

基于相对移动性预测的 k 跳AdHoc网络分簇算法

孟洛明^① 江彦馥^{*①} 刘彦君^② 苏 汉^② 徐思雅^① 齐 峰^①

^①(北京邮电大学网络与交换技术国家重点实验室 北京 100876)

^②(国网河北省电力有限公司 石家庄 050021)

摘 要: 针对网络节点随机移动造成的AdHoc网络分簇结构变化和路由失效问题, 该文提出一种基于相对移动性预测的 k 跳分簇算法, 分析和预测网络节点运动状态, 自适应地调整分簇结构, 提高簇结构稳定性。首先, 使用多普勒频移计算节点间相对移动速度, 预测节点移动性, 得到节点间链路保持时间。然后, 在簇形成阶段, 采用面向节点稳定性的MAX-MIN启发式算法, 根据节点的平均链路保持时间对簇首进行选择。进而, 在簇保持阶段, 提出一种基于节点运动状态的网络自适应调整算法, 一方面调整节点信息数据发送周期以平衡数据开销和精确度, 另一方面通过预测节点间链路通断情况调整分簇结构, 以减少链路失效时的链路重建时间, 提高网络运行质量。仿真实验表明, 所提算法可以有效延长簇首持续时间, 提高簇结构在动态环境下的稳定性。

关键词: AdHoc网络; 分簇; 移动性预测; 多普勒频移; MAX-MIN算法

中图分类号: TN929.5

文献标识码: A

文章编号: 1009-5896(2018)12-2954-08

DOI: 10.11999/JEIT180192

Relative Mobility Prediction Based k -Hop Clustering Algorithm in AdHoc Networks

MENG Luoming^① JIANG Yanfu^① LIU Yanjun^② SU Han^②

XU Siya^① QI Feng^①

^①(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

^②(State Grid Hebei Electric Power Company, Shijiazhuang 050021, China)

Abstract: To solve the problem of network structure change and route failure caused by random movement of network nodes, a relative mobility prediction based k -hop clustering algorithm is proposed, the movement of nodes are analyzed and predicted, the cluster structure is adjusted adaptively, the stability of cluster structure is improved. First, the Doppler shift is used to calculate the relative moving speed and obtain the link expiration time between nodes. Then, during the cluster formation stage, the MAX-MIN heuristic algorithm is used to select the cluster head according to the average link expiration time of the node. Furthermore, during the cluster maintenance stage, a network adaptive adjustment method is proposed based on node motion. On the one hand, the node information transmission cycle is adjusted to balance the data overhead and accuracy; On the other hand, the cluster structure is adjusted by predicting the link disconnection to reduce link reconstruction time and improve the quality of network operation. Simulation results show that the proposed algorithm can effectively prolong the duration of cluster head and improve the stability of cluster structure in dynamic environment.

Key words: AdHoc network; Clustering; Mobility prediction; Doppler shift; MAX-MIN algorithm

1 引言

随着移动通信网络异构性的逐渐增强, AdHoc

将成为其中重要的组成部分。由于AdHoc网络拓扑的动态变化性, 给通信节点间路由重建带来较大开销和时延, AODV^[1]和DSR^[2]等平面路由协议缺乏可扩展性, 而分层路由协议^[3-5]在大规模AdHoc网络中取得良好效果。建立分层结构需要将网络进行分簇, 分簇管理依赖于簇首, 簇首负责协调和管理簇成员节点。频繁的簇首改变将产生过度的控制开

收稿日期: 2018-02-11; 改回日期: 2018-08-13; 网络出版: 2018-08-21

*通信作者: 江彦馥 paperJYF@163.com

基金项目: 国家自然科学基金(61702048)

Foundation Item: The National Natural Science Foundation of China (61702048)

销^[6], 并且簇首的选择将影响网络收敛时间^[7], 因此研究快速高效并能维持拓扑相对稳定的分簇算法十分重要。

典型单跳分簇算法包括最小节点ID^[8,9]和最高连通度分簇算法^[10,11]。最小ID分簇算法选择任一节点的邻居节点中ID值最小的节点作为簇首, 即使该簇首在其自身邻居节点中并不具有最小ID值, 因此将产生较多簇首; 基于最高连通度的分簇方法选择连通度最大的节点作为簇首, 将导致簇首频繁切换, 带来较大簇维护开销。文献^[12]将分簇方法扩展到 k 跳, k 跳分簇是指节点距离簇首的最大距离为 k 跳, k 跳分簇相比单跳分簇将产生较少的簇首, 为大规模AdHoc网络提供了良好的可扩展性。

