

面向高动态移动自组织网络的 生物启发分簇算法

于云龙¹, 茹乐², 方堃², 贾旭峰²

(1. 空军工程大学防空反导学院, 陕西西安 710038; 2. 空军工程大学航空航天工程学院, 陕西西安 710038)

摘要: 分簇可以有效地提高大规模移动自组织网络的性能. 但高动态的移动自组织网络具有节点移动性强、网络拓扑变化快的特点, 应用传统的分簇算法会造成网络性能迅速下降, 频繁的簇拓扑更新造成了簇结构的不稳定和开销的增加. 为了解决传统分簇算法无法适应高动态的大规模移动自组织网络的问题, 提出了一种基于生物启发的移动感知分簇算法, 该算法对多头绒泡菌的觅食模型进行了改进, 使其适用于移动自组织网络领域. 由于该算法与节点的移动特性进行了结合, 所以该算法可以有效地在高动态移动自组织网络中进行簇的建立与维护. 实验结果表明, 相较于其他传统分簇算法, 本文算法提高了平均链路连接保持时间和平均簇首保持时间, 使得簇结构更加稳定, 提高了对高动态、大规模移动自组织网络的适应能力.

关键词: 移动通信网络; 仿生算法; 移动感知; 高动态

中图分类号: V271.4

文献标识码: A

文章编号: 0372-2112 (2018)04-0918-12

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2018.04.021

Bio-Inspired Clustering Algorithm for Highly Dynamic Mobile Ad Hoc Networks

YU Yun-long¹, RU Le², FANG Kun², JIA Xu-feng²

(1. School of Air and Missile Defense, Air Force Engineering University, Xi'an, Shaanxi 710038, China;

2. College of Aeronautics & Astronautics Engineering, Air Force Engineering University, Xi'an, Shaanxi 710038, China)

Abstract: Clustering can increase the performance of large-scale mobile ad hoc networks effectively. But the highly dynamic mobile ad hoc networks have some of characteristics, such as high mobility and fast network topology change; applying traditional clustering algorithms will cause the sharp decrease of the performance of the network, frequent updates of cluster topology will cause the instability of cluster structure and the increase of control overhead. For purpose of solving the problems that traditional clustering algorithms cannot fit the highly dynamic large-scale mobile ad hoc networks, BIMAC (Bio-Inspired Mobility-Aware Clustering) algorithm is proposed. This algorithm ameliorates the forage model of physarum polycephalum, which can make it adapt to the domain of mobile ad hoc networks. On account of this algorithm includes the mobility characteristic of the node, we can carry through the cluster formation and maintenance effectively. Experimental results have indicated that the BIMAC algorithm increases average link connection lifetime and average cluster head lifetime compared with other traditional clustering algorithms; BIMAC algorithm can make the cluster structure more stable. This algorithm can increase the adaptive capacity for highly dynamic large-scale mobile ad hoc networks.

Key words: mobile communication networks; bionic algorithm; mobility-aware; highly dynamic

1 引言

当前, 移动自组织网络的应用越来越广泛, 已涵盖到民用和军用多个领域^[1,2]. 当移动自组织网络的分布范围较广, 且节点数量较多时, 如果采用传统的方法来

维护如此大规模的移动自组织网络, 必然会造成性能的下降. 分簇是一种有效的网络拓扑管理模式, 它提供的分级网络结构可以提高移动自组织网络的扩展性, 降低网络管理的复杂度, 十分适用于大规模的移动自组织网络^[3,4].

在一些场景中,节点的移动速度较高且变化较快,因此形成了高动态的移动自组织网络.高动态移动自组织网络主要在航空领域起重要作用,该网络可将航空飞行器互连互通,可满足特定条件下的军、民航通信的需求^[5].目前,高速无人机技术飞速发展,高动态自组织网络在无人机领域已成为研究热点.高动态移动自组织网络与移动自组织网络相比,具有节点移动性更强、网络拓扑结构变化更快等特点,因此,节点之间的链路会很不稳定.具有移动意识的分簇算法正是利用节点的移动特性来进行簇的建立和维护,其主要思想为:将具有相似移动速度和移动方向的节点划分到同一个簇内,这样簇内的节点可以建立稳定的链路,从而减少链路或簇的重新建立.

文献[6]提出的 MOBIC 算法使用两次接收到的消息功率来估计相对移动,并没有精确的计算两节点间的距离和相对移动速度,因此,它只适用于节点的移动速度和方向都大致相同的场景,如高速公路.在高动态移动自组织网络中,各个节点运动速度和方向都是时变的,因此 MOBIC 算法并不适用.文献[7]提出了 CEMCA 算法,该算法考虑了节点的连接度、剩余能量和节点的移动性,并对上述三个指标进行加权求和.该方法通过调整权值的大小,可以适应不同的场景,但是,它对节点移动性处理方法的不完善依然无法满足高动态的移动自组织网络.文献[8]针对高动态的移动自组织网络提出了 MPCR 分簇路由算法,其中的分簇算法通过 GPS 提供的位置信息计算两节点间的链路保持时间预测值,进而进行移动性预测并计算出链路连接可能性,具有最大一跳邻居链路连接可能性之和的节点将被选为簇首.文献[9]将 GPS 提供的位置信息带入到马尔科夫模型中,从而可以对节点的移动性进行预测.但上述方法容易引入 GPS 的定位误差以及外界对 GPS 信号的干扰,造成移动参数不准确,使得算法失效.文献[10]提出的 MACA 算法可以对节点进行移动性预测,但是它仅适用于移动性较弱的无线传感器网络.

本文将利用接收到的 Hello 消息的信号特征来进行节点的移动性预测,该做法不但可以减少对 GPS 的依赖,同时能够充分的利用 Hello 消息所能提供的信息,从而充分地利用系统资源,避免了资源浪费.文献[11]提出的 SECA 算法将信号强度作为分簇的一个重要标准,但是信号强度并不能够完全的预测某一节点的移动性.文献[12]提出的 MPBC 算法通过采集 Hello 消息的信号强度和 Doppler 频移,计算出节点间的相对移动速度,与邻居节点具有最小相对速度的节点将成为簇首.但文献[12]在公式推导过程中使用了近似处理方法,使得相对速度的计算并不准确,容易造成判断的失误.文献[13]提出的 DDVC 算法根据 Hello 消息的

多普勒频移计算出节点的相对径向移动速度,然后可以用于簇的建立与维护.但是,该算法仅适用于节点移动速度变化不频繁的场景.

近年来,多头绒泡菌已经成为生物启发算法的研究重点.多头绒泡菌是由管状形体和流动在管状形体中的液体组成,这些流动的液体为多头绒泡菌各部分运送营养和传输信号,同时也调节管状形体发生改变.文献[14,15]根据多头绒泡菌的觅食行为提出了一个与此相适应的数学模型.文献[16,17]将多头绒泡菌觅食行为的数学模型引入到无线传感器网络领域,并用于路由算法中下一跳节点的选择.文献[18]利用多头绒泡菌数学模型解决现有互联网路由策略静态、僵化的问题.目前,没有文献将该数学模型引入到分簇算法中.

本文结合高动态移动自组织网络的特征,将多头绒泡菌的觅食模型引入到移动自组织网络领域,并与节点的移动特性相结合,提出了基于生物启发的移动感知分簇(BIMAC)算法.改进后模型的输出值可以为簇的建立和簇的维护提供准确的依据.

2 系统模型

2.1 移动模型

文献[19]提出的移动模型基本可以满足本文所需移动场景的要求,其中包含群组的拆分和合并,但仍有需要改进的方面.本文对上述模型进行了修改,使其可以模拟高动态网络中单个节点和群组共存的移动场景,其包含的核心要素为:

(1)多个节点将随机的分布在仿真区域内,同时在仿真区域内随机产生几个互不相交的圆,半径通常与节点的通信半径相等.本文定义,在同一个圆内的节点将形成一个群组,在所有圆形之外的节点将视为独立的群组.

(2)每个群组都会被分配一个目的区.群组内的每个节点分别在其对应的目的区内随机选择一个位置作为目的点,然后以不同的速度向其目的点移动.当一个群组内所有的节点都到达相应的目的点后,群组新的目的区将产生,重复(2)的步骤.

(3)以随机的时间间隔,产生一个参考点.距离参考点最近的一个群组将被分割为两个更小的群组.然后目的区将会被分配至上述两个群组,重复(2)所述的步骤.

