

Ad Hoc 网络中一种基于相关度的分布式分簇算法^{*}

孟 斌 张尧学

(清华大学计算机系普适计算教育部重点实验室 北京 100084)

摘 要 无线自组网 ad hoc 是一种不依赖于基础设备的无线移动网络,分簇是管理 ad hoc 的一种较为有效的方式。本文提出 ad hoc 网络中一种新的分簇算法,此算法以节点间的相关度以及节点密度作为选择簇头和分簇的标准,算法在各节点分布执行,并且通过节点间消息的传递来得到最终的簇结构。实验结果表明,该算法产生的簇结构比节点度算法更均衡,也因此在此应用上具有更好的表现。

关键词 ad hoc 网络,分簇,分布式算法

A Relationship-based Distributed Clustering Algorithm in MANET

MENG Bin ZHANG Yao-Xue

(Key Laboratory of Pervasive Computing, Tsinghua University, Beijing 100084)

Abstract Clustering is an effective management method in ad hoc networking(MANET). This paper proposes a new distributed clustering algorithm for MANET. The algorithm is based on relationship between nodes and density of nodes in selecting clusterheads, then exchange information between nodes to get the final clustering structure, the experiment results show the structure it produces has bigger balance factor than the degree algorithm and better performance in application.

Keywords MANET, Clustering, Distributed algorithm

在 ad hoc 网络(MANET)中,对节点进行分簇管理是比较有效的方式,其优点是网络可扩展性好,路由和控制开销较小,并且容易实现移动性管理和网络的局部同步^[1]。由于节点的移动性和难以统一管理,目前主要采用分布式的分簇算法。分布式的分簇算法一般可分为两个步骤:确立簇头节点以及形成簇结构,即确定其它节点的归属。目前主要有以下几种分簇算法:对节点 ID 进行排序的最小 ID 启发式算法;基于节点度数的最高节点度算法^[3];基于移动性、能量耗费、负载均衡以及簇稳定性的算法;还有就是折衷考虑各个方面的加权组合分簇算法^[4],权重的分配与动态调整是该算法实际运用中的难点。

目前,最高节点度算法由于简单高效,在实际中应用最广泛。然而,该算法在选择簇头时只考虑了节点的度数,没有考虑邻居节点的相对位置以及邻居节点周围节点的分布情况,对节点周围拓扑结构考虑得不完全,使得选择出来的簇头并不一定具有最大连通性。同时,由于非簇头节点与簇头节点的关系没有一个度量尺度,造成非簇头节点加入簇时具有很大的盲目性,因而容易造成分簇的不平衡性。本文改进了这一算法,首先定义了节点间的相关度和节点的相关密度,并在相关度和相关密度基础上提出了一种分布式的分簇算法,最后对算法进行了分析和模拟测试。

1 网络模型

我们可以把无线自组网简化为一个无向图 $G=(V, E)$, $|V|=n$, 表示的是移动节点的集合。当且仅当 u 和 v 相互在对方传输范围内的时候,我们把 u, v 称为邻居节点,边 $(u, v) \in E$ 。我们用 $\Gamma(v)$ 来表示 v 的邻居节点集($v \in V$)。 G 的拓扑结构随着节点的移动而改变。节点通过 GPS 等设备能获知自己的位置,通常是一个坐标三元组 (x, y, z) 。对 ad hoc 网络进行分簇意味着把网络划分为多个子节点集——簇(cluster),

每个簇有一个簇头节点和多个簇内节点。簇有可能相交,相交的节点有多个簇头,称为网关节点。

2 算法相关定义

定义 1 相邻节点的相关度。设 R_u 为节点 u 的无线传播范围, $\text{dist}(u, v) = |uv|$ 为邻居节点 u, v 的距离,距离可通过邻居节点定期交换位置信息得到。

