

一种基于移动预测的自适应 Ad hoc 网络分簇算法^{*})

蒋 毅 史浩山

(西北工业大学电子信息学院 西安 710072)

摘 要 在移动 Ad hoc 网络中采用分簇机制可以达到在拓扑结构频繁变化的网络环境中快速部署节点通信的目的。本文在深入研究移动预测技术的基础上,分析已有分簇算法存在的诸多不足,提出了一种基于移动预测的自适应分簇算法,包括分簇算法和分簇保持策略两方面的内容,并通过模拟证明该算法保持了分簇结构的稳定性,提高了网络负载平衡能力,减少了节点之间的通信路由开销,有效防止了通信的中断,弥补了已有分簇算法的不足。

关键词 移动预测, 自适应, Ad hoc 网络, 分簇算法

An Adaptive Clustering Algorithm Based on Mobility Prediction in Ad hoc Networks

JIANG Yi SHI Hao-Shan

(Northwestern Poly-technical University, Xi'an 710072)

Abstract Using clustering mechanism in mobile Ad hoc networks with variable topology framework, it can archive the purpose of deploying the communication of nodes rapidly. Based on studying deeply in the technology of mobility prediction, analyzing the shortages in clustering algorithm previously, this paper presents an adaptive clustering algorithm based on mobility prediction in mobile Ad hoc networks. It contains tow aspects: clustering algorithm and clustering maintenance, compensating shortages in clustering algorithm previously, keeping stability of clustering framework, increasing ability of Load-Balancing, reducing spending of communication routing between nodes, avoiding the communication interrupt through simulating.

Keywords Mobility prediction, Adaptive, Ad hoc networks, Clustering algorithm

1 引言

移动 Ad hoc 网络是由不需要固定基础设施支持的、移动节点组成的自组织无线通信网络。主要应用于一些临时性或军事紧急场合。因为每个节点的传输功率有限,所以节点之间的通信需要借助于其它节点的转发。节点同时承担主机和路由器两种功能。针对无线 Ad hoc 网络多跳、移动等特点,要实现整个网络的快速部署以及拓扑结构变化后的动态重建,最有效的方法是采用分簇机制。利用分簇算法构造网络结构,可以使网络的规模不受限制,可扩充性好,路由和控制开销较少,并且容易实现移动性管理和网络的局部同步,提供了一定的业务服务质量(QoS)。在分簇结构中,可以采用分级路由算法,簇内采用主动式路由算法,节点维护到簇内其它节点的完整路由信息,簇间使用按需式路由算法,来减少通信和路由开销。由此可以看出,通过分簇算法将网络划分成簇,可以在很大程度上提高 Ad hoc 网络的性能,具有极其重要的意义。

本文提出了一种基于移动预测的自适应分簇(Mobility prediction MP)机制,包括基于移动预测的自适应分簇算法和分簇保持策略。模拟结果表明,该算法可以有效地保持分簇结构的稳定性,提高网络的负载平衡能力^[1],减少节点之间的通信和路由开销,有效地防止了通信的中断。

2 相关研究

目前,国内外对分簇算法进行了深入的研究,但每种算法

都只考虑了网络性能的某一方面,所以需要一种综合考虑各个因素的分簇算法。与相关研究相比,文[2]最小 ID 算法(LID)是一个最简单的分簇算法,它选择相邻节点中 ID 最小的节点作为簇头,其一跳邻节点为该簇的成员节点。这种算法简单、实现方便、算法收敛较快,但簇头的数目较大,增加了分组投递的时延,由于簇内节点分布不均,簇头的更新频率较高,且没有考虑网络负载平衡。文[3]最高节点度算法(HD)选择相邻节点中具有最高度数的节点作为簇头。这种算法产生的簇头数目较少,即源和目的节点之间的平均跳数较少,从而减少了分组投递的时延,也使信道空间的重用率降低,但簇头分布不合理。当节点移动性较强时,簇头更新率上升,增加了网络的维护开销。

本文提出的基于移动预测的自适应分簇算法(MP),很好地解决了文[2]和文[3]的不足。算法是以节点之间链路的连接强度作为选择簇头的标准,而连接强度的计算要利用移动预测技术,本文 3.1 节会对这种技术给出具体的论述。该算法可以有效地控制簇头的数目,减少分组投递的时延,减轻网络计算的负担。在节点移动性较强时,可以明显减少簇头的更新率,提高网络的负载平衡能力。在第 4 部分的算法模拟中可以得出以上结论,所以 MP 算法非常适合于 Ad hoc 网络的分簇。