(4)以随机的时间间隔,距离最近的两个目的区,其对应的群组会合并为一个更大的群组,然后,新的目的区会分配给该群组,重复(2)所述的步骤.

2.2 多头绒泡菌觅食模型及其改进

根据文献[14],可以得到多头绒泡菌的觅食模型为:

$$Q_{ij} = \frac{\pi r_{ij}^4 (P_i - P_j)}{8\eta L_{ij}} = \frac{D_{ij} (P_i - P_j)}{L_{ij}} = \frac{D_{ij} \Delta P_{ij}}{L_{ij}} \quad (1)$$

其中, Q_{ij} 是单位时间内通过管状体 ij 的流体通量, $\Delta P_{ij} = P_i - P_j$ 是多头绒泡菌的管状体两端流体的压差, r_{ij} 是管状体的半径, η 是管状体内流体的粘度, $D_{ij} = \pi r_{ij}^4 / 8\eta$ 是管状体的流通性, L_{ij} 是管状体的长度.

式(1)代表了管状体 ij 的流体通量主要受管状体流通性 D_{ij} 、管状体两端流体压差 ΔP_{ij} 和管状体长度 L_{ij} 的影响. 管状体 ij 的流通性越好且两端压差越大, 则管状体的流体通量越大; 然而, 管状体 ij 的长度越长, 其流体通量就越小.

本文假设每个节点的容量为 0, 则每个节点的守恒定律可由下式计算:

$$\sum_j Q_{ij} = \begin{cases} I, & i = s \quad j \in N_i \\ -I, & i = d \quad j \in N_i \\ 0, & \text{others} \quad j \in N_i \end{cases} \quad (2)$$

其中, I 是从源节点流出的流体通量 (或流入汇聚节点的流体通量). 在多头绒泡菌模型中, I 是恒定值, 同时也说明了总的流体通量在整个过程中是恒定常数.

式(2)说明了每个节点流体通量的关系. 对于源节点 s , I 是从其中流出的流体通量; 对于汇聚节点 d , I 是流入其中的流体通量; 对于中间节点, 流入和流出的总和为 0.

由于多头绒泡菌体内流体的总量保持不变, 它通过调节身体的形状寻找食物, 调节模型为:

$$\frac{d}{dt} D_{ij} = f(|Q_{ij}|) - \delta D_{ij} \quad (3)$$

其中, δ 是管状体的衰减率, $f(\cdot)$ 是单调递增函数并且满足 $f(0) = 0$, 该式表明管状体的流通性会随着通过它流体量的减小而减小.

式(1)、(2)和(3)中变量的物理含义均来自流体力学, 本文将多头绒泡菌觅食模型引入到移动自组织网络中. 因此, 需要将式(1)和式(3)中变量的物理含义和网络中的物理量相结合. 下面采用无量纲分析方法, 将式(1)和(3)迁移到移动自组网领域中.

首先, D_{ij} 是管状体的流通性, 是表示管状体对流体输送能力的物理参数. 同样, 自组织网络中也需要考虑链路的数据传输能力. 因此, 可以用带宽 B_{ij} 代替 D_{ij} , 以此来表示节点 i 与节点 j 之间链路的数据传输能力. 其次, L_{ij} 表示管状体的长度, 其数值越大, 对流体的输送越不利. 同样的, 在移动自组织网络中, 两相邻节点之间的距离也是影响数据传输的主要因素, 节点间距离越大, 受到的衰减以及干扰就越多, 对数据的传输就越不利. 因此, 可以用两相邻节点间距离来代替管状体长度, 也用 L_{ij} 表示. 再其次, ΔP_{ij} 表示管状体两端流体的压力, 压力越大, 对流体的输送越有利. 在移动自组织网络中, 两

相邻节点间链路生存可能性越大, 对数据的传输越有利. 同样的, 相邻节点间运动的稳定性也是影响数据传输的主要因素. 因此, ΔP_{ij} 值的大小受链路生存可能性和节点运动稳定性两方面因素的影响, 由于上述二者均是时变的, 所以用 $\Delta P_{ij}(t)$ 表示. 最后, Q_{ij} 是单位时间内通过管状体 ij 的流体通量, 在移动自组织网络中, Q_{ij} 表示可以通过链路 ij 的虚拟通信量. 则, 式(1)可以转化为式(4).

$$Q_{ij} = \frac{B_{ij} \Delta P_{ij}(t)}{L_{ij}} = \frac{B_{ij} (k_1 \times NS_{ij} + k_2 \times LSP_{ij})}{L_{ij}} \quad (4)$$

式中, L_{ij} 为节点 ij 之间距离的归一化值; NS_{ij} 代表相邻节点间的运动稳定性; LSP_{ij} 代表相邻节点间链路的生存可能性; k_1, k_2 代表权重系数, 它们的和为 1.

在多头绒泡菌模型中, 由于 D_{ij} 随着流体在管状体中流动而变化, 因此, 通过 D_{ij} 的进化实现多头绒泡菌的自适应调节. 在移动自组织网络中, B_{ij} 保持不变, $\Delta P_{ij}(t)$ 对于不同的链路是不相等的, 而且随着时间的变化而变化, 因此, 可以通过 $\Delta P_{ij}(t)$ 的进化实现分簇决策的自适应调节. 假设单调递增函数 $\varphi(Q) = Q^u$, 可以得到:

$$\begin{aligned} \frac{d}{dt} \Delta P_{ij}(t) &= \varphi(|Q|) - \delta \Delta P_{ij}(t) \\ &= \left(\frac{B_{ij} \Delta P_{ij}(t)}{L_{ij}} \right)^u - \delta \Delta P_{ij}(t) \end{aligned} \quad (5)$$

式中, δ 是 $\Delta P_{ij}(t)$ 的衰减率; u 是大于零的常数.

2.3 相邻节点间运动稳定性

根据自由空间衰减模型, 当前节点 i 接收到距离 d 处的邻居节点 j 的 Hello 消息的信号强度 P_{ij} 为

$$P_{ij} = P_t G_r G_t \left(\frac{\lambda}{4\pi d} \right)^2 \quad (6)$$

式中, λ 为无线电波的波长, G_r 是接收天线的增益, G_t 是发射天线的增益, P_t 为邻居节点 Hello 消息发射功率.

本文假设天线的覆盖范围为一个半径为 R 的圆形区域. 将接收到邻居节点 Hello 消息的信号强度的临界值定义为

$$P_{\text{threshold}} = P_t G_r G_t \left(\frac{\lambda}{4\pi R} \right)^2 \quad (7)$$

当前节点可以根据本节点已知信息和邻居节点 Hello 消息中包含的信息计算出接收信号强度的临界值, 并与实测的信号强度进行计算, 得到 $X_{ij}(t_k)$.

$$X_{ij}(t_k) = 1 - \frac{P_{\text{threshold}}}{P_{ij}} \quad (8)$$

根据文献[20], 可以得到比安内梅一切比雪夫不等式

$$P\{|X - E(X)| < \varepsilon\} \geq 1 - \frac{\text{var}(X)}{\varepsilon^2} \quad (9)$$

式中, X 为离散变量; $E(X)$ 为 X 的数学期望; $\text{var}(X)$ 为 X 的方差, ε 为任意正数.

如果 $\text{var}(X) = 0$, 那么 $P\{|X - E(X)| < \varepsilon\} = 1$, 说明了变量 X 与其期望值相等, 同时也说明了变量 X 的方差越小, 变量 X 越接近其期望, 变量 X 的变化量越小.

根据变量 X 的多次测量值, 可得到其方差 $\text{var}(X)$ 为

$$\text{var}(X) = \left(\sum_k \frac{X_k^2}{n} \right) - \left(\sum_k \frac{X_k}{n} \right)^2 \quad (10)$$

将 $X_{ij}(t_k)$ 不同时刻的值作为变量 X 的多次测量值, 并带入式(10)中, 便可以得到当前节点 i 接收到相邻节点 j 发出的 Hello 消息的功率变化情况, 进而可以判断出相邻节点 j 相对于当前节点 i 的运动情况, 得到相邻节点间运动的稳定性 NS_{ij} .

$$\text{var}(X_{ij}) = \left(\sum_k \frac{X_{ij}(t_k)^2}{n} \right) - \left(\sum_k \frac{X_{ij}(t_k)}{n} \right)^2 \quad (11)$$

$$NS_{ij} = 1 - \text{var}(X_{ij}) \quad (12)$$

$X_{ij}(t_k)$ 在某一时间内变化越大, $\text{var}(X_{ij})$ 就越大, 因此, NS_{ij} 就越小, 说明了该相邻节点间的运动稳定性较差; 反之, NS_{ij} 越大, 代表相邻节点间运动稳定性较好.

