$ ,则进入未分配状态,重新选择簇加入.

## (2) 加入一个簇

① 当一个节点 A(不论其处于何种状态)进入某个簇内时,先检查该簇的簇成员数是否达到  $\delta$ ,如果没有达到  $\delta$ ,则发送请求加入该簇,如果达到  $\delta$  值,则按(1)中第四进行操作。

② 当处于未分配状态的节点到达某簇的边缘(即该节点的邻居节点中有该簇的簇成员)时,节点检查该簇的信息,如果符合加入该簇的条件,则发送请求加入该簇。

③ 若自己是一个簇的  $d$  跳成员或自己所在的簇已经达到饱和且自己不是该簇簇头的邻居节点,而自己的邻居成为一个新簇的簇头,则发送请求加入该簇。

## (3) 离开一个簇

① 如果一个簇的簇成员离开该簇或者其到达其簇头的跳数超过了  $d$  跳,则该节点进入未分配状态,重新选簇。

② 如果某簇的簇头节点离开该簇,则按(1)中③进行操作。

# 3 算法仿真与分析

## 3.1 仿真环境

WCA 是一种非常典型的 Ad Hoc 网络分簇算法,这里就利用 NS2 来比较 WCA 与 WDCA 的性能。

仿真是在  $200 * 200$  的系统里进行的,节点运动速度是在  $[0,5]$  之间随机选取,模拟时间  $t = 120\text{ s}$ ,当节点传输范围在  $[10,80]$  变化时,节点数  $n = 60$ ,当节点在  $[30,90]$  变化时,节点传输范围为 40,特别地在仿真求  $LBF$  时, $n = 60$ ,传输范围为 40。

WCA 与 WDCA 共有的参数:最佳成员个数  $\delta = 10$ ;  $w_1 = 0.5, w_2 = 0.2, w_3 = 0.2, w_4 = 0.1$ 。

WDCA 算法中其他参数的:最大跳数  $d = 2$ ;权值阈值  $\alpha = 0.3$ ;能耗阈值  $P_{\max} = 80$ 。

## 3.2 仿真结果及分析

(1) 比较二者在节点传输范围变化情况下,簇头数目的变化情况,如图 1 所示。由图可知,簇头数目随着节点传输范围的增大而减少,且 WCA 的簇头变化幅度比 WDCA 大,这与本文所提出的算法规则较吻合,因为我们允许有到簇头两跳的簇成员节点。

(2) 比较二者在节点传输范围变化情况下,平

均簇保持时间的变化情况,如图 2 所示。由图可知,二者的簇保持时间都随节点的传输范围变化先增大后减小,这是因为当节点传输范围增大到一定程度时,网络中节点的邻居数相近,导致节点权值相近,簇的保持时间有所下降。

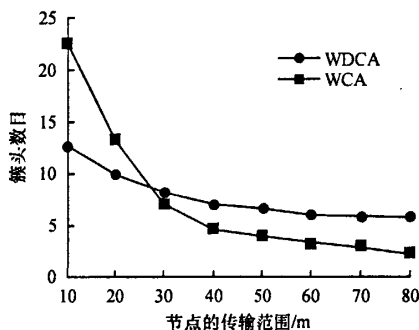


图 1 簇头数目随节点传输范围的变化

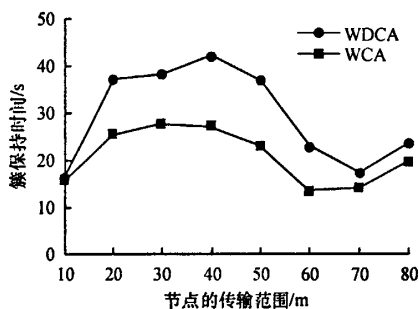


图 2 簇保持时间随节点传输范围的变化

(3) 比较二者在节点数目变化情况下,簇的形成与维护所用控制包的变化情况,如图 3 所示。由图可知,WDCA 的控制包数目要比 WCA 高一些,这是因为我们要维护的簇的规模要大于 WCA。