3 基于移动预测的自适应分簇算法

分簇算法的目标就是构造一个能够覆盖整个网络的、可以较好支持资源管理的簇的集合。为了减少分簇算法带来的

^{*})教育部博士点基金项目“高效能移动智能体空地一体化系统及其在战术网中的应用研究”(20050699037)。蒋 毅 博士研究生,主要研究方向:数据通信与计算机网络、移动 Ad hoc 网络和无线传感器网络;史浩山 教授,博士生导师,主要研究领域:多媒体通信、计算机通信网络。

通信计算开销,分簇算法应该简单高效。在只有很少的节点移动和拓扑结构变化较慢时,该分簇机制应尽量保持簇的原有结构,从而减少重新生成簇所引入的开销,提高网络的总体性能。所以,分簇机制包括两方面的内容:分簇算法;分簇保持策略^[4]。

3.1 移动预测

为了更好地掌握高动态性的 Ad hoc 网络的拓扑变化情况,本文使用移动预测的思想来反映各节点之间的连接强度(此处用链路维持时间的长短来反映连接强度)。假设 Ad hoc 网络中的节点均处于一个自由空间传播模型当中,信号强度只与其传播距离有关;各节点对信号的转送能力一致,且传播的有效范围相同;每个节点的时钟都是同步的(可以用 NTP(网络时间协议)或 GPS 的时钟定位系统来获得同步),那么对于任意时刻的两个相邻节点之间的距离不大于有效的传播范围,即可认为此两点保持连接。基于上述思想,通过 GPS 获得 Ad hoc 网络各节点的坐标、运动速度、运动方向,利用以下公式,就可以预测出任意两节点的链路连接时间 LET (Link Expiration Time)^[5,6]:

$$LET = \frac{-(ab+cd) + \sqrt{(a^2+c^2)r^2 - (ad-bc)^2}}{a^2+c^2} \quad (1)$$

式中: $a = v_i \cos \theta_i - v_j \cos \theta_j$, $b = x_i - x_j$, $c = v_i \sin \theta_i - v_j \sin \theta_j$, $d = y_i - y_j$ 。

r 为节点 i 和 j 的有效传输距离, (x_i, y_i) 和 (x_j, y_j) 分别为节点 i 和 j 的坐标。 v_i 和 v_j 为节点 i 和 j 的平均移动速度, θ_i 和 θ_j 为节点 i 和 j 的移动方向 ($0 \leq \theta_i, \theta_j < 2\pi$)。当 $v_i = v_j$, $\theta_i = \theta_j$ 时,即两个节点的运动方向一致, $LET \rightarrow \infty$, 此时认为相邻两节点可以一直保持连接状态;如果 $LET < 0$, 则认为相邻两节点不可以连接; $LET > 0$ 且值越大,表示连接强度越大。在 Ad hoc 网络中,随着网络节点的频繁移动,网络的拓扑结构在不断发生变化,各节点之间的链路连接情况也在不断地改变着。

3.2 基于移动预测的自适应分簇思想及算法

为了适应节点的频繁移动特性,提高分簇结构的稳定性,防止由于网络节点的移动导致分簇结构的改变,从而带来的重新分簇时出现过多计算,簇头的选择是十分重要的。在 Ad hoc 网络中,节点的电源能量是非常有限的。正确地选择簇头来划分簇,可以减少由于重新分簇带来的过多计算,从而节省了整个网络的能量。本文提出的基于移动预测的分簇思想将选择一些与其邻节点连接强度较大的节点做为簇头。连接强度用链路连接时间来表示,链路连接时间的计算可由 3.1 节的公式(1)进行计算得到。

具体的分簇思想如下。

Step1: 若本节点与其邻节点连接强度的平均值 $ALET$ ($\sum LET / \text{节点度}$) 比其邻节点的 $ALET$ 大,则宣布成立新簇,簇号为本节点的 ID; 若在 $ALET$ 相等的情况下,选择节点度大的节点作为簇头;若节点度也相等,则选择 ID 号最小的节点;否则执行 Step2 和 Step3。