2.4 相邻节点间链路生存可能性

在移动自组织网络中, 每个节点周期地广播 Hello 消息, 同样的, 发出的数据信息同样可以起到和 Hello 消息相同的作用. 当前节点 i 可以接收到其邻居节点 j 发出的 Hello 消息或数据消息, 通过提取来波信号中的特征参数, 就可以计算出当前节点 i 与邻居节点 j 之间链路生存的可能性, 用一系数表示. 需要在来波信号中提取的参数主要包括: 来波频率、来波信号幅度.

利用两个连续发送的 Hello 消息的多普勒频移和信号幅度, 就可以计算出相邻节点间相对移动速度和移动方向, 进而计算出相邻节点间链路生存的可能性. 由于两个连续的 Hello 消息时间间隔 Δt 一般较小, 所以本文假设, 在 Δt 内, 节点的相对移动速度和移动方向都保持不变.

2.4.1 相对移动速度大小及方向的计算

图 1 所示的是当前节点 i 与其邻居节点 j 的相对移动模型. 事实上, 节点 i 与节点 j 都是移动的, 但是为了简化计算, 假设当前节点 i 是固定的, 其邻居节点 j 是移动的, 只需要计算两节点的相对移动速度即可. 如图 1 所示, 当邻居节点 j 由 b 运动到 c 时, 称该场景为“先接近再远离”场景, 原因为: 在 c 点之后, 节点 j 的运动趋势是一个先接近再远离当前节点 i 的过程, 因此称其为“先接近再远离”场景; 当邻居节点 j 由 c 运动到 b 时, 称该场景为“远离”场景, 在 b 点之后, 节点 j 的运动趋势是一直远离节点 i , 因此称其为“远离”场景. 当然, 本文所定义的这两个场景仅代表两个节点在当前时刻的运动趋势, 是建立在两个节点保持当前移动速度和方向的假设之上的, 事实上, 该两个节点的移动速度和方向是可能发生改变, 用上述方式划分场景其实并不

准确. 用上述方法划分场景只是为了满足后续评估当前链路生存可能性的需要.

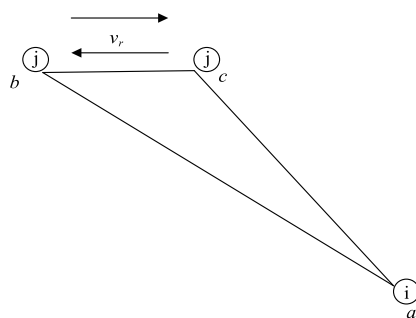


图1 相对移动模型

节点 j 在 b 、 c 两处各发送一次 Hello 消息, 并且为连续发送的 Hello 消息, Δt 是两次发送 Hello 消息的时间间隔. 节点 i 可以接收到 Hello 消息, 根据自由空间衰减模型, 节点 i 接收到节点 j 从 c 、 b 处传来消息的功率分别为

$$P_{ij}^c = P_t G_t G_r \left(\frac{\lambda}{4\pi d_{ac}} \right)^2 \quad (13)$$

$$P_{ij}^b = P_t G_t G_r \left(\frac{\lambda}{4\pi d_{ab}} \right)^2 \quad (14)$$

式中, d_{ac} 为 a 、 c 之间的距离; d_{ab} 为 a 、 b 之间的距离.

根据式(13)、式(14)可以得到 a 、 c 与 a 、 b 之间的距离.

$$d_{ac} = \frac{\lambda}{4\pi} \sqrt{\frac{P_t G_t G_r}{P_{ij}^c}} \quad (15)$$

$$d_{ab} = \frac{\lambda}{4\pi} \sqrt{\frac{P_t G_t G_r}{P_{ij}^b}} \quad (16)$$

假设 f 为节点 j 发出消息的载波频率, f_c 、 f_b 为节点 i 接收到节点 j 处于 c 、 b 两点时发出 Hello 消息的载波频率. 由于二者存在相对运动, 所以存在多普勒效应. 如果 $f_b > f$ 且 $f_c > f$, 表示节点 i 、 j 处于“先接近再远离”场景; 如果 $f_b < f$ 且 $f_c < f$, 表示节点 i 、 j 处于“远离”场景, 当然, 也存在一些特殊情况, 也应归属于此场景. 在图 2(a) 中, 节点 j 在 c 处的运动方向与 a 、 c 的连线垂直, 造成节点 i 、 j 之间的径向速度为零, 因此 $f_c = f$. 所以, 当 $f_c = f_b$ 有且仅有一个与载频相等时, 表示节点 i 、 j 处于“远离”场景. 当 $f_b < f$ 且 $f_c > f$ 或者 $f_b > f$ 且 $f_c < f$ 时, 表示节点 i 、 j 也处于“远离”场景, 如图 2(b) 所示. 最后, 当 $f_b = f_c = f$ 时, 节点 i 、 j 之间没有相对运动, $v_r = 0$.

(1) “先接近再远离”场景

对于连续接收的两个 Hello 消息, 通常以后一个消息为准进行相对速度的计算. 在图 1 中, 节点 j 从 b 运动到 c , 属于“先接近再远离”场景. 根据多普勒效应, 节点 i 接收到节点 j 在 c 处发送的 Hello 消息的载波频率 f_c 可以计算为

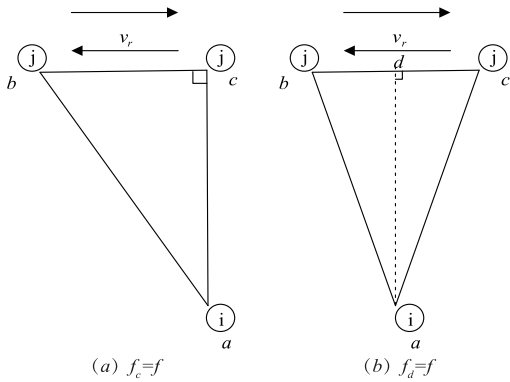


图2 “远离”场景特殊情况

$$f_c = f \frac{c - v_r \cos \angle acb}{c} \quad (17)$$

式中, c 代表光速; v_r 是节点 j 相对于节点 i 的移动速度.

在 $\triangle abc$ 中, 由余弦定理可以得到:

$$\cos \angle acb = \frac{d_{bc}^2 + d_{ac}^2 - d_{ab}^2}{2d_{bc}d_{ac}} \quad (18)$$

由于已经假设在 Δt 时间内, 节点的相对移动速度和方向都保持不变, 所以 d_{bc} 可以表示为

$$d_{bc} = v_r \times \Delta t \quad (19)$$

根据式(17)、式(18)和式(19), 可以得到:

$$v_r = \sqrt{\frac{d_{ab}^2 - d_{ac}^2}{\Delta^2 t} - \frac{2cd_{ac}(f_c - f)}{f\Delta t}} \quad (20)$$

将式(15)、式(16)带入式(20), 可以得到“先接近再远离”场景中节点 j 相对于节点 i 的运动速度大小 v_r .

$$v_r = \sqrt{\frac{\lambda^2 P_t G_t G_r}{16\pi^2 \Delta^2 t} \left(\frac{1}{P_{ij}^b} - \frac{1}{P_{ij}^c} \right) - \frac{\lambda c (f_c - f)}{2\pi f \Delta t} \sqrt{\frac{P_t G_t G_r}{P_{ij}^c}}} \quad (21)$$

节点 j 相对于节点 i 的运动方向可由 $\angle acb$ 确定.

(2) “远离”场景

在图 1 中, 节点 j 从 c 运动到 b , 属于“远离”场景. 其计算流程与 2.4.1(1) 相同, 最终可以得到“远离”场景中节点 j 相对于节点 i 的运动速度大小 v_r .

$$v_r = \sqrt{\frac{\lambda^2 P_t G_t G_r}{16\pi^2 \Delta^2 t} \left(\frac{1}{P_{ij}^c} - \frac{1}{P_{ij}^b} \right) + \frac{\lambda c (f - f_b)}{2\pi f \Delta t} \sqrt{\frac{P_t G_t G_r}{P_{ij}^b}}} \quad (22)$$

节点 j 相对于节点 i 的运动方向可由 $\angle abc$ 确定.