文献^[13]提出了一种快速收敛的MAX-MIN启发式 k 跳分簇方法, MAX-MIN能够在 $2k$ 轮数据包交换之后选择出簇首, 但是簇形成过程中节点信息从网关向簇首汇聚, 容易丢失非网关节点的边缘节点信息, 另外MAX-MIN的簇首选择度量为节点ID, 当网络拓扑动态性较高时, 会导致频繁的簇重构。文献^[14]在MAX-MIN基础上, 针对节点移动可能造成的簇重构, 将节点间链路过期时间和连通度混合度量作为簇首选择标准, 但计算链路过期时间时假设节点保持匀速直线运动, 与实际运动情况相比存在较大偏差。文献^[15]提出伪线性移动场景下基于多普勒值的分簇方法, 假设节点在相对线性的路径上移动, 方向和运动参数不频繁变化, 多普勒值作为节点的稳定性度量以提高簇结构稳定性。以上研究为动态网络环境下基于节点移动性预测的分簇提供了可能。

基于上述分析, 本文在文献^[13]提出的MAX-MIN启发式 k 跳分簇方法基础上, 提出了一种 k 跳移动性预测分簇算法(k -hop Mobility Prediction Clustering, KMPC)。首先建立基于多普勒频移的节点相对移动性预测模型, 计算节点间相对移动速度, 预测节点移动性, 得到节点间链路保持时间。然后, 在簇形成阶段, 采用面向节点稳定性的MAX-MIN启发式算法, 根据节点的平均链路保持时间对簇首进行选择, 同时提出基于距离的汇聚传输策略, 保证簇首收集到全部簇成员信息。进而, 在簇保持阶段, 提出一种基于节点运动状态的网络自适应调整算法, 一方面调整节点信息数据发送周期以平衡数据开销和精确度, 另一方面通过预测节点间链路通断情况调整分簇结构, 以减少链路失效时的链路重建时间, 提高网络运行质量。

2 基于多普勒频移的相对移动性预测模型

k 跳AdHoc网络中, 所有簇成员距离簇首的最

大距离为 k 跳。使用KMPC方法, 移动AdHoc网络将被分成多个非重叠簇。如图1所示为2跳分簇网络, 网络中节点有3种状态: 簇首, 网关, 普通节点。位于簇边缘并且连接到相邻簇的节点, 称为网关节点。

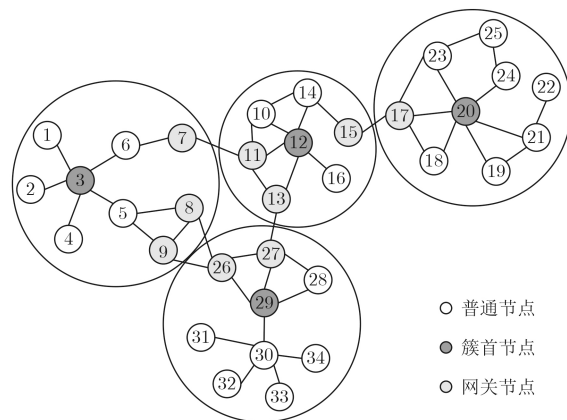


图1 2跳AdHoc网络分簇结构

假设网络节点的移动服从随机游走模型(Random Walk Mobility, RWM), 节点以时间间隔 T 为周期变化各自运动状态, 每个节点运动速率和方向的改变分别服从区间 $[v_{\min}, v_{\max}]$ 与 $[0, 2\pi]$ 上的均匀分布。本文提出的KMPC方法, 将平均链路保持时间作为簇首选择标准。首先根据节点间交互数据包的多普勒频移, 估计节点间的相对移动速率, 进而预测节点间链路保持时间。最后将节点与其邻居节点的链路保持时间的平均值, 作为节点的移动性度量。

2.1 相对移动速率估计

如图2所示, 节点 x 相对静止地停留于位置 O , 在 t 时刻, 节点 y 位于位置 A , 经过 Δt 时间间隔后, 节点 y 以相对速率 v 从位置 A 移动到 B 。节点 x 以频率 f_o 与功率 P_o 周期性广播数据包, t 时刻节点 y 在位置 A 接收到 x 发送的一个数据包, 接收功率为 P_A , 由于相对移动产生的多普勒频移为 f_A ; 经过时间间隔 Δt 之后, y 移动到位置 B , 再次接收到

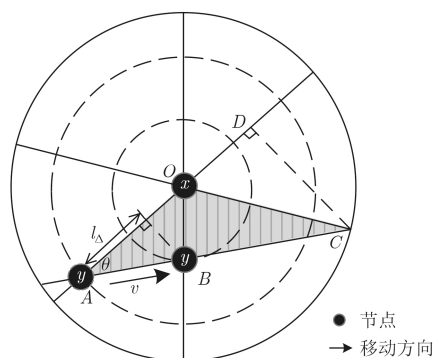


图2 节点相对位置示意图

x 发送过来的另一个数据包,接收功率为 P_B ,多普勒频移为 f_B 。节点 y 相对于 x 的相对运动趋势可由式(1)进行判断。

$$f_{\Delta} = f_B - f_A = \begin{cases} > 0, & \text{相互靠近} \\ < 0, & \text{相互远离} \\ = 0, & \text{相对静止} \end{cases} \quad (1)$$