定义

$$\text{Rel}(v \rightarrow u) = e^{-\frac{\text{dist}(u, v)}{R_u}} \quad (1)$$

(e 为自然对数底)为节点 v 对 u 的相关度。 R_u 在这里起到对不同节点的传播距离进行归一化的作用。

定义 2 单条 m -hop 路径上节点的相关度。设节点 u_0 经过 m 跳可到达节点 u_m , 中间节点为 $u_1, u_2, \dots, u_{m-1}$, 即 $l: u_0 \rightarrow u_1 \rightarrow \dots \rightarrow u_m$, 则节点 u_0 在该路径对节点 u_m 的相关度定义为

$$\begin{aligned} \text{Rel}(u_0 \xrightarrow{l} u_m) &= \text{Rel}(u_0 \rightarrow u_1) \cdots \text{Rel}(u_{m-1} \rightarrow u_m) = \\ &= e^{-\left[\frac{\text{dist}(u_0, u_1)}{R_{u_1}} + \frac{\text{dist}(u_1, u_2)}{R_{u_2}} + \dots + \frac{\text{dist}(u_{m-1}, u_m)}{R_{u_m}}\right]} \end{aligned} \quad (2)$$

可以看出,相关度用节点间的路径长度与节点传播范围的对比来反映节点间的紧密程度。由相关度的定义,可得到相关度的几个性质:

性质 1 对相邻节点而言, $e^{-1} \leq \text{Rel}(v \rightarrow u) < 1$, 且当 v 在 u 的传播范围边缘时相关度最小,为 e^{-1} , v 离 u 越近,相关度越大,当 v 与 u 趋于重合时,相关度趋于 1。

性质 2 相关度对路径上的节点距离具有相加性,这个性质由公式(2)可看出。

性质 3 相关度随着路径的增长而递减,即若路径 $l: w \rightarrow \dots \rightarrow v \rightarrow \dots \rightarrow u$, 则有 $\text{Rel}(w \rightarrow u) < \text{Rel}(v \rightarrow u)$ 。由式(2)以及 $\text{Rel}(w \rightarrow v) < 1$ 可以得出结论。

定义 3 节点 1-hop 内节点相关密度的度量。设节点集

^{*} 基金项目:课题得到国家自然科学基金(69873024)资助。孟 斌 硕士生,主要研究方向为 ad hoc 网络;张尧学 教授,博士生导师。

$S=\{v_1, v_2, \dots, v_n\}$ 为 u 的邻居节点集, 则节点 u 一跳内相关密度由下面公式定义:

$$\text{Dens1}(u) = \sum_{v_i \in S} \text{Rel}(v_i \rightarrow u) \quad (3)$$

定义 4 节点 2-hop 内节点相关密度的度量。设节点集 $S=\{v_1, v_2, \dots, v_n\}$ 为 u 的邻居节点集, 则节点 u 两跳内相关密度由下面公式定义:

$$\text{Dens2}(u) = \sum_{v_i \in S} [\text{Dens1}(v_i) \text{Rel}(v_i \rightarrow u)] \quad (4)$$

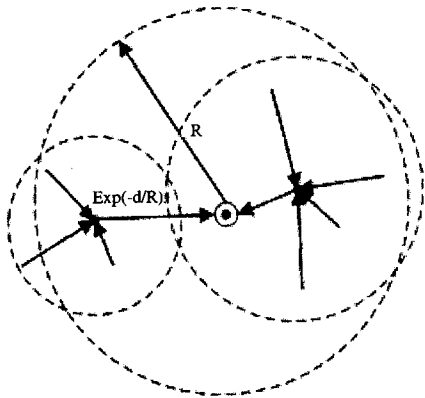


图 1 2-hop 相关密度的计算

说明: 由于不同节点的邻居节点可能重合, 重合节点对 u 的相关度就可能沿不同路径计算了两次以上, 但由于要把 1-hop 节点和 2-hop 节点分离开来需要比较复杂的计算, 同时相加也反映了节点沿不同路径进行通讯的能力, 所以这里只用简单相加进行度量。

3 分簇算法

我们将用到以下的术语来描述该算法:

ClusterHead/CH——簇头节点。每个簇都存在唯一一个簇头节点。

Normal/NM——普通节点。簇内除了簇头外的其他节点。

GateWay/GW——网关节点。连接两个或两个以上簇头的节点。网关节点同时也是普通节点。

UnCovered/UC——节点的原始角色。还没加入到任何簇中。

Cluster(v)——以 v 为簇头节点的簇内节点集。

Master(v)—— v 所属簇的簇头节点的集合。只有 v 是网关节点时, Master(v) 内元素的个数大于 1。

Ch(-) 和 Join(-, -)——布尔变量, 用来记录节点接收到的 Ch 消息和 Join 消息。

我们的分簇算法在每个节点分布执行, 每隔一定的时间周期 T 执行一次。下面我们节点 v 作为对象来描述基于相关度的分布式分簇算法。在算法描述中, 我们用 $\wedge$ 和 $\vee$ 来表示布尔操作符“and”和“or”。算法步骤如下:

Step1 初始化。每个节点分别设自己的 1-hop 相关密度和 2-hop 相关密度的初始值为 0, 簇内节点集和簇头节点集设为空。

Step2 第一轮广播。向邻居节点广播 HELLO 消息, 在消息中包含自己的位置信息。

Step3 第一轮计算。节点接收到邻居节点的第一轮广播消息后, 计算邻居节点对自己的相关度(公式(1)), 并且更新邻居节点表。当接收完所有邻居节点的信息后(可通过设

定时限来实现), 计算自己的 1-hop 相关密度(公式(3))。

Step4 第二轮广播。向邻居节点广播自己的 1-hop 相关密度。

Step5 第二轮计算。接收到所有邻居节点的 1-hop 相关密度后, 计算自己的 2-hop 相关密度(公式(4))。

Step6 第三轮广播。向邻居节点广播自己的 2-hop 相关密度。

Step7 簇头的确立。节点接收到邻居节点的 2-hop 相关密度后, 比较自己的 2-hop 相关密度和邻居节点的 2-hop 相关密度。如果自己的 2-hop 相关密度值为最大, 则把自己设为簇头节点, 并且广播 CH 信息。

Step8 收到邻居节点 u 的 Ch 消息后, 将 u 加入自己的簇头节点集中。节点 v 检查自己是否收到在所有邻居节点中 $\text{Rel}(v \rightarrow z) > \text{Rel}(v \rightarrow u)$ 的 Join(z, x) ($x \in V$) 消息。如果是的话, 表示节点 v 将不会从这些节点接收 Ch 消息, 且 u 是 v 的发送 Ch 消息的邻居节点中与 v 具有最大相关度的节点, 因此 v 加入以 u 为簇头节点的簇, 并且广播 Join(v, u) 消息, 结束运算。否则, 记录下 u 已发送过 Ch 消息, 继续等待其他邻居节点发送消息。此时, 如果接收超过 1 个节点的 Ch 消息, 则 v 的角色变为网关节点, 但只加入具有最高相关度簇头的簇中, 即只发送一个 Join 消息。用伪代码描述如下:

```
On Receiving Ch( $u$ );
Begin
    Ch( $u$ ) := true;
    Master( $v$ ) := Master( $v$ )  $\cup$  { $u$ };
    If  $\wedge_{z \in \text{Join}(v)} \text{Rel}(v \rightarrow z) > \text{Rel}(v \rightarrow u) (\vee x \in V \text{Join}(z, x))$ 
    Then begin
        Send Join( $v, u$ );
        If |Master( $v$ )| > 1
        Then begin
            Role( $v$ ) := GW;
        End
        Else begin
            Role( $v$ ) := NM;
        End
    End
End;
```

Step9 接收到邻居节点的 Join(u, t) 消息后, 节点 v 检查自身是否簇头节点, 如果是, 节点 v 检查是否 $v=t$, 如果是, 则把节点 u 加入以 v 为簇头的簇。如果 v 所有的邻居节点 z 中 $\text{Dens2}(z) < \text{Dens2}(v)$ 的都已发送过 Join 消息, 运算结束。 v 并不需要关注邻居节点中 $\text{Dens2}(y) > \text{Dens2}(v)$ 的节点, 因为这些节点肯定已经加入以 x 为簇头 ($\text{Dens2}(x) > \text{Dens2}(v)$) 的簇(只有这样 v 才有可能成为簇头节点)。

如果 v 为普通节点, 运算结束。否则 v 还没做出决定, 角色为 UC, 进行下面的操作。首先 v 检查邻居节点中所有符合 $\text{Dens2}(z) > \text{Dens2}(v)$ 条件的节点的角色选择。如果 v 接收到所有的符合该条件节点的信息, 那么 v 就通过这些消息的类型及参数来决定自己的角色。如果 v 接收过来自邻居节点的 Ch 消息, 则按照 8 中的判断依据来加入簇。否则, 那些节点发送的都是 Join 消息, 这就意味着那些节点都以非簇头节点的身份加入簇, 因此, v 在所有还没有做出决定的节点中具有最大的 2-hop 相关密度, v 成为簇头节点。同时, 检查所有满足 $\text{Dens2}(y) < \text{Dens2}(v)$ 的邻居节点, 看它们是否已经发送过 Join 消息。如果都已发送过的话, v 结束运算, 自身独立成为一个簇。否则, v 广播 Ch 消息。限于篇幅, 伪代码略去。