当然, 在“远离”场景中也可能存在一些特殊情况, 对于图 2(a)、(b) 所示的情况, 依然可以采用 2.4.1(1) 中的方法进行计算. 其中, 在图 2(a)、(b) 中, 节点 j 若从 c 运动到 b , 则节点 j 相对于节点 i 的运动速度大小 v_r 与式(22) 相同. 在图 2(b) 中, 节点 j 若从 b 运动到 c , 则节点 j 相对于节点 i 的运动速度大小 v_r 与式(21) 相同.

在图 2(a) 中, 节点 j 若从 b 运动到 c , 则节点 j 相对于节点 i 的运动速度大小 v_r 为

$$v_r = \sqrt{\frac{\lambda^2 P_t G_t G_r}{16\pi^2 \Delta^2 t} \left(\frac{1}{P_{ij}^b} - \frac{1}{P_{ij}^c} \right)} \quad (23)$$

2.4.2 链路生存可能性计算

两个节点之间的距离在运动过程中超过通信半径时, 二者的通信链路中断. 利用两节点的相对移动速度和移动方向, 可以估计出二者通信链路生存的可能性. 链路生存可能性的估算是建立在场景划分的基础之上的, 也假设了两个节点保持当前的运动速度和方向, 因此也是不准确的. 链路生存的可能性仅仅是为了评估当前链路“存活”的几率. 链路生存可能性的值越大, 代表当前链路“存活”的几率越高.

(1) “先接近再远离”场景

图 3 中的要素与图 1 中的相似, 节点 i 在 a 处被视为静止不动, 节点 j 以 v_r 的相对移动速度从 b 处移动到 c 处, 经历的时间为 Δt . 虚线圆以 a 为圆心, 代表节点 i 的最大通信范围 R , 节点 j 超出虚线圆就代表二者之间的链路中断.

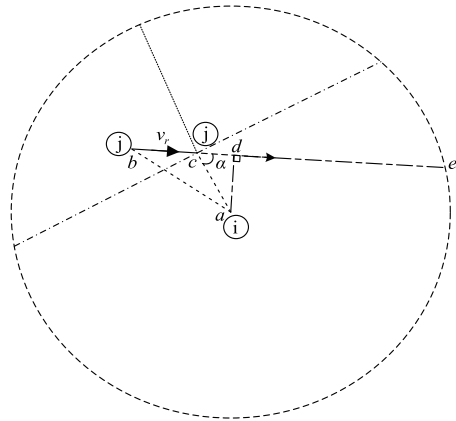


图3 “先接近再远离”场景

节点 i 、 j 之间链路的生存可能性主要与节点 j 相对于节点 i 的运动速度大小、运动方向和二者之间的距离有关, 因此, 链路生存的可能性可以表示为:

$$LSP_{ij} = \begin{cases} \frac{d_{ce}}{v_r t_{\text{threshold}}}, & \frac{d_{ce}}{v_r} < t_{\text{threshold}} \\ 1, & \frac{d_{ce}}{v_r} \geq t_{\text{threshold}} \end{cases} \quad (24)$$

式中, LSP_{ij} 代表节点 i 、 j 之间链路的生存可能性, 数值越大越好; d_{ce} 代表 ce 之间的距离; $t_{\text{threshold}}$ 代表链路生存时间的门限值, 之所以要使用门限值, 是因为当节点之间的相对移动速度接近零时 (两节点的移动速度和方向十分接近时), $\frac{d_{ce}}{v_r}$ 趋于无穷大, 所以当 $\frac{d_{ce}}{v_r} \geq t_{\text{threshold}}$ 时, 就认定其值等于 $t_{\text{threshold}}$, 因此 LSP_{ij} 就等于 1.

对于 d_{ce} 的大小, 计算过程如下.

在 2.4.1(1) 中, $\cos \angle acb$ 可以计算出, 同时其补角的余弦值 $\cos \angle acd$ 也可以求出, 因此, d_{ad} , d_{cd} 可以表示为:

$$d_{cd} = d_{ac} \cos \angle acd \quad (25)$$

$$d_{ad} = \sqrt{d_{ac}^2 - d_{ac}^2 \cos^2 \angle acd} \quad (26)$$

所以 d_{ce} 可以表示为:

$$d_{ce} = d_{ac} \cos \angle acd + \sqrt{R^2 - d_{ac}^2 + d_{ac}^2 \cos^2 \angle acd} \quad (27)$$

将式(18)、(15)、(16)代入式(27),可得:

$$d_{ce} = \frac{A}{-2v_r \Delta t} + \sqrt{R^2 - \frac{\lambda^2 P_i G_i G_r}{16\pi^2 P_{ij}^c} + \frac{A^2}{4v_r^2 \Delta^2 t}} \quad (28)$$

式中, $A = \frac{\lambda^2 P_i G_i G_r}{16\pi^2} \left(\frac{1}{P_{ij}^c} - \frac{1}{P_{ij}^b} \right) + v_r^2 \Delta^2 t$.

(2)“远离”场景

在图 4 中,节点 j 从 c 处以相对速度 v_r 运动到 b 处,历时 Δt .

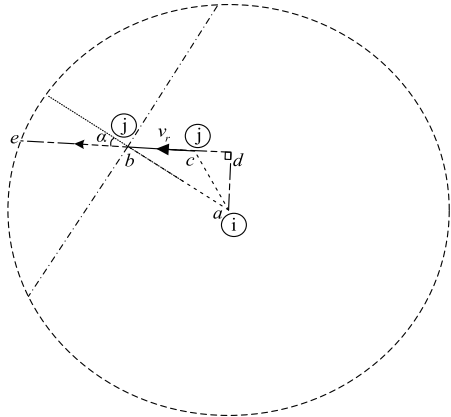


图4 “远离”场景

节点 i 、 j 之间链路的生存可能性可以表示为:

$$LSP_{ij} = \begin{cases} \frac{d_{be}}{v_r t_{\text{threshold}}}, & \frac{d_{be}}{v_r} < t_{\text{threshold}} \\ 1, & \frac{d_{be}}{v_r} \geq t_{\text{threshold}} \end{cases} \quad (29)$$

对于 d_{be} 的大小,计算过程如下.

由于 v_r 在 2.4.1(2) 中可以求得,所以 $\triangle acb$ 中三边的长度均已知,根据余弦定理可得:

$$\cos \angle acd = \frac{d_{ac}^2 + d_{bc}^2 - d_{ab}^2}{-2d_{ac} d_{bc}} \quad (30)$$

根据 $\cos \angle acd$ 即可以求得 d_{ad} 和 d_{cd} , 因此, d_{be} 可以表示为:

$$d_{be} = \sqrt{R^2 - d_{ac}^2 + d_{ac}^2 \cos^2 \angle acd} - d_{ac} \cos \angle acd - v_r \Delta t \quad (31)$$

将式(30)、(15)和式(16)带入式(31),可得

$$d_{be} = \sqrt{R^2 + \frac{A^2}{4v_r^2 \Delta^2 t} - \frac{\lambda^2 P_i G_i G_r}{16\pi^2 P_{ij}^c} + \frac{A}{2v_r \Delta t}} - v_r \Delta t \quad (32)$$

式中, $A = \frac{\lambda^2 P_i G_i G_r}{16\pi^2} \left(\frac{1}{P_{ij}^c} - \frac{1}{P_{ij}^b} \right) + v_r^2 \Delta^2 t$.

对于“远离”场景中的特殊情形,如图 2(a)(b),其链路生存可能性 LSP_{ij} 计算方法的差异主要体现在 d_{ce} 、

d_{be} 和 v_r 上. v_r 的计算已经在 2.4.1(2) 中给出. 下面具体给出 d_{ce} 、 d_{be} 的计算方法.