根据多普勒频移的定义可知, $f_A = f_O \cdot (v/c) \cdot \cos \theta$, 其中 c 为光速, θ 为 OA 与 AB 间夹角, 因此 v 可以表示为

$$v = \frac{f_A \cdot c}{f_O \cdot \cos \theta} \quad (2)$$

由余弦定理可知:

$$\cos \theta = \frac{l_{OA}^2 + l_{AB}^2 - l_{OB}^2}{2l_{OA}l_{AB}} \quad (3)$$

使用式(4)所示信道损耗模型描述信号传输过程中的功率损耗:

$$P_{rx} = P_{tx} G \left(\frac{l_{ref}}{l} \right)^{\alpha} \quad (4)$$

其中, P_{rx} 是接收信号的平均功率, P_{tx} 为信号发射功率, G 是依赖于天线特性和信道平均损耗的常数, l_{ref} 是天线远场参考距离, l 为收发端间距, α 是路径损耗系数, 取 $\alpha = 4$ 。位置 O 到 A 与 B 的距离分别为 l_{OA} 与 l_{OB} , 当 x 与 y 相互靠近时, $l_{\Delta} = l_{OA} - l_{OB}$, y 在位置 A 与 B 的信号接收功率差 P_{Δ} 可表示为

$$\begin{aligned} P_{\Delta} &= P_B - P_A = \frac{P_O \cdot G \cdot l_{ref}^4}{l_{OB}^4} - \frac{P_O \cdot G \cdot l_{ref}^4}{l_{OA}^4} \\ &= P_O \cdot G \cdot l_{ref}^4 \left(\frac{1}{(l_{OA} - l_{\Delta})^4} - \frac{1}{l_{OA}^4} \right) \end{aligned} \quad (5)$$

令 $G' = P_O \cdot G \cdot l_{ref}^4$, 利用1阶泰勒级数将式(5)右侧展开, 可得

$$P_{\Delta} = G' \left(\frac{1}{l_{OA}^4} + \frac{4l_{\Delta}}{l_{OA}^5} + o(l_{\Delta}) - \frac{1}{l_{OA}^4} \right) \approx G' \frac{4l_{\Delta}}{l_{OA}^5} \quad (6)$$

由式(4)可知 $P_A = P_O \cdot G \left(\frac{l_{ref}}{l_{OA}} \right)^4$, 同时 $G' = P_O \cdot G \cdot l_{ref}^4$, 推出 $l_{OA} = \sqrt[4]{G'/P_A}$, 代入式(6)可得

$$P_{\Delta} \approx 4l_{\Delta} \sqrt[4]{P_A^5/G'} \quad (7)$$

将式(7)进行恒等变换可得

$$G' \approx P_A^5 (4l_{\Delta}/P_{\Delta})^4 \quad (8)$$

根据式(8), l_{OA} 与 l_{OB} 可以被表示为

$$l_{OA} = \sqrt[4]{G'/P_A} = 4P_A(l_{\Delta}/P_{\Delta}) \quad (9)$$

$$l_{OB} = \sqrt[4]{G'/P_B} = 4 \frac{l_{\Delta}}{P_{\Delta}} \sqrt[4]{\frac{P_A^5}{P_B}} \quad (10)$$

当时间间隔 Δt 非常小时, θ 也将变得非常小, 此时 l_{Δ} 可被近似地表示为

$$l_{\Delta} = l_{OA} - l_{OB} \approx l_{AB} \cos \theta \quad (11)$$

l_{AB} 可以被表示为

$$l_{AB} \approx \frac{l_{\Delta}}{\cos \theta} \quad (12)$$

将式(9), 式(10), 式(12)代入式(3)可得

$$\frac{1}{\cos \theta} = 2 \sqrt{2 \frac{P_A}{P_{\Delta}} - 4 \frac{P_A^2}{P_{\Delta}^2} + 4 \frac{1}{P_{\Delta}^2} \sqrt{\frac{P_A^5}{P_B}}} \quad (13)$$

联立式(2)和式(13), 可得节点 y 与 x 的相对速率 v 为

$$v = \frac{2f_A \cdot c}{f_O} \sqrt{2 \frac{P_A}{P_{\Delta}} - 4 \frac{P_A^2}{P_{\Delta}^2} + 4 \frac{1}{P_{\Delta}^2} \sqrt{\frac{P_A^5}{P_B}}} \quad (14)$$

从式(14)可以看出, 只要节点 y 成功接收到两个从 x 发送过来的数据包, 就可以独立完成一次对当前时刻相对速率 v 的估计。

2.2 链路保持时间预测

完成对节点相对运动速率的估计之后, 将预测两节点之间链路仍能有效保持的持续时间。如图2所示, 假设节点 x 的通信覆盖范围为以自身为圆心的圆形区域, 节点 y 在 $t + \Delta t$ 时刻位于位置 B , 在相对运动状态不发生改变的前提下, 节点 x 与 y 之间的无线链路, 将会在 y 运动到位置 C 之外后失效。所以节点 x 与 y 之间链路保持时间 LET_{xy} 为

$$LET_{xy} = l_{BC}/v \quad (15)$$

其中, l_{BC} 可通过式(16)计算:

$$l_{BC} = l_{AC} - l_{AB} = \frac{l_{OC} \sin \angle AOC}{\sin \theta} - v \cdot \Delta t \quad (16)$$