4 算法分析

• 没有两个簇头节点是相邻的。从算法中可以看出, 簇

头的产生有两种方式:一是 2-hop 相关密度为邻居节点中最大,这样它的邻居节点都不可能成为簇头;二是所有邻居节点都以普通节点加入其它的簇,这样邻居节点中也没有簇头节点。所以,没有两个簇头节点是相邻的。

• 节点以发送 Join 消息或者接收 Join 消息为结束。对普通节点而言,发送 Join 消息后算法结束;对簇头节点而言,接收到所有邻居节点的 Join 消息后算法结束。

• 最坏情况下,消息传递次数为 $O(d)$, d 为网络直径(任意两个节点的最小跳数的最大值)。最坏情况在节点沿线性由疏到密排列时出现。

5 实验结果和讨论

我们在模拟器 GloMoSim 上实现了我们的分簇算法,

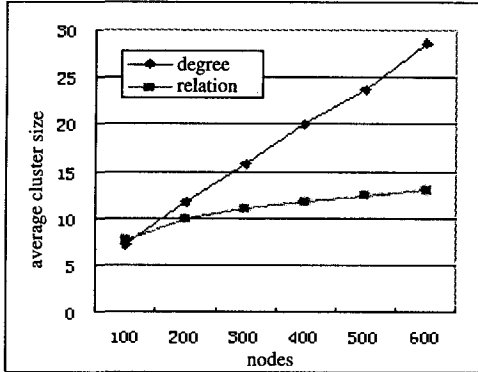


图 2(a) 平均簇内节点数 vs 节点数

从图 2(a)中我们可以看出,随着节点数的增加,节点度算法的平均簇内节点数显现出几乎是线性增长的现象,而相关度算法的平均簇内节点数的增长则平缓得多。相关度算法使每个簇的簇内节点数保持一个较为恒定的数值。

分簇比较理想的结果是各个簇有比较平均的节点个数。因为某些簇头节点负担重而某些簇头负担轻的话,会导致网络流量的不平衡性,从而形成网络的瓶颈。为了度量分簇算法的不平衡性,在文[4]中,定义了分簇平衡指数 LBF:

$$LBF = \frac{n_c}{\sum_i (x_i - \mu)^2} \quad (5)$$

n_c 代表的是簇头的个数, x_i 是第 i 簇的节点个数, $\mu = N/n_c$ 代表的是平均每簇的节点个数。可以看出, LBF 值越大,分簇结果越均衡。

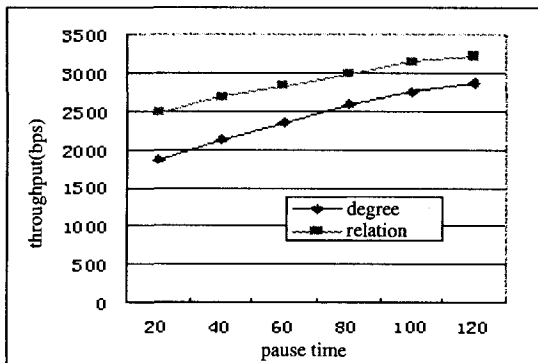


图 3(a) 单位节点吞吐量

吞吐率和端到端延时是反映路由算法应用性能的两个比较直观的指标,由于我们是把两种分簇算法分别应用于同一

GloMoSim 模拟器是 UCLA 提供的专门为无线网络系统设计的一种可扩展的仿真环境。实验中,节点分布为均匀分布,移动模型采用 Random Way-point 模型, MAC 层采用 802.11 协议。由于我们的算法是基于网络拓扑结构的算法,没有考虑能量、移动性等因素,所以只与最大节点度启发式算法进行比较。

5.1 分簇算法在簇结构方面的表现

我们把移动区域固定为 $200m \times 200m$, 每个节点的传输半径都固定为 15m, pause time 固定为 60s,通过改变节点数目来观察算法在不同节点密度下的性能表现。从 a)簇头的数目;b)分簇平衡指数^[4]两个方面进行了测试,并与最大节点度算法做比较。