对于图 2(a),

$$d_{be} = \sqrt{R^2 - \frac{\lambda^2 P_i G_i G_r}{16\pi^2 P_{ij}^c} - v_r \Delta t}; c \rightarrow b \quad (33)$$

$$d_{ce} = \sqrt{R^2 - \frac{\lambda^2 P_i G_i G_r}{16\pi^2 P_{ij}^b}}; b \rightarrow c \quad (34)$$

对于图 2(b),

$$d_{be} = \sqrt{R^2 + \frac{B^2}{4v_r^2 \Delta^2 t} - \frac{\lambda^2 P_i G_i G_r}{16\pi^2 P_{ij}^c} - \frac{B}{2v_r \Delta t}}; c \rightarrow b \quad (35)$$

$$d_{ce} = \sqrt{R^2 - \frac{\lambda^2 P_i G_i G_r}{16\pi^2 P_{ij}^c} + \frac{A^2}{4v_r^2 \Delta^2 t} - \frac{A}{2v_r \Delta t}}; b \rightarrow c \quad (36)$$

式中, $B = \frac{\lambda^2 P_i G_i G_r}{16\pi^2} \left(\frac{1}{P_{ij}^c} - \frac{1}{P_{ij}^b} \right) + v_r^2 \Delta^2 t$.

3 基于生物启发的移动感知分簇算法

本文提出了一种适用于大规模、高动态移动自组织网络的分簇算法(BIMAC 算法). 该算法包括簇的建立和簇的维护两个阶段. 当自组织网络形成时,簇的建立也就开始进行. 簇的维护主要用于解决由于节点的运动而出现的问题,保持簇的稳定性.

BIMAC 算法首先会统计当前节点和其一跳邻居节点 $\frac{d(\Delta P_{ij}(t))}{dt}$ 值之和,但为了保证所建立簇的稳定性,

所以,并不是当前节点一跳范围内的所有节点都在统计范围之内. 对于 Q_{ij} 小于 Q_{\min} 的邻居节点,计算不予考虑. 在移动自组织网络中,每一个节点都会计算自身成为簇首的可能性,计算过程为:

$$CHP_i(t) = \sum_{j \in N} \frac{d}{dt} \Delta P_{ij}(t) \quad (37)$$

式中, $CHP_i(t)$ 表示当前节点 i 成为簇首的可能性,其值越大越可能成为簇首; N 为当前节点 i 的一跳范围内的邻居节点的集合,但是 N 不包含 Q_{ij} 小于 Q_{\min} 的邻居节点.

3.1 簇的建立

3.1.1 簇首的选择

在移动自组织网络刚建立时,所有的节点都处于孤立的状态,他们会周期性地广播 Hello 消息并基于其他节点发出的 Hello 消息建立其邻居列表. 簇的建立首先要进行簇首的选择. 簇首的选择过程如下:

(1) 当某一节点连续接收到两个由邻居节点发送来的 Hello 消息时,那么当前节点会计算邻居节点相对其的运动速度和运动方向,并由 2.4.2.3 计算二者之间链路的生存可能性和二者的运动稳定性.

(2) 所有的节点都基于步骤(1)的计算结果根据式

(37) 计算自身成为簇首的可能性, 并通过 Hello 消息发送给其所有的邻居节点. 当某一节点接收到其他节点发来的 $CHP_i(t)$ 时, 会与自身成为簇首的可能性作比较, 若当前节点成为簇首的可能性大于其他节点, 那么当前节点会通过 Hello 消息广播自身成为簇首的可能性; 若当前节点成为簇首的可能性小于其他节点, 那么当前节点在 Hello 消息中不附加自身成为簇首的可能性.

(3) 如果某一节点 i 长时间未收到其他节点的附加成为簇首可能性的 Hello 消息, 那么说明在其一跳范围内, 当前节点 i 成为簇首的可能性是最大的, 没有节点比当前节点 i 更适合成为簇首, 因此, 节点 i 成为簇首.

3.1.2 簇成员的选择

如果当前节点仅接收到一个节点发出的附加有成为簇首信息的 Hello 消息且该节点与簇首的可用虚拟通信量大于 Q_{min} , 那么当前节点加入该簇.

如果当前节点接收到多个节点发出的附加有成为簇首信息的 Hello 消息, 那么当前节点选择 Q_{ij} 最大节点作为其簇首, 并加入该簇. 图 5 为簇建立过程的流程图.

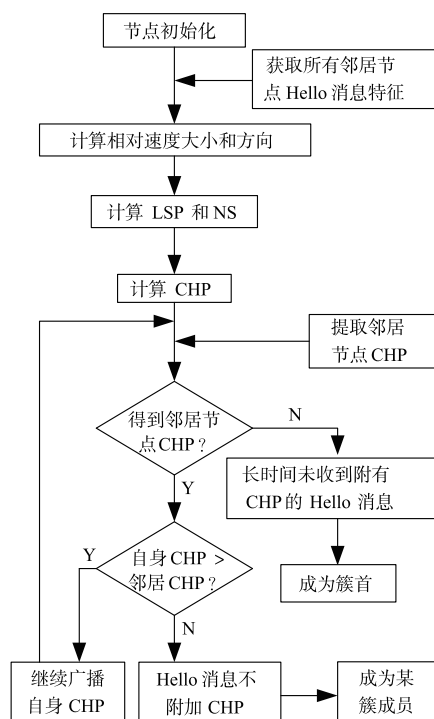


图5 簇建立过程的流程图

3.2 簇的维护

移动自组织网络的高动态性会降低簇结构的稳定性, 因此, 簇的重建或链路的重建在此网络中十分常见, 进而造成控制维护信息的大量增加. BIMAC 算法将节点的移动特性与多头绒泡菌的觅食过程相结合, 从而使该算法在簇的维护阶段能够预测链路的断路或者簇的变化, 进而减少簇的重建或链路的重建, 最终降低控

制维护信息的发送次数. 在移动自组织网络中, 簇的维护主要分为以下几个情形: 簇成员离开或加入某簇、簇的拆分、簇的合并、簇首的更新.

3.2.1 簇成员离开、加入某簇

由于节点的高速移动和其有限的通信半径, 所以某一簇成员和该簇的簇首之间的链路可能会出现断路的现象. 可以对某一簇中的所有簇成员做如下分类:

(1) 一部分簇成员不仅可以接收到本簇簇首发出的 Hello 消息, 还可以接收到其他簇簇首发出的 Hello 消息.

(2) 一部分簇成员仅能接收到本簇簇首发出的 Hello 消息.

分类(1)中的簇成员自身保存着一个特殊的邻居列表, 该列表用于存储该簇成员与其他簇簇首的可用虚拟通信量 Q_{ij} . 如果簇成员 i 与本簇簇首 m 之间的可用虚拟通信量 Q_{im} 在某一时刻大于簇成员 i 与其他簇簇首 j 的可用虚拟通信量 Q_{ij} , 那么簇首 j 将移出簇成员 i 的特殊邻居列表, 直到 $Q_{ij} > Q_{im}$. 当某一簇成员感知到自身有远离本簇簇首的趋势时, 便开始建立特殊的邻居列表, 并实时更新. 当 $Q_{im} < Q_{min}$ 时, 说明簇成员 i 与本簇簇首 m 之间的链路即将中断, 簇成员 i 会将该信息通知其簇首 m , 并且在特殊邻居列表中选择一个可以提供最大可用虚拟通信量的簇首, 并向其发送加入请求. 本文提出的方法能够在簇成员离开当前簇(与簇首之间的链路出现中断)之前预测该现象的发生, 并提前为该簇成员找到合适的簇首并建立连接. 因此, 该方法能够大大的减少簇成员离开当前簇后再次与其他簇首建立连接所需要的时间, 提高了效率.

当然, 分类(1)中的簇成员存在一种特殊的情况, 其特殊邻居列表中并没有其他簇首, 当该部分簇成员与其簇首之间的链路出现断路时, 他们将保持孤立的状态并等待其他簇首发出的 Hello 消息, 直到找到合适的簇首并加入该簇. 该方法同样适用于分类(2)中的情况.

3.2.2 簇首的更新

由于簇中节点的移动状态是时变的, 所以簇首可能随着时间的推移变得不再适合成为簇首. 在 BIMAC 算法中, 当某一簇首的 $CHP_i(t)$ 出现大幅下降时, 需要重新选择新的簇首. 当簇建立后, 簇首会根据接收到的 Hello 消息随时更新其成为簇首的可能性, 并根据所在簇的规模计算其平均值.

$$ACHP_i(t) = \frac{1}{M} \sum_{j \in N} \frac{d}{dt} \Delta P_{ij}(t) \quad (38)$$

式中, M 代表该簇中节点的数量.