某一节点与其所有邻居节点的链路保持时间的平均值, 用来衡量节点与其邻居节点之间链路的稳定性。节点 x 与其邻居节点的平均链路保持时间 LET_x 可表示为

$$LET_x = \frac{1}{d_x} \sum_{y \in N(x)} LET_{xy} \quad (17)$$

其中, $N(x)$ 为节点 x 的邻居节点集合, d_x 为 x 的邻居节点个数。具有较长平均链路保持时间的节点, 能够与邻居节点保持较长时间的连接, 具有较高稳定性, 更适合被选为簇首。

3 KMPC 算法

3.1 分簇形成阶段

假设在整个网络最初启动时，全网节点都处于孤立节点状态并进入分簇形成阶段。此时，每个节点根据收到的数据包中相关信息建立邻居节点列表，计算相对运动速率和平均链路保持时间。在进行簇首选择之前，首先，每个节点将平均链路保持时间广播到邻居节点中，然后，每个节点维护一个 WINNER 数据结构，节点将 WINNER 初始值设置为其平均链路保持时间。接下来进行 FloodMax 阶段。

FloodMax: 每个节点将自己的 WINNER 值向 1 跳邻居节点进行广播，当节点收到所有 1 跳邻居节点的 WINNER 值之后，比较所有接收到的 WINNER 值和自身 WINNER 值，记录下其中的最大值，并将自己的 WINNER 值更新为该最大值。这样就完成了 1 轮 FloodMax。FloodMax 共持续 k 轮。接下来进行 FloodMin 阶段。

FloodMin: 与 FloodMax 过程类似，每个节点向 1 跳邻居节点广播 WINNER 值，当节点接收到所有 1 跳邻居节点的 WINNER 值之后，记录下其中的最小值，并将自己的 WINNER 值更新为该最小值。FloodMin 过程持续 k 轮。

接下来依据以下 4 条规则进行簇首选择：

规则 1 每个节点比较 $2k$ 轮更新后的 WINNER 值与自身初始 WINNER 值，如果相等，则该节点成为簇首，并向其 k 跳邻居节点发送簇首宣告报

文。当节点不满足规则 1 时，进行规则 2 匹配。

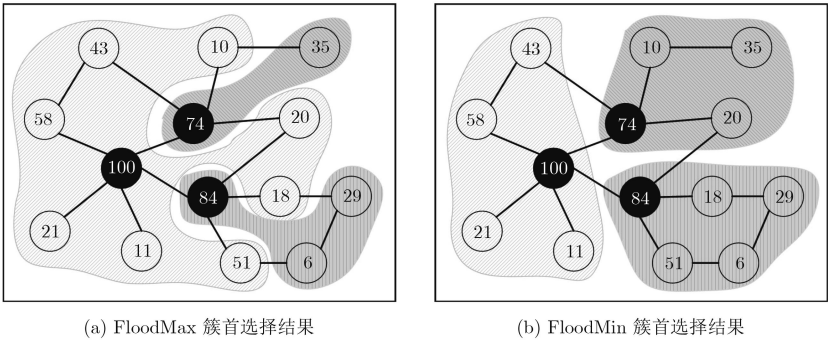
规则 2 每个节点检查 k 轮 FloodMax 过程中记录的所有 WINNER 值，与 k 轮 FloodMin 过程中的所有 WINNER 值，如果存在某一值同时出现在 FloodMax 和 FloodMin 过程中，则记录下该值。在所有被记录下的值中，选择其中的最小值，具有该最小平均链路保持时间的节点被选择为簇首。如果未找到任何一个值能够同时出现在 k 轮 FloodMax 和 k 轮 FloodMin 中，则进行规则 3 匹配。

规则 3 节点选择 FloodMax 过程中链路保持时间最大的节点作为簇首。

规则 4 执行完前 3 条规则之后，节点将选择出某一节点作为其簇首。如果该节点到所选簇首的路径上，需要经过其他已经被选择为簇首的节点，则该节点将更改簇首选择，选择其进行汇聚传输路径上所经过的第 1 个簇首。

图 3 给出了 2 跳分簇结构的簇首选择过程，每个节点经过 2 轮 FloodMax 和 2 轮 FloodMin 之后选择出了各自的簇首。

KMPC 中基于 MAX-MIN 的簇首选择方法，将每个节点发送的报文数量控制在最大跳数 k 倍数量级，而不是网络中节点总数 N 的数量级，这保证了算法具有较低的复杂度。另外， k 作为算法的重要输入变量，能够控制计算出的簇首数量和密度。由于 FloodMax 是一种贪心算法，可能导致簇首负载不均，因此设置规则 2，选取具有较小平均链路保持时间的节点作为簇首，以保证负载均衡。