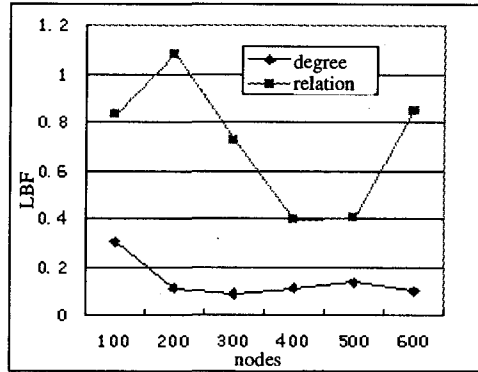


图 2(b) 分簇平衡指数 vs 节点数

从图 2(b)中可以看出,相关度算法由于综合考虑了节点 2-hop 内的节点分布情况,所以簇内节点数分布比较均匀,得到了比最高节点度算法高得多的分簇平衡指数。这表明,相关度算法具有较好的负载平衡能力,从而可以减少网络拥塞和瓶颈节点出现的概率。

5.2 分簇算法在应用中的性能表现

我们在路由算法 CBRP 中分别应用节点度算法和相关度算法, CBRP 是一种基于分簇的层次路由协议。把移动区域固定为 $200m \times 200m$, 每个节点的传输半径都固定为 20m, 节点数固定为 100, 随机选取了 100 对节点间的恒定比特流作为应用负载,通过改变 pause time 的值来观察两种算法在不同移动性下的路由性能表现。

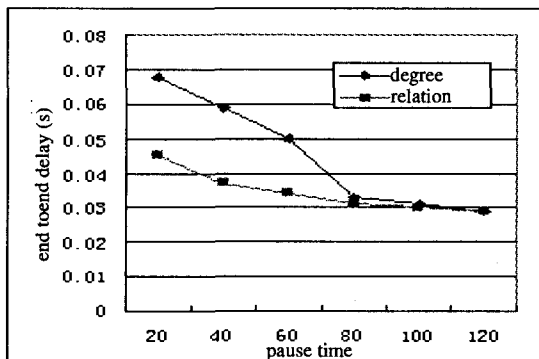


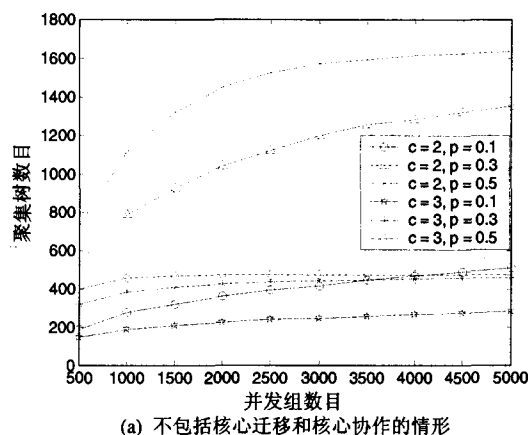
图 3(b) 平均端到端延时

个簇路由协议进行比较,所以这两个指标也从侧面反映了分簇路由协议的性能表现。

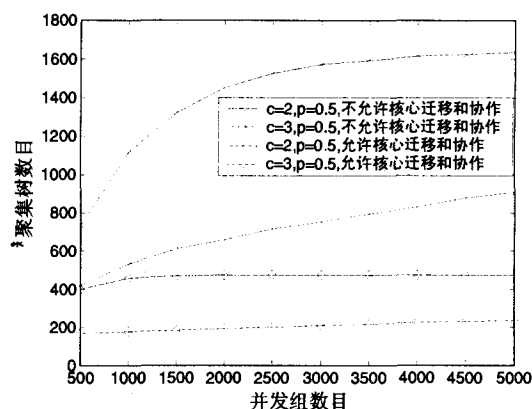
(下转第 37 页)

度的不同而有明显差异。随着组密度的增加,分发树数目减少率增加,会话密度相同时,BEAMBTs 获得的状态减少率明显高于 BEAM。当会话密度达到 0.5,并发组数目为 5000 时,BEAM 获得的 TSORR 仅为 0.38%,而此时 BEAMBTs 则获得 89.80%。