当 $ACHP_i(t)$ 的值在 T 时间内的下降幅度超过 ϕ 时, 说明当前簇首 i 不再适合担任簇首的工作, 需要更换该簇的簇首, 因此, 节点 i 放弃簇首身份并广播簇首即将更换的信息. 所有的簇内成员将进入簇的初始化

环节,进行新簇首的选择.

$$\Delta ACHP_i(T) = ACHP_i(t-T) - ACHP_i(t) > \phi \quad (39)$$

式中, T 为发送 Hello 消息周期的整数倍.

3.2.3 簇的合并

在移动自组织网络中,簇的合并主要是指两个簇合并为一个簇.其主要特征为两个簇的簇首可以互相接收到对方发出的 Hello 消息,当两个簇的大部分已经重合时,信息的碰撞会十分频繁,以至于降低网络的性能,因此,进行簇的合并十分必要.

对于簇的合并,文献[21]提出了一种较为普遍的做法,在进行了相关的比较之后,某一个簇的簇首退化为另一个簇的簇成员,原本属于退化簇首的簇成员将寻找合适的簇加入.本文基于上述方法,提出了一种适用于高动态移动自组织网络的簇的合并方法.具体过程如下:

某一簇首 i 可以接收到簇首 j 发出的 Hello 消息,如果该现象仅持续了很短的时间,则不会产生簇的合并.如果簇首 i 可以长时间连续接收到簇首 j 发出的 Hello 消息,并出现如下现象时,说明这两个簇即将合并.同时簇首 i 会将簇合并的消息通知本簇所有成员和簇首 j .

(1) 簇首 i 与簇首 j 之间的可用虚拟通信量 Q_{ij} 大于 λQ_{\min} .

(2) 簇首 i 与 j 一半以上的簇成员之间的可用虚拟通信量也大于 λQ_{\min} .

对于两个即将合并的簇,首先要进行簇首的选择.两个簇的簇首 i 、 j 会根据式(37)分别计算其成为簇首的可能性, $CHP_i(t)$ 值最大的将成为合并之后新簇的簇首,需要注意的是,这里将使用特殊的 Hello 消息,只统计原来两簇中的节点.新簇中的簇成员也只选择原来两簇中的节点.上述方法可以减少因为簇的合并而对周围其他簇造成的连锁反应.

3.2.4 簇的拆分

在移动自组织网络中,簇的拆分主要是指由原簇分裂为两个更小的簇,当某个簇发生分裂时,会出现如下现象:当前簇首与一部分簇成员的可用虚拟通信量 Q_{ij} 基本保持不变,而该簇首与另一部分簇成员的可用虚拟通信量却在下降,出现可用虚拟通信量下降的该部分簇成员的数量必须不少于 αM 个节点, M 为该簇中节点的数量.

本文将可用虚拟通信量基本保持不变的簇成员与当前簇首统称为“集合 A ”,将可用虚拟通信量下降的簇成员统称为“集合 B ”.当簇首与集合 B 中一半以上的节点的可用虚拟通信量小于等于 Q_{\min} 时,就认定原簇已经不存在,集合 A 和集合 B 已经成为独立的两个簇.

对于分裂形成的两个新簇,需要重新进行簇首的选择和簇成员的选择.与 3.1.1、3.1.2 不同的是,集合

A 、 B 中的节点在计算自身成为簇首可能性的时候,只统计与之在相同集合内的节点的 $\frac{d(\Delta P_{ij}(t))}{dt}$,因此,在此处需要用到特殊的 Hello 消息.同样的,只有原簇中的节点才可以加入分裂产生的新簇.上述方法可以减少因为簇的拆分而对周围其他簇造成的连锁反应.

4 仿真结果分析

在仿真环境下,本文将 BIMAC 算法与部分主流算法进行了性能比较,这些算法包括:MOBIC 算法^[6]、DDVC 算法^[12]、MPBC 算法^[11] 和 MPCR 算法中的分簇算法^[8].其中,运用的仿真软件为 NS-2 (Network Simulator v2.34).MAC 层采用基于 CSMA/CA 的载波竞争接入协议. Hello 消息的载波频率为 600MHz.网络协议配置采用软件默认配置.调制方式为 BPSK.数据传输率为 500kbit/s.本文通过改变发射功率、接收灵敏度和天线增益来控制节点之间的有效通信半径.信道采用自由空间衰减模型,噪声功率忽略不计.在初始化阶段,节点在仿真区域内随机分布.其他仿真参数如表 1 所示.

表 1 仿真参数

参数	值
移动区域	50km * 50km
节点数量	100 个
最大通信半径	10km - 15km
移动模型	2.1 中提到的模型
最大移动速度	40m/s - 70m/s
节点放置策略	Random

图 6 给出了在节点之间的最大通信距离为 10km 的条件下,不同分簇算法平均链路连接保持时间随最大移动速度的变化比较.平均链路连接保持时间是簇首与簇成员从建立链路到链路断开所经历时间的平均值.该性能指标代表了簇首与其簇成员之间链路的平均连接时间,时间越长,代表分簇算法性能越好.随着最大移动速度的变化,BIMAC 算法与 MPCR 算法、MPBC 算法、DDVC 算法和 MOBIC 算法相比,有着最长的平均链路连接保持时间.由于 MOBIC 算法在簇首和簇成员的选择过程中没有对其一跳邻居节点进行区分对待,所以会存在一些与簇首之间链路连接保持时间并不长的节点被选为簇成员,造成了该算法的平均链路连接保持时间最小.随着最大移动速度的增加,上述五种算法的平均链路连接保持时间都有不同程度的下降,这是因为移动速度增加造成了网络拓扑变化越来越明显,因此,链路连接保持时间会普遍的下降.其中,DDVC 算法与其他四种算法相比,下降速度最快,原因为:DDVC 算法是专门为伪线性移动自组织网络设计的,适用于速度变化不频繁的移动场景,而移动自组织网络中节点的速度变化是较为频繁的.本文提出的 BIMAC 算法在簇的建立中使

用了较为严格的筛选条件,使得簇首与簇成员的稳定性更高,链路连接保持时间也最长。

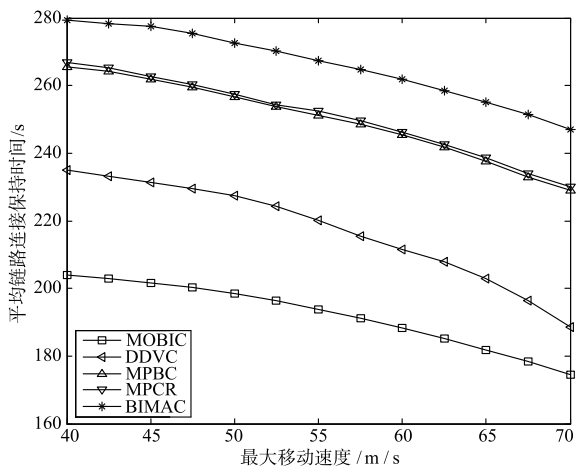


图6 平均链路连接保持时间随最大移动速度的变化图

图7给出了在最大移动速度为55m/s的条件下,不同分簇算法平均链路连接保持时间随最大通信距离的变化比较.当节点间的最大通信半径增加时,上述五种算法的平均链路连接保持时间也都有所增加,主要原因为:最大通信距离增加时,簇成员移出簇首通信范围之外的概率减小,因此簇首与簇成员的平均链路连接时间会增加.无论最大通信半径如何变化,BIMAC算法的平均链路连接保持时间始终维持在一个较高的水平。

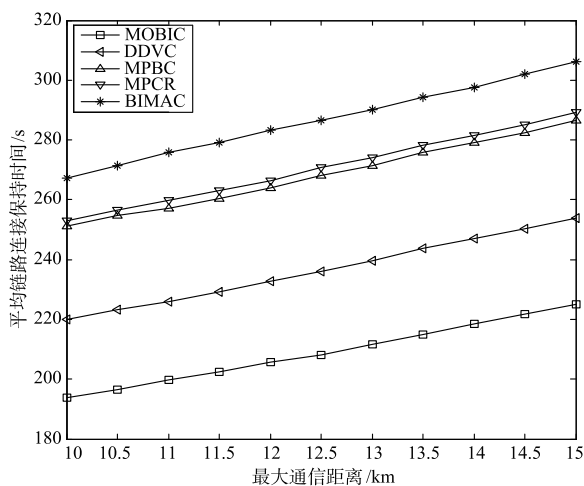


图7 平均链路连接保持时间随最大通信距离的变化图

图8表示的是最大通信半径为10km时,不同分簇算法平均簇首保持时间随最大移动速度的变化比较.平均簇首保持时间是某一节点从获得簇首身份到失去簇首身份所经历时间的平均值.随着节点最大移动速度的增加,网络拓扑结构变化的也就更为明显,簇首的更替也会变得较为频繁,因此,上述五种分簇算法的平均簇首保持时间都会有所下降. BIMAC算法将节点的移动特征与多头绒泡菌觅食的数学模型相结合,提出