节点平均链路保持时间 (s)		58	43	10	35	74	20	100	21	11	84	18	29	51	6
Flood- Max	WINNER1	100	74	74	35	100	84	100	100	100	100	84	29	84	51
	WINNER2	100	100	100	74	100	100	100	100	100	100	100	84	100	84
Flood- Min	WINNER3	100	100	74	74	100	100	100	100	100	100	84	84	84	84
	WINNER4	100	100	74	74	74	100	100	100	100	84	84	84	84	84
簇首选择		100	100	74	74	74	100	100	100	100	84	84	84	84	84

(c) 簇首最终选择结果

图 3 KMPC 簇首选择过程

每个节点经过 $2k$ 轮迭代之后确定了各自的簇首。接下来,簇首需要通过汇聚传输获得簇成员信息。KMPC采用基于距离的汇聚传输策略,从所有边缘节点开始,启动异步汇聚传输过程。首先节点判断自身是否为簇内1跳邻居节点中,距离簇首最远的节点。如果是,则该节点成为汇聚传输起始节点。然后,汇聚传输从所有起始节点开始,起始节点将自身节点信息打包,逐跳传输到簇首。在起始节点到簇首的路径上的节点,也将自身信息发送给簇首。当簇首收到簇中所有节点的数据包之后,簇首获得了所维护簇的全局信息,汇聚传输过程结束。

3.2 分簇保持阶段

簇形成以后,簇首通过定期广播控制报文,并接收簇内成员节点发送的响应报文从而维护簇成员信息。在使用多普勒频移进行链路保持时间预测时,假设在两次连续的控制报文的时间间隔 τ_s 内,节点保持匀速运动,这一假设成立的条件是 τ_s 足够小。但是 τ_s 很小时,周期性发送控制报文将造成大量控制开销。为了在控制开销和速度精确度之间取得折中,本文提出一种自适应调整算法,根据这一算法动态调整控制报文发送周期,对网络拓扑进行调整。

3.2.1 相关定义

定义 1 链路保持时间变化率:节点 n_1 和 n_2 之间的链路保持时间在 t_1 和 t_2 时刻分别是 LET_1 和 LET_2 ,则链路保持时间变化率 a 为

$$a = \frac{LET_2 - LET_1}{t_2 - t_1} = \frac{\Delta LET}{\Delta t} \quad (18)$$

定义 2 剩余保持时间占比:假设两节点能够保持有效通信的最大距离为 r ,当前时刻 t ,节点相对运动速率为 v_t ,节点间剩余链路保持时间为 LET_t ,剩余链路保持时间在总估计保持时间中所占比例,即剩余保持时间占比为

$$\xi_t = \frac{LET_t}{r/v_t} = \frac{LET_t \cdot v_t}{r} \quad (19)$$

定义 3 链路保持时间阈值:预先设置一个较小的时间长度 δ ,当两节点间链路保持时间小于 δ 时,假设该条链路失效。 δ 值即为链路保持时间阈值。

定义 4 簇成员丢失率:设簇成员节点到达簇首的跳数为 κ 。簇首更换之后,簇内部分节点到达新簇首的跳数 $\kappa > k$,假设这部分节点的个数为 n_κ ,簇内节点总数为 n ,则簇成员丢失率 ls_n 为

$$ls_n = n_\kappa / n \quad (20)$$

定义 5 指数增大/减小:每经过一个传输轮次,节点间发送数据包的时间间隔加倍/减小为当

前时间间隔的一半,即时间间隔呈指数形式增长/减小。

定义 6 加法增大:每经过一个传输轮次,节点间发送数据包的时间间隔增加 x s,即时间间隔按线性规律缓慢增长。

3.2.2 基于节点运动状态的网络自适应调整算法

在前述定义基础上,自适应调整算法在两次连续的重新成簇过程执行的时间间隔内进行,分别对包含节点信息数据的数据包发送周期和分簇拓扑进行调整。

(1) 自适应数据包发送周期调整算法

步骤 1 簇内节点根据交互数据包的多普勒频移,判断与邻居节点的相对运动趋势。当两节点相互趋近或保持相对静止时,更新当前两节点间的数据包发送时间间隔为上一次时间间隔的2倍,即时间间隔进行指数增大。当两节点相互远离时,执行步骤2;

步骤 2 在当前时刻 t ,计算两节点间剩余链路保持时间 LET_t ,剩余保持时间占比 ξ_t ,及链路保持时间的变化率 a_t 。当 $0.3 \leq \xi_t \leq 1.0$ (区间大小可根据实际情况进行设定)时,执行步骤3,否则,执行步骤4;

步骤 3 预先设置阈值 ε ,当 $|a_t| < \varepsilon$ 时,说明链路保持时间变化率较小,此时数据包发送时间间隔进行加法增大;当 $|a_t| < \varepsilon$ 时,链路保持时间变动较大,此时应当缩小数据包发送时间间隔,提高数据采集精确度,因此时间间隔进行指数减小;