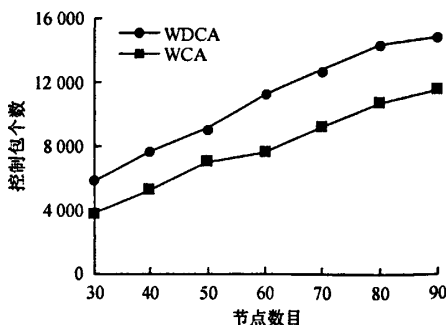


图 3 簇的形成与维护所用控制包随节点数目的变化

(4) 比较二者的负载均衡因子 ( $LBF$ , Load Balancing Factor) 随时间的变化情况,如图 4 所示 ( $LBF$

$$= \frac{n_c}{\sum_i (x_i - p)^2}, \text{其中 } p = \frac{N - n_c}{n_c}, x_i \text{ 是簇 } i \text{ 的成员节点数, } n_c \text{ 是簇头的数目, } N \text{ 为网络节点数}.$$

$LBF$  越大,算法的负载均衡度就越好. 由于设定的初始化为 10 s,这段时间内我们没有进行选簇,而主要是让节点了解各自邻居的信息. 由图可知,WDCA 的负载均衡度相对 WCA 有了很大的提高.

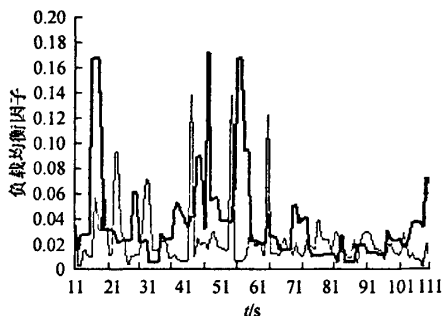


图4 簇的负载均衡因子(LBF)随时间的变化

## 4 结论

基于分簇的路由协议对提高较大型 Ad Hoc 网络的性能有很重要的意义. 我们在 WCA 算法的基础上,结合多跳分簇的思想,提出了一个新型的多跳分簇算法. 该算法产生的簇中,簇成员到簇头的跳数是可以变化的;当网络稀疏时,形成簇的最大跳数较大,当网络稠密时,形成簇的最大跳数较小. 该算法也继承并优化了 WCA 的选簇方案,综合考虑了节点与邻居节点的相对速度,到邻居节点的平均距离,能量以及邻居数等因素来选择簇头,簇比较稳定,算法适应性也很好.

## [参考文献]

- [1] AMIS A D, PRAKASH R, VUONG T, et al. Max-Min D-Cluster formation in wireless Ad Hoc networks [C] // Proceeding of IEEE Conference on Computer Communications (INDOCOM), 2000:32-41.
- [2] CHATTERJEE M, DAS S, TURGUT D. WCA: A weighted clustering algorithm for mobile ad hoc networks [J]. Cluster Computing, 2002:193-204.
- [3] EPHREMIDES A, WIESELTHIER J E, BAKE D J. A design concept for reliable mobile radio networks with frequency hopping signaling [C] // Proceedings of IEEE, 1987:56-73.
- [4] PAREKH A K. Selecting routers in ad-hoc wireless networks [C] // Proceedings of the SBT/IEEE International Telecommunications Symposium, Aug, 1994.
- [5] GERLA M, TSAI J T C. Multiclustet, mobile, multimedia radio network [J]. Wireless Networks, 1995:255-265.
- [6] BASAGNI S. Distributed clustering for ad hoc networks [C] // Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks. Jun, 1999: 310-315.
- [7] GHOSH R K, GARY V, MERTEI M S, et al. Dense cluster gateway based routing protocol for multi-hop mobile ad hoc networks [J]. Ad Hoc Networks, 2006: 168-185.
- [8] LIN C R, GERLA M. Adaptive clustering for mobile wireless networks [J]. IEEE Journal on Selected Areas in Communications, 1997, 15(7): 1265-1275.

[责任编辑:王景周]