了一个较为严格的簇首选择机制,该算法选出的簇首与其他算法相比有着更强的稳定性,因此,BIMAC算法中的簇首可以将其簇首身份保持较长的时间。

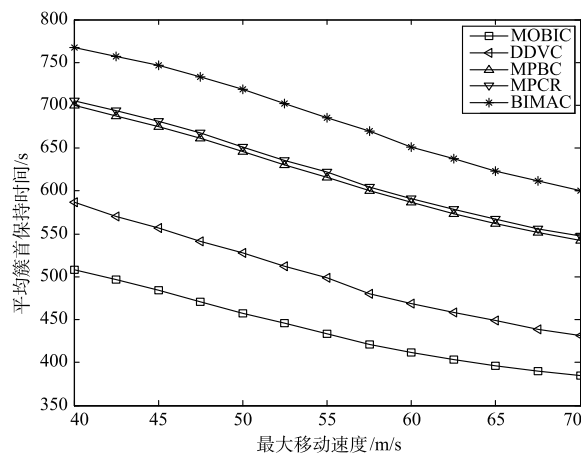


图8 平均簇首保持时间随最大移动速度的变化图

图9给出了在最大移动速度为55m/s的条件下,不同分簇算法平均簇首保持时间随最大通信距离的变化比较.随着节点间最大通信距离的增加,各个算法的平均簇首保持时间都有所增长,这是因为最大通信距离的增加会使节点的移动变得不那么明显,增加了簇首的稳定性.无论节点的最大通信距离如何变化,BIMAC算法的平均簇首保持时间与其他四种算法相比,一直维持在一个较高的水平。

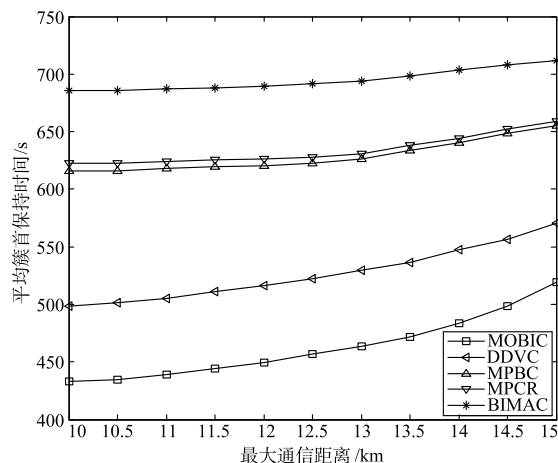


图9 平均簇首保持时间随最大通信距离的变化图

图10表示的是最大通信半径为10km时,不同分簇算法平均簇成员个数随最大移动速度的变化比较.对每一个簇内成员个数进行周期采样,然后取平均值,就可以得到平均簇成员个数.无论最大移动速度如何变化,MOBIC算法、MPBC算法和MPCR算法的平均簇成员个数几乎相等.当最大移动速度增加时,簇成员有更大的几率“逃出”簇首的通信范围,簇内节点数量会逐渐减少,因此,对于上述五种算法中的任意一种算法,

最大移动速度均与平均簇成员个数成负相关. BIMAC 算法与其他四种算法相比,其平均簇成员个数最低,原因是: BIMAC 算法对于簇成员的加入和离开都有着较为严格的规定,所以其平均簇成员个数最少. 这里所说的“严格”指的是:并不是进入簇首通信范围内的所有节点都可以作为其簇成员,需要满足一定的条件才可以; BIMAC 算法能够预测某一簇成员即将离开当前簇,为了维护当前簇的稳定,该算法会提前删除该簇成员,然而其他算法在该种情况下依然认为该簇成员属于当前簇.

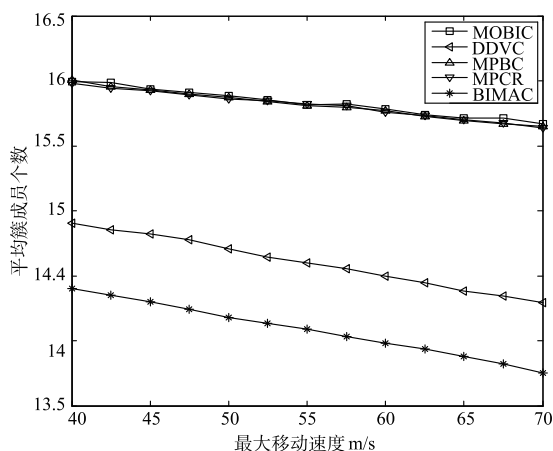


图10 平均簇成员个数随最大移动速度的变化图

图 11 给出了在最大移动速度为 55m/s 的条件下,不同分簇算法平均簇成员个数随最大通信距离的变化比较. 无论节点间的最大通信距离如何变化, MOBIC 算法、MPBC 算法和 MPCR 算法的平均簇成员个数几乎相等. 随着最大通信距离的增加,各个算法的平均簇成员数量都会有所增加,这是因为:最大通信距离增加,会导致簇首一跳范围内的邻居节点数量也增加,这些邻居节点都有可能成为该簇的簇成员,因此,簇成员数量会增加. BIMAC 算法依然保持了最低的平均簇成员个数.

图 12 表示的是最大通信半径为 10km 时,不同分

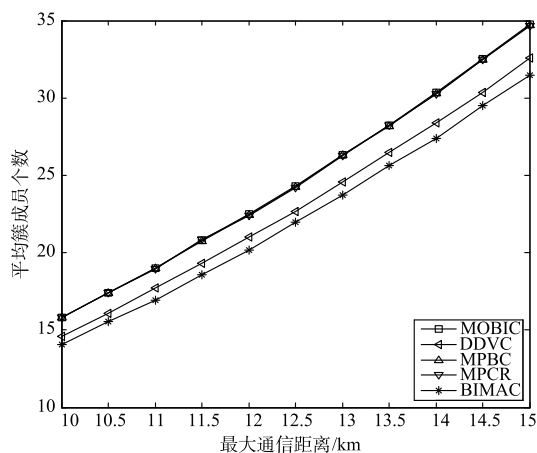


图11 平均簇成员个数随最大通信距离的变化图

簇算法平均再次连接间隔时间随最大移动速度的变化比较. 平均再次建立连接间隔时间为某一节点与当前簇首断开连接到其与其他簇首重新建立连接的间隔时间的平均值. 本文提出的算法与其他四种算法相比,保持了较低的平均再次建立连接间隔时间,这是由于本文算法具有移动预测的功能.

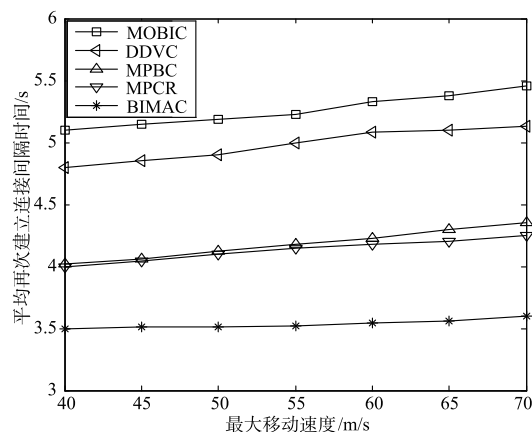


图12 平均再次建立连接间隔时间随最大移动速度的变化图

图 13 给出了在最大移动速度为 55m/s 的条件下,不同分簇算法平均再次连接间隔时间随最大通信距离的变化比较. 最大通信距离的增加增大了再次连接的可能性,因此,平均再次连接间隔时间会随着最大通信距离的增大而减少. 无论最大通信距离如何变化, BIMAC 算法均保持了较低的平均再次连接间隔时间.