步骤 4 当 $\xi_t < 0.3$ 时,数据包发送时间间隔保持某一较小值不变,直到时刻 t' ,若满足 $LET_{t'} < \delta$ 且 $|a_{t'}| \geq \varepsilon$,假设当前链路即将失效,提前启动拓扑调整过程。

(2) 自适应分簇拓扑调整算法

步骤 1 当前两节点间链路即将失效时,节点在邻居节点信息表中将两节点间的跳数设置为无穷大,簇中节点更新到达邻居节点的跳数。同时,判断即将失效的链路是否是簇首与普通节点之间的链路。如果是,执行步骤2,否则执行步骤3;

步骤 2 当簇首与普通节点之间链路断开,计算此时簇成员丢失率,如果大于某一阈值 ls_ξ ,则选举当前簇中链路保持时间最大的节点作为临时簇首。对于到达临时簇首距离大于 k 跳的节点,执行步骤4;

步骤 3 当两普通节点间链路断开,判断断开后两节点距离簇首的跳数 κ_1 和 κ_2 ,如果满足 $\kappa_1 \leq k$ 且 $\kappa_2 \leq k$,则保持当前拓扑,否则,对于距离簇首跳数大于 k 的节点执行步骤4;

步骤4 当节点到达当前簇首的跳数大于 k 时, 节点在维护的邻居节点信息表中寻找跳数小于 k 的节点集合, 在该集合中搜索是否存在簇首节点, 如果存在, 节点将加入该簇首所在的簇; 如果不存在, 节点成为孤立节点, 等待下一次重新成簇过程的启动。

在自适应调整算法中, 当节点移动缓慢时, 延长数据包发送周期, 降低发送频率, 减少控制开销; 当节点移动速度较快时, 又能缩短发送周期, 提高数据的精确度。当节点间链路有可能失效时, 提前发起簇首更换, 降低链路重建时间, 减少对数据传输的影响, 提高网络运行质量。

4 仿真

4.1 仿真参数设置和评测指标

本节通过仿真实验来比较KMPC与MAX-MIN^[13], KCMM^[16]和K-CONID^[12]3个分簇算法的性能, 采用NS2作为仿真实验平台, 仿真实验中, 节点随机分布在 200×200 个单元的区域, 节点总数为 $100 \sim 400$, 节点通信范围为 $20 \sim 30$ 个单元, 簇内普通节点到达簇首的最大跳数为 $2 \sim 4$ 跳, 节点运动速度不大于 $1/2$ 倍的通信范围, 每组仿真实验持续 2000 s, 每隔 4 s进行一次数据采样。

从簇首持续时间, 簇维护开销, 簇首切换时间和节点运动速度的精确度4方面对算法性能进行分析。其中簇首持续时间指节点成为簇首的时间长度, 簇首持续时间越长, 簇结构越稳定; 簇维护开销主要包括定期的簇状态广播报文和由簇首改变引起的控制开销, 用每秒产生的报文数量表示; 簇首切换时间指节点失去与其簇首的连接并重新加入新的簇首的平均时间; 衡量速度精确度时, 对采用自适应调整数据包发送周期和固定周期两种方式所刻画出的节点运动速度变化轨迹进行对比。

4.2 仿真结果与分析

(1)簇首持续时间: 图4显示了节点总数不同时4种算法的簇首持续时间, 簇首持续时间随节点总

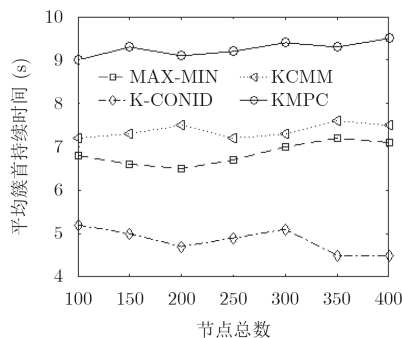


图4 簇首持续时间随节点总数的变化($k=2$)

数的变化较平缓, 其中KMPC的簇首持续时间最长, 与MAX-MIN, KCMM和K-CONID相比分别延长了36%, 26%和93%, 这是因为KMPC将平均链路保持时间较长的节点选为簇首, 而MAX-MIN和KCMM将节点ID作为簇首选择标准, K-CONID选择连通度最大的节点作为簇首, 当簇首移动性较高时, 后3种算法会导致簇首与成员节点间频繁断开, 缩短簇首持续时间。图5显示了节点最大移动速度对簇首持续时间的的影响。当节点移动速度较大时, 网络拓扑变化迅速, 簇首持续时间随之缩短。当节点移动速度相同时, KMPC具有最长的簇首持续时间。簇首持续时间越长, 簇结构越稳定, 因此KMPC能够有效提升动态环境下簇结构的稳定性。

(2)簇维护开销: 如图6所示, 对于相同的最大跳数 k , KMPC与MAX-MIN, KCMM相比, 簇维护开销分别减少了88%和80%。在簇维护阶段, KMPC根据节点运动情况自适应调整报文发送周期, 当节点移动较缓时, 延长报文发送周期, 降低发送频率, 减少控制开销。同时随跳数 k 的增长, 簇维护开销增长较缓, 表明KMPC具有良好的可扩展性。