Step2: 一直等待,直到收到邻节点中有宣布成立新簇的控制消息,并加入此簇。

Step3: 若发现比自己 $ALET$ 小的邻节点都已宣布加入其它簇了,而自己还没有加入任何簇,则宣布自己成立一个新簇。

Step4: 为了控制簇的规模,对每个簇所能包含的节点数规定了上限 H 和下限 L 。如果一个簇所包含的节点数少于

L , 则将其归入邻近的一个簇,这个过程称为簇的合并;如果一个簇所包含的节点数大于 H , 就要将其分解为多个簇,这个过程称为簇的分解。

控制簇规模的原因是簇规模较小时,过多的簇可能增加分组投递的时延,路由时每个节点维护的路由表信息过多,会造成网络资源的浪费。而簇规模较大时,簇数太少又起不到分簇的作用。

Ad hoc 网络可以由一个无向图 $G=(V, E)$ 来表示, V 表示节点的集合, E 表示链路的集合, $v_i \in V$ (v_i 表示 ID 号为 i 的节点), $(v_i, v_j) \in E$ ((v_i, v_j) 表示节点 i 和 j 之间的链路), $C_i \subset V$ (C_i 表示编号为 i 的簇)。为了更好地描述簇结构还需要有 3 点假设^[7]: 1) 对于簇中的任意两节点之间的距离 $d(i, j) \leq 2$ 。2) $V = \bigcup C_i$ 且 $C_i \cap C_j = \emptyset$ ($i \neq j$) 即簇之间是不交叠的。3) 分簇算法执行期间网络的拓扑结构不发生改变。

基于移动预测的自适应分簇算法由如下伪代码表示:

```

Step1: if ( ALET(i) > ALET(j) ) //j 为节点 i 的邻节点, 比较节点 i 和 j 的 ALET 的值
{
    ALET(i) =  $\sum LET(v_i, v_j) / i$  度数
    clusterhead = i; // 选择节点 i 为簇头
    send cluster (CH(i)); // 向其邻节点广播成簇的消息
    V = V - {i}; // 删除已属于某簇的节点
}
else if ( ALET(i) == ALET(j) ) // 邻节点 j 的 ALET 值与 i 的相等
{
    if (i 的节点度 > j 的节点度) // 比较节点 i 和 j 的节点度
    {
        clusterhead = i;
        send cluster (CH(i));
        V = V - {i};
    }
    else if (i 的节点度 == j 的节点度)
    {
        if ( ID(i) < ID(j) ) // 比较节点 i 和 j 的 ID 大小
        {
            clusterhead = i;
            send cluster (CH(i));
            V = V - {i};
        }
    }
} // 反复进行上述过程, 直到 V 为空, 所有节点入簇

Step2: receive cluster(j); // 收到邻节点 j 成簇的消息, j 为簇头
Step3: if ( ALET(i) < ALET(j) ) // 如果宣布成簇的邻节点的 ALET 比本节点大
{
    Join (i, j); // 加入邻节点的簇
}
else
{
    clusterhead = i;
    send cluster (CH(i));
    V = V - {i};
} // 若宣布成簇的邻节点的 ALET 比本节点 i 小, i 不属于任何簇, 则自己成立一个新簇

Step4: if ( cluster (i) < L ) // 簇 i 所包含的节点数少于下限 L
{
    Join (i, j); // 簇合并, i 加入临近的簇, 必须保证新生成的簇满足假设 1)
}
Else if ( cluster (i) > H ) // 簇 i 所包含的节点数多于上限 H
{
    divide (i, j); // 簇分解, 选择 i 邻节点中 ALET 最大的 j 自成一簇
    clusterhead = j;
    send cluster (CH(j));
}

```

该分簇算法,在节点移动性较强的 Ad hoc 网络中,可以减少簇头的更新频率。模拟证明,这种算法比最小 ID 法具有较低的簇头更新率,比最高节点度算法具有较高的负载平衡能力,适当减少了簇头的数目,由此源和目的节点之间的平均跳数较少,从而减少了分组投递的时延。

例如,某网络的拓扑结构如图 1,用上述基于移动预测的自适应分簇算法处理后的分簇结果如图 2。

3.3 基于移动预测的自适应分簇保持策略

在网络节点频繁移动的 Ad hoc 网络中,网络的拓扑结构

在不断地发生变化,随之而来的是簇结构的改变,每次改变都会增加网络控制和通信的开销,降低系统的性能。本文提出了一种分簇保持策略,在一定程度上来维持簇结构的稳定性。

如果簇内节点之间的跳数超过 2 跳,就需要使用簇保持机制。分簇保持机制的基本原则:将一些稳定性高、连接强度大的节点依然留在原簇中,让一些不符合要求的节点离开簇,以维持簇的结构。具体方法如下:

- 若本节点 i 的一跳邻节点数 < 3 跳邻节点数, i 出簇;
- 若本节点 i 的一跳邻节点数 > 3 跳邻节点数, 3 跳邻节点出簇;
- 若本节点 i 的一跳邻节点数 $= 3$ 跳邻节点数, 则比较各自的 ALET 值, ALET 值大的留下, 小的离开。

通过判断节点之间的跳数来作为使用分簇保持策略的依据

据是十分有效的。因为用于判断跳数的信息可以很方便地从控制信息中最新获取,且方法是对称的、唯一的,所以保证了簇的绝大部分节点仍保留在原簇中。

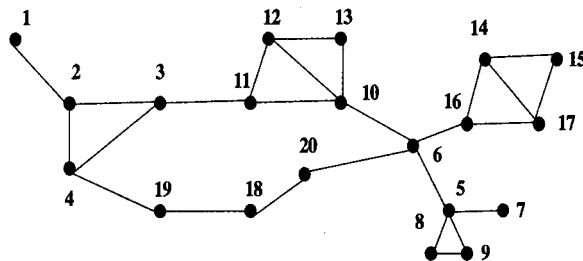


图1 某网络的拓扑结构图

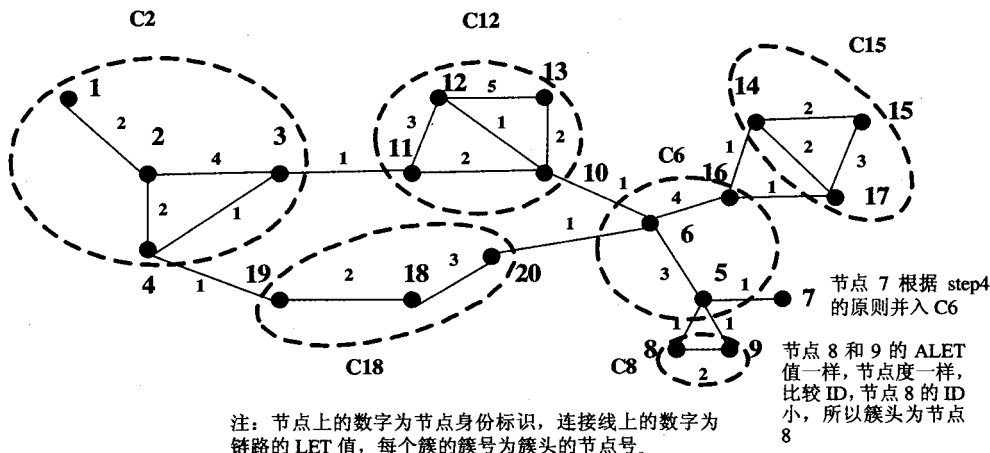


图2 用基于移动预测的自适应分簇算法处理后的分簇结果

4 算法性能仿真及分析

节点的传输功率将决定网络拓扑,从而影响分簇结构的性能。如果增加节点功率,源和目的节点之间的平均跳数减少,系统的吞吐量将会增加,但功率的增加也会增大节点间的干扰,从而限制吞吐量的增加。然而,较小的功率会导致较多的簇和较多的跳数,从而增加了网络的计算负担和分组投递时延。在仿真模拟中,将以节点传输功率作为一个参数来测试本文提出的基于移动预测的自适应分簇算法的网络性能。

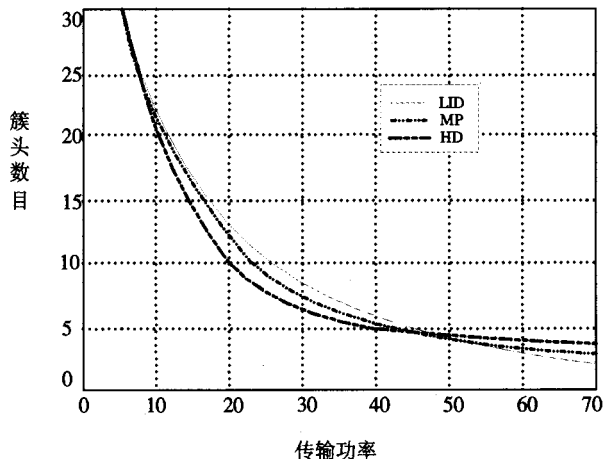


图3 簇头数目随节点传输功率变化曲线

模拟在一个 100×100 单位距离内进行,选取固定节点数目 40 个,节点的移动方向可以在 $(0, 2\pi)$ 内随机分布,节点的移动速度在 $[0, 5]$ 之间随机选择,单位是距离/时间。