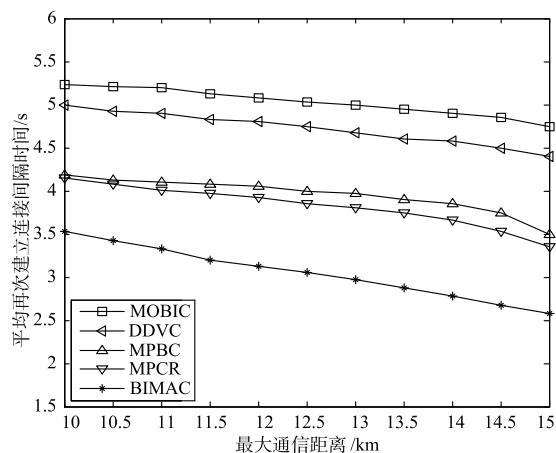


图13 平均再次建立连接间隔时间随最大通信距离的变化图

5 结束语

本文针对传统分簇算法应用于高动态的大规模移动自组织网络中性能不佳的情况,提出了一种基于生物启发的移动感知分簇算法(BIMAC 算法),该算法将多头绒泡菌觅食模型中需要调节的参数映射到移动自组织网络领域的相应参数,并与节点的移动特性相结

合. 利用该经过改进的模型, 可以对高动态的大规模移动自组网实现簇的建立与簇的维护, 而且建立的簇结构更加稳定, 从而可以减少因为簇的重新建立而产生的控制开销. 通过仿真, BIMAC 算法与 MOBIC 算法、MPBC 算法、MPCR 算法和 DDVC 算法进行了性能比较, 结果表明, BIMAC 算法可以有效地提高平均链路连接保持时间和平均簇首保持时间, 且性能优于另外四种算法, 同时也说明了 BIMAC 算法可以提高分簇结构的稳定性, 十分适用于高动态、大规模的移动自组织网络. 本文仅对移动自组织网络的分簇算法进行了研究, 下一步将在此分簇算法的基础上研究其路由算法, 适用于该分簇算法的路由算法应包含簇内路由和簇间路由, 如何在簇间进行高效的数据转发将成为下一步研究的工作重点.

参考文献

- [1] CHANG Sang-woo, LEE Sang-sun. A routing protocol for urban vehicular multi-hop data delivery[J]. Chinese Journal of Electronics, 2016, 25(2): 348 – 356.
- [2] 陶桦, 冯富琴, 等. 基于运行轨迹特征分析的车辆自组织网路由算法[J]. 通信学报, 2016, 37(6): 144 – 153.
TAO Hua, FENG Fu-qin, et al. Routing algorithm based on characteristics analysis of vehicle trace in vehicular ad hoc network[J]. Journal on Communications, 2016, 37(6): 144 – 153. (in Chinese)
- [3] 徐宁, 张沪寅, 等. 认知 Ad Hoc 网络中基于信道相似度的分簇算法研究[J]. 电子学报, 2016, 44(10): 2323 – 2329.
XU Ning, ZHANG Hu-yin, et al. Channel similarity based clustering algorithm in cognitive ad hoc network[J]. Acta Electronica Sinica, 2016, 44(10): 2323 – 2329. (in Chinese)
- [4] 张继, 张大方, 等. 一种基于演化博弈的分簇协作路由算法[J]. 电子学报, 2016, 44(9): 2158 – 2163.
ZHANG Ji, ZHANG Da-fang, et al. A cluster cooperative routing algorithm based on evolutionary game[J]. Acta Electronica Sinica, 2016, 44(9): 2158 – 2163. (in Chinese)
- [5] 李杰, 宫二玲, 等. 航空自组网中面向容错的中继节点速度控制[J]. 国防科技大学学报, 2015, 37(4): 158 – 164.
LI Jie, GONG Erling, et al. Relay speed control for realization of fault-tolerant aeronautical ad hoc networks[J]. Journal of National University of Defense Technology, 2015, 37(4): 158 – 164. (in Chinese)
- [6] BASU P, KHAN N, LITTLE T. A mobility based metric for clustering in mobile ad hoc networks[A]. Proceedings of International Conference on Distributed Computing Systems Workshop[C]. Mesa, AZ, USA: IEEE, 2001. 413 – 418.
- [7] TOLBA F, MAGONI D, LORENZ P. Connectivity, energy and mobility driven clustering algorithm for mobile ad hoc networks[A]. Proceedings of IEEE GLOBECOM[C]. Washington, DC USA: IEEE, 2007. 2786 – 2790.
- [8] SHU Jian, GE Yangfan, LIU Linian. Mobility prediction clustering routing in UAVs[A]. Proceedings of IEEE International Conference on Computer Science and Network Technology[C]. USA: IEEE, 2011. 1983 – 1987.
- [9] CAI Mengqing, RUI Lanlan, LIU Danmei. Group mobility based clustering algorithm for mobile ad hoc networks[A]. Proceedings of the 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)[C]. USA: IEEE, 2015. 340 – 343.
- [10] Al Qadami, Laila, et al. Mobility adaptive clustering algorithm for wireless sensor network with mobile nodes[A]. Proceedings of the 17th International Conference on Advanced Communication Technology (ICACT)[C]. USA: IEEE, 2015. 121 – 126.
- [11] TAN Xiaolin, XIONG Zhongyang, HE Yun. Signal attenuation-aware clustering in wireless mobile ad hoc networks[J]. Journal of Networks, 2013, 8(4): 796 – 803.
- [12] NI Minming, ZHONG Zhangdui, ZHAO Dongmei. MP-BC: A mobility prediction-based clustering scheme for ad hoc networks[J]. IEEE Transactions on Vehicular Technology, 2011, 60(9): 4549 – 4559.
- [13] Ehssan SAKHAEI, Abbas JAMALIPOUR. Stable clustering and communications in pseudolinear highly mobile ad hoc networks[J]. IEEE Transactions on Vehicular Technology, 2008, 57(6): 3769 – 3777.
- [14] TERO A, TAKAGI S, SAIGUSA T, et al. Rules for biologically inspired adaptive network design[J]. Science, 2010, 327(5964): 439 – 442.
- [15] TERO A, KOBAYASHI R, NAKAGAKI T. A mathematical model for adaptive transport network in path finding by true slime mold[J]. Journal of Theoretical Biology, 2007, 244(4): 553 – 564.
- [16] ZHANG M, XU C, GUAN J, et al. P-iRP: Physarum-inspired routing protocol for wireless sensor networks[A]. Proceedings of the VTC[C]. Las Vegas, USA: IEEE, 2013. 1 – 6.
- [17] ZHANG M, XU C, GUAN J, et al. A novel physarum-inspired routing protocol for wireless sensor networks[J]. International Journal of Distributed Sensor Networks, 2013, Article ID 483581: 1 – 12.
- [18] 张明川, 许长桥, 等. 一种面向智慧协同网络的自适应路由策略研究[J]. 电子学报, 2015, 43(7): 1249 – 1256.
ZHANG Ming-chuan, XU Chang-qiao, et al. Adaptive allocation routing scheme for smart and cooperative networks[J]. Acta Electronica Sinica, 2015, 43(7): 1249 – 1256. (in Chinese)

- [19] NG J M, ZHANG Y. A mobility model with group partitioning for wireless ad hoc networks [A]. Proceedings of IEEE International Conference on Information Technology and Applications [C]. USA: IEEE, 2005. 289 – 294.
- [20] CSISZAR V, MORI T F. A bienayme-chebyshev inequality for scale mixtures of the multivariate normal distribution [J]. Mathematical Inequalities & Applications, 2009, 12 (4): 39 – 44.
- [21] LENG S, ZHANG Y, CHEN H-H, et al. A novel k-hop compound metric based clustering scheme for ad hoc wireless networks [J]. IEEE Transactions on Wireless Communications, 2009, 8(1): 367 – 375.

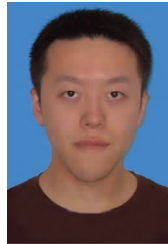
作者简介



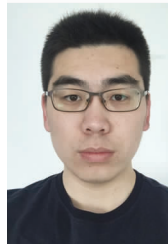
于云龙 男, 1991 年生于山东蓬莱, 现为空军工程大学防空反导学院博士研究生, 主要从事人工智能、路由算法的研究。
E-mail: 576336913@qq.com



茹乐 男, 1978 年生于陕西西安, 现为空军工程大学航空航天大学工程学院教授, 主要从事智能信息系统的研究。



方堃 男, 1990 年生于甘肃兰州, 现为空军工程大学航空航天大学工程学院硕士研究生, 主要从事路由算法的研究。



贾旭峰 男, 1992 年生于河北石家庄, 现为空军工程大学航空航天大学工程学院硕士研究生, 主要从事路由算法的研究。