(3)簇首切换时间: 如图7所示, 簇首切换时间随节点最大移动速度的增大而延长。当移动速度增大时, 节点有可能遭受更多通信中断, 簇首切换时间随之延长。KMPC与KCMM, MAX-MIN和K-CONID相比, 簇首切换时间分别减少了69%, 74%

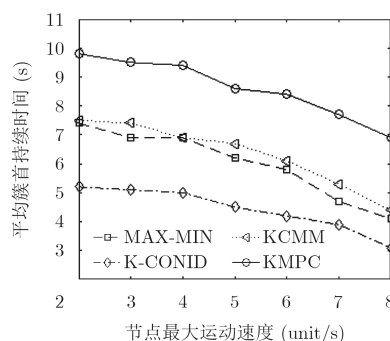


图5 簇首持续时间随最大运动速度的变化($k=2$)

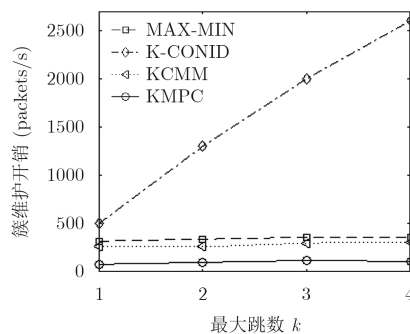


图6 簇维护开销

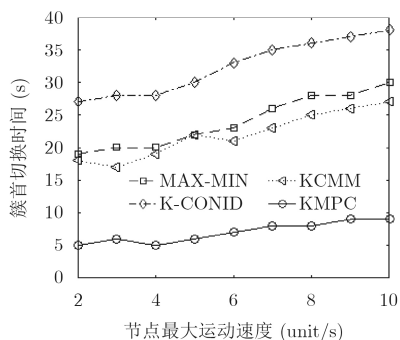


图7 簇首切换时间

和81%。这是因为MAX-MIN, KCMM和KCONID在簇首已经失效后, 需要等待一段时间进行失效确认再进行簇首更换, 而KMPC对节点间链路失效情况进行预测, 提前发起簇首更换, 因此有效降低了链路重建时间。

(4)运动速度的精确度: 图8对采用自适应调整数据包发送周期和固定周期两种方式所刻画出的节点运动速度变化轨迹进行了对比。数据包发送周期初始化为 $\tau_s = 4$ s, 在此基础上KMPC根据节点运动情况动态调整发送周期, 当节点运动速度较快时, 缩短周期, 例如在8~14 s内, 节点速度变化剧烈, KMPC将周期缩短到1 s, 获取到了这一时段速度的变化轨迹, 而采用固定周期则未能获取到这一时段的速度变化情况。因此KMPC具有更高的速度获取精确度, 能够更好适应移动性较高的网络环境。

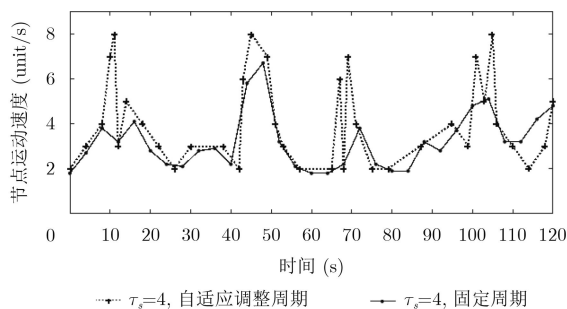


图8 节点运动速度随时间的变化

5 结束语

针对高移动性AdHoc网络分簇问题, 提出基于相对移动性预测的 k 跳分簇算法。在簇形成阶段, 采用面向节点稳定性的MAX-MIN算法, 根据节点平均链路保持时间对簇首进行选择。在簇保持阶段, 根据节点运动状态自适应调整控制报文发送周期, 以平衡数据开销和精确度, 并通过预测节点间链路通断情况, 调整分簇结构。仿真表明, 本文算