图 4 中比较了 BEAM 和 BEAMBTs 都进行完全匹配时,传输节点转发状态减少率 SRR(State Reduction Rate)与并发组数目及会话密度之间的关系。从图中可以看到,SRR 和 TSORR 曲线有相同的趋势,随并发组增加而加大。采用 BEAMBTs 协议时,SRR 受组密度的影响明显小于 BEAM 协



(a) 不包括核心迁移和核心协作的情形



(b) 包括核心迁移和核心协作的情形

图 5 包含 3 个 RP 核心时核心迁移和协作对聚集分发树数目的影响

结论 为了解决多播状态的可扩展性问题,本文在基于分发树切分的聚集多播的基础上,提出了一种基于分发树切分的分布式聚集多播协议 BEAMBTs,详细描述了工作过程,BEAMBTs 协议简单,易于实现,从仿真结果可以看到,当骨干网络中并发组数目较大时,BEAMBTs 协议可以取得比 BEAM 更好的聚集效果。允许核心迁移和协作的 BEAMBTs 协议极大地缩减了网络中可能需要维护的分发树数目,充分发挥了聚集多播的优势,对大规模并发的多播组应用而言具有良好的可扩展性,是一个对传输域非常有效的多播实现方案。

参考文献

- 1 Almeroth K. The evolution of multicast: From the Mbone to inter-domain multicast to Internet2 deployment. *IEEE Network*, 2000,14(1):10~20
- 2 El-Sayed A, Roca V, Mathy L. A survey of proposals for an alternative group communication service. *IEEE Network*, 2003

(special issue on Multicasting; an enabling technology)

- 3 Costa L H M, Fdida S, Duarte O C M. Hop-by-hop multicast routing protocol. In: *SIGCOMM'01*, San Diego, CA. 2001. 249~262
- 4 Tian J, Neufeld G. Forwarding state reduction for sparse mode multicast communications. In: *IEEE INFOCOM'98*, 1998. 711~719
- 5 Thaler D, Handley M. On the aggregatability of multicast forwarding state. In: *IEEE INFOCOM'00*, 2000. 1654~1663
- 6 Radoslavov P I, Estrin D, Govindan R. Exploiting the bandwidth-memory tradeoff in multicast state aggregation. Technical report, USC Dept. of CS Technical Report 99-697 (Second Revision), July 1999
- 7 Cui Jun-Hong, Kim Jinkyu, Maggiorini D, et al. Aggregated Multicast — A Comparative Study. In: the special issue of Cluster Computing: The Journal of Networks, Software and Applications, 2003
- 8 刘志峰, 宴文华. 一种基于分发树切分的聚集多播方案. *计算机研究与发展*, 2004,41(11): 1895~1901
- 9 Liu Zhifeng, Dou Wenhua, Liu Yajie. AMBTs: A Scheme of Aggregated Multicast Based on Tree Splitting. In: *IFIP Networking'04*, Athens, Greece, 2004. 829~830

(上接第 29 页)

簇算法的网络结构的稳定性和拓扑管理性能的高低。从图 3(a)和图 3(b)可以看出,相关度算法比节点度算法有着更好的表现,尤其是在 pause time 比较小即节点移动性比较大的情况下,相关度算法在端到端延时上性能提高显著。

结束语 移动自组网的移动特性导致了节点间拓扑关系的变化,在分层次的网络结构中节点间的相互关系直接影响着网络节点的角色和簇结构。本文首先提出了节点间相关度的定义,并在相关度的基础上提出了节点相关密度的概念。利用相关度和相关密度作为分簇的依据,提出了一种分布式的分簇算法,并对此算法进行了分析和模拟测试。由于充分考虑了节点周围的节点分布情况,测试结果表明,同节点度算法相比,该算法具有更好的分簇均衡性,也在应用的吞吐率、

延时方面具有更好的性能。

参考文献

- 1 Kawadia V, Kumar P R. Power Control and Clustering in Ad Hoc Networks. In: *Proc. Infocom*, 2003
- 2 王海涛, 张学涛. ad hoc 网络中的分簇算法. *数据通讯*, 2003,3:32~34
- 3 Gerla M, et al. Mobile, Multimedia Radio Network. *Wireless Networks*, 1995,1(3):255~265
- 4 Chatterjee M, Das S K, Turgut D. WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks. *Journal of Cluster Computing*, Special issue on Mobile Ad hoc Networking, 2002, (5):193~204