首先,比较一下最小 ID(LID)、最高节点度(HD)分簇算法与本文提出的基于移动预测的自适应分簇算法(MP)的簇头数目随节点传输功率变化的情况,模拟结果如图 3。3 种算法中簇头的数目都随着节点传输功率的增加而减少,并且当传输功率大于 30 后,簇头数目的变化缓慢,基本趋近于 1;当传输功率很小的时候,每个节点各成一簇,所以簇的数目很大。模拟结果表明,节点的传输功率越大,簇的覆盖范围也越大。从图中可以发现 3 种分簇算法的区别,相同传输功率的情况下,HD 算法的簇头数目明显低于其它算法,LID 最高,MP 居中,这与本文所论述的各算法的分簇规则是吻合的。

其次,比较一下簇头更新率随节点传输功率变化的情况。模拟结果如图 4(簇头更新率 $P = \frac{N - n_c}{n_c}$, 其中 N 是整个网络节点数, n_c 是簇头的数目)。从图上可以看到,所有算法的变化规律是相同的。在节点传输功率较小时,簇的覆盖范围小,节点容易出簇。当传输功率增加后,节点出簇的机会就会减小,簇头的更新率也会减小。从各个算法模拟结果的差别可以看出,HD 算法选簇时只考虑了节点的度数,簇头分布不合理,簇头很容易出簇,所以相同传输功率的情况下 P 值最高;由于 MP 是基于移动预测的自适应分簇算法,选择簇头时是以节点之间链路的连接强度为依据,以 HD\LID 算法为辅,

(下转第 34 页)

based heterogeneous networks. IEEE Personal Communications, 2001, 8 (6): 18~24

3 全贤唐, 张健. 经济博弈论. 北京: 机械工业出版社, 2003

4 王兴伟, 王志军, 黄敏, 等. 基于服务质量的多媒体通信初始路由建立算法的研究. 计算机学报, 2001, 24 (8): 830~837

5 杨纶标. 模糊数学原理及应用. 第三版. 广州: 华南理工大学出版社, 2002

6 胡杨梅, 张金水. Internet 拥堵的差别定价策略研究. 数量经济技术经济研究, 2004, 35 (7): 81~85

7 靳晓强. 双向 Dijkstra 算法及中间链表加速方法. 计算机仿真,

2004, 21 (9): 78~81

8 Fall K, Varadhan K. The ns Manual. <http://www.isi.edu/nsnam/ns/doc/>

9 徐雷鸣, 庞博, 赵耀. NS 与网络模拟. 北京: 人民邮电出版社, 2003

10 Wang Xingwei, Yuan Changqing, Huang Min. A fuzzy-tower-based unicast QoS routing algorithm. In: Proceedings of EUC'04, 2004, 988~997

11 Waxman B M. Routing of multipoint connections. IEEE Journal on Selected Areas in Communications, 1988, 6 (11): 478~489

(上接第 29 页)

所以簇头的稳定性较强, 分布也比较均匀, 并且可以根据节点的分布自适应的调节簇结构, 所以 P 值在 3 种算法中最低; LID 算法居于两者之间, 表明虽然用 LID 算法分簇, 簇头比较稳定, 但其分布还是不够均匀。