法能够延长动态环境下的簇首持续时间, 具有较小的控制开销, 可适用于移动性较强的AdHoc网络。

参考文献

- [1] KASHYAP V K, ASTYA R, NAND P, *et al.* Comparative study of AODV and DSR routing protocols in wireless sensor network using NS-2 simulator[C]. International Conference on Computing, Communication and Automation, Greater Noida, India, 2017: 687–690.
- [2] AHN S and KIM C K. An adaptive mechanism to resolve the DSR cached route reply storm of multi-hop wireless networks[C]. International Conference on Ubiquitous and Future Networks, Milan, Italy, 2017: 1022–1024.
- [3] 张军强, 王汝传, 黄海平. 基于分簇的无线多媒体传感器网络数据聚合方案研究[J]. 电子与信息学报, 2014, 36(1): 8–14. doi: [10.3724/SP.J.1146.2012.00933](https://doi.org/10.3724/SP.J.1146.2012.00933).
ZHANG Junqiang, WANG Ruchuan, and HUANG Haiping. Research on cluster-based data aggregation for wireless multimedia sensor networks[J]. *Journal of Electronics & Information Technology*, 2014, 36(1): 8–14. doi: [10.3724/SP.J.1146.2012.00933](https://doi.org/10.3724/SP.J.1146.2012.00933).
- [4] CHAI Yuan and SHI Wenxiao. Access-enhanced hybrid routing protocol for hybrid wireless mesh network[C]. International Conference on Communication Software and Networks, Guangzhou, China, 2017: 138–141.
- [5] 牛晓光, 崔莉, 黄长城. 移动自组织网络中基于优化分簇的混合路由协议[J]. 通信学报, 2010, 31(10): 58–67. doi: [10.3969/j.issn.1000-436X.2010.10.007](https://doi.org/10.3969/j.issn.1000-436X.2010.10.007).
NIU Xiaoguang, CUI Li, and HUANG Changcheng. Hybrid optimal cluster routing protocol for mobile ad hoc network[J]. *Journal of Communication*, 2010, 31(10): 58–67. doi: [10.3969/j.issn.1000-436X.2010.10.007](https://doi.org/10.3969/j.issn.1000-436X.2010.10.007).
- [6] ABUASHOUR A and KADOCH M. Performance improvement of cluster-based routing protocol in VANET[J]. *IEEE Access*, 2017, 99(5): 15354–15371. doi: [10.1109/ACCESS.2017.2733380](https://doi.org/10.1109/ACCESS.2017.2733380).
- [7] 张浩军, 朱艳琴, 纪其进. 面向异构网络的动态负载均衡算法及其收敛性分析[J]. 电子与信息学报, 2013, 35(9): 2247–2253. doi: [10.3724/SP.J.1146.2012.01360](https://doi.org/10.3724/SP.J.1146.2012.01360).
ZHANG Haojun, ZHU Yanqin, and JI Qijin. A diffusion-based dynamic load balancing algorithm for heterogeneous networks and its convergence analysis[J]. *Journal of Electronics & Information Technology*, 2013, 35(9): 2247–2253. doi: [10.3724/SP.J.1146.2012.01360](https://doi.org/10.3724/SP.J.1146.2012.01360).
- [8] NARENDHRAN M and PRAKASAM P. Optimized lowest ID in wireless sensor network using Invasive Weed Optimization (IWO)-Genetic Algorithm (GA)[C]. International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies, Chennai, India, 2017: 1–6.

- [9] NGUYEN V D, KIM O T T, DANG D N M, *et al.* Application of the lowest-ID algorithm in cluster-based TDMA system for VANETs[C]. International Conference on Information Networking, Cambodia, Cambodia, 2015: 25–30.
- [10] ZHANG Jinbei, FU Luoyi, WANG Qi, *et al.* Connectivity analysis in wireless networks with correlated mobility and cluster scalability[J]. *IEEE/ACM Transactions on Networking*, 2017, 25(4): 2375–2390. doi: [10.1109/TNET.2017.2692774](https://doi.org/10.1109/TNET.2017.2692774).
- [11] HAN S, LEE H, KIM J, *et al.* On the connectivity in opportunistic D2D networks with hierarchical and non-hierarchical clustering[C]. IEEE Global Communications Conference Workshops, Washington DC, USA, 2016: 1–6.
- [12] CHEN Geng, NOCETTI F G, GONZALEZ J S, *et al.* Connectivity based k -hop clustering in wireless networks[C]. International Conference on System Sciences, Hawaii, USA, 2002: 2450–2459.
- [13] AMIS A D, PRAKASH R, VUONG T, *et al.* Max min D-cluster formation in wireless ad hoc networks[C]. IEEE International Conference on Computer Communications, Tel Aviv, Israel, 2000: 32–41.
- [14] LENG Supeng, ZHANG Yan, CHEN Xiaohua, *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. doi: [10.1109/TWC.2009.080186](https://doi.org/10.1109/TWC.2009.080186).
- [15] SAKHAE E and JAMALIPOUR A. A new stable clustering scheme for pseudo-linear highly mobile ad hoc networks[C]. IEEE Global Communications Conference, Washington DC, USA, 2007: 1169–1173.
- [16] MA Xufeng. A k -hop cluster maintaining mechanism for mobile ad hoc networks[C]. International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, China, 2011: 1–4.

孟洛明：男，1955年生，教授，研究方向为通信软件和网络管理。

江彦馥：女，1993年生，硕士生，研究方向为AdHoc网络管理与优化。

刘彦君：男，1964年生，高级工程师，研究方向为电力通信网络管理。

苏 汉：男，1977年生，高级工程师，研究方向为电力通信网络管理。

徐思雅：女，1988年生，博士后，研究方向为智能电力通信网络管理与优化。

元 峰：男，1971年生，教授，研究方向为电力通信网络管理。