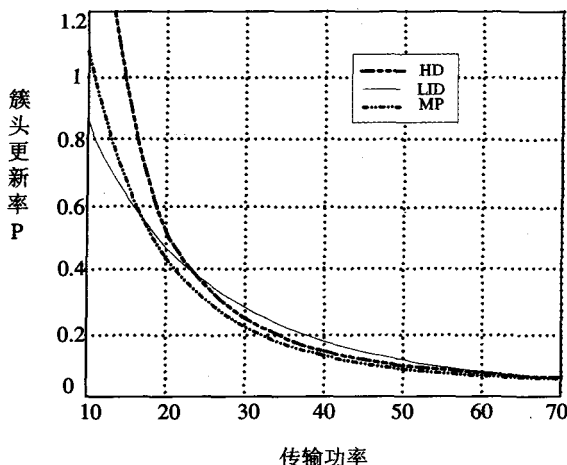


图 4 簇头更新率 P 随节点传输功率变化曲线

最后, 比较一下负载平衡因子 LBF 随节点传输功率变化

的情况。模拟结果如图 5 ($LBF = \frac{n_c}{\sum_i (x_i - p)^2}$, 其中 x_i 是簇 i

的成员节点数)。从图中可以看出 3 种算法的变化趋势相同。当节点的传输功率较小时, 每个节点都单独成簇, 此时的 LBF 很大; 当节点的传输功率增大时, 簇头数目减少, 网络的负载平衡能力开始下降; 当节点的传输功率足够大时, 簇的数目几乎为 1, 这时 $LBF \rightarrow \infty$, 此时网络的连通性较好, 但簇头数目过少, 会增加计算开销, 空间重用率降低。从图中还可以看出 3 种算法的区别, MP 算法的 LBF 明显高于其它算法, 这是由于 MP 算法是以移动预测技术为依托, 以链路的连接强度为标准来进行分簇, 簇结构的稳定性强, 省去了许多更新簇头的开销; LID 算法由于产生的簇头数目较多, 更新频率快, 没有考虑负载平衡能力, 所以居中; HD 算法由于簇中成员分布不均匀而导致了较低的 LBF。由此可以看出, MP 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。

结论 基于分簇算法的分簇网络结构对于提高 Ad hoc 网络的性能有着重要的意义, 已经开始得到广泛的研究, 本文提出的基于移动预测的自适应分簇算法, 可以有效地弥补已有很多分簇算法只是考虑某一方面网络性能的缺点, 多方位

地考虑了影响网络性能的众多因素, 在保持分簇结构的稳定性和提高负载平衡能力等方面都收到了很好的效果。由于分簇结构既有利于移动管理、资源分配和信道接入, 又可以比较方便地采用主动式和按需式优点的混合式路由, 而且目前仍处于研究和模拟阶段, 所以分簇机制在实际网络中将有很大的应用前景。

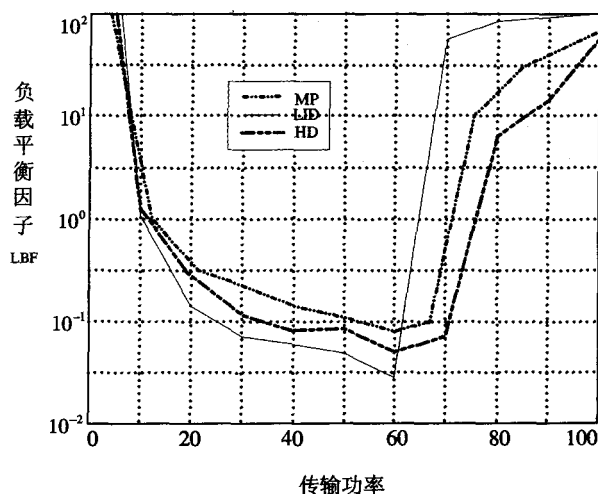


图 5 负载平衡因子 LBF 随节点传输功率变化曲线

参考文献

- 1 Amis A D, Prakash R. Load-Balancing Clusters in Wireless Ad Hoc Networks. In: Proceedings of ASSET 2000, March 2000
- 2 Lin C H R, Gerla M. A Distributed Architecture for Multimedia in Dynamic Wireless Networks [J]. IEEE GLOBECOM, 1995, 1468~1472
- 3 Gerla M, Tsai J T C. Multicluster, Mobile, Multimedia Radio Network [J]. Wireless Networks, 1995, 1(3): 255~265
- 4 Basagni S. Distributed Clustering for Ad Hoc Networks [A]. In: International Symposium on Parallel Architectures, Algorithms and Networks [C]. Perth, June 1999. 310~315
- 5 Su W, Lee S J, Gerla M. Mobility prediction in wireless networks. In: 21st Century Military Communications Conference Proceedings, Vol 1, 2000. 491~495
- 6 Chellappa R, Jennings A, Shenoy N. A comparative study of mobility prediction in fixed wireless networks and mobile ad hoc networks. In: IEEE International Conference on, Vol 2, 2003. 891~895
- 7 王海涛, 张学平. Ad hoc 网络中的分簇算法. 数据通信, 2003(4): 32~35