

## 航空自组网中面向容错的中继节点速度控制\*

李杰, 宫二玲, 孙志强, 刘伟, 谢红卫

(国防科技大学 机电工程与自动化学院, 湖南长沙 410073)

**摘要:**由于飞机节点的通信距离有限,航空自组网的网络拓扑高动态的变化会导致频繁的网络分割并严重影响网络上层应用的正常运行。为了保证飞机节点之间端到端的连通性不受影响,航空自组网必须具备容错性,即任意一个节点或链路失效后网络仍然连通。通常情况下飞机节点的运动不可控,因此可在网络中加入一定数量的中继节点,通过控制中继节点的运动速度来实现并维持航空自组网的容错性。提出了一种在线中继节点速度控制方法,该方法根据网络当前状态计算出中继节点的最佳运动方式,在保证网络容错的前提下使得中继节点在网络运行时间内所运动的总路程最短。仿真结果表明该中继节点速度控制方法在航空自组网的容错控制方面具有潜在的应用前景。

**关键词:**自组织网络;容错设计;网络连通性;速度控制

**中图分类号:**TP393 **文献标志码:**A **文章编号:**1001-2486(2015)04-158-07

## Relay speed control for realization of fault-tolerant aeronautical ad hoc networks

LI Jie, GONG Erling, SUN Zhiqiang, LIU Wei, XIE Hongwei

(College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha 410073, China)

**Abstract:** Due to the limited transmission range of airborne nodes, aeronautical ad hoc network suffers from frequent network partitioning in the highly-dynamic environment, which may affect the operation of network applications. In order to guarantee end-to-end connectivity, aeronautical ad hoc network should have the ability of fault-tolerance against link or node failures. Therefore, one or more relay nodes are required for constructing such a fault-tolerant network. As airborne nodes move, relay nodes need to move as well in order to re-establish the topology as quickly as possible. An online algorithm is proposed for relay nodes' speed control to realize the network fault-tolerant during running time. Based on the network's actual state, the online algorithm calculates relay nodes' velocities such that the network can keep fault-tolerance and relay nodes, can travel a short total distance during the running time. Simulations demonstrate that the proposed algorithm is of great potential to be applied to aeronautical ad hoc networks.

**Key words:** ad hoc networks; fault-tolerant design; network connectivity; speed control

航空自组网(Aeronautical Ad hoc NETWORK, AANET)<sup>[1]</sup>是在配备无线通信设备的飞机之间形成一种特殊的无中心移动自组织网络(Mobile Ad hoc NETWORK, MANET),为飞机间提供直接的通信服务。AANET的基本思想是在飞机的通信范围内,飞机之间可以相互交换控制信息和命令数据,而在通信范围之外的飞机可以通过多跳方式传递数据,形成一个空中的MANET。在AANET中,每个飞机不仅仅是收发器,而且还可以起到路由器的作用来转发数据。与简单地利用飞行器作为中继节点进行通信的方式不同,AANET采用动态组网、动态路由和无线中继等技术,将航空飞行器互连互通,具备自组织、自修复的能力和快速、

高效组网的优势,可满足特定条件下的军、民航通信的需求<sup>[2]</sup>。

在AANET中,飞机节点的高速运动使得网络拓扑高动态地变化。由于飞机的通信距离有限,在节点较为稀疏的区域,AANET则无法保证飞机之间持续的端到端可靠连接。延迟容忍网络(Delay-Tolerant Network, DTN)<sup>[3]</sup>技术可以解决网络频繁分割状态下的数据通信问题。DTN利用网络节点的存储空间对数据进行暂时缓存,并基于“存储-携带-转发”的路由模式实现节点间的通信。然而DTN的这种路由模式会带来较大的端到端延迟。AANET中的多数应用对实时性要求很高,所以DTN技术并不能解决AANET的

\* 收稿日期:2014-10-15

作者简介:李杰(1985—),男,江苏徐州人,博士研究生,E-mail:ljkjhk@126.com;

谢红卫(通信作者),男,教授,博士,博士生导师,E-mail:sunzq@nudt.edu.cn

实时通信问题。此外,当 AANET 应用于战场环境等一些恶劣的网络环境时,网络还要求具有一定的容错性,即部分节点或通信链路失效不会对网络的连通性产生影响。

实现容错网络的拓扑控制方法可分为三类。第一类是通过控制天线的发射功率来改变节点的最大通信距离,进而改变网络的拓扑结构。其控制目标是使得网络的拓扑结构达到容错性要求的同时,网络消耗的能量最少。该类方法在无线传感器网络(Wireless Sensor Network, WSN)中应用较多<sup>[4-5]</sup>。而在 AANET 中,节点的分布区域较广,即使使用最大的发射功率,网络拓扑也无法达到容错性要求。如果网络中节点的运动方式可控,则可用第二类方法对网络拓扑进行控制,即:通过控制网络节点的运动,使得网络拓扑实现容错性要求<sup>[6]</sup>。第三类方法是向网络中加入额外的中继节点来重构网络的拓扑结构,使得网络拓扑结构满足容错性要求<sup>[7-10]</sup>。因此,可以在 AANET 中网络连通性差的区域加入额外的中继节点(长航时的无人机或其他可控空中平台)来增强网络的连通性<sup>[11]</sup>。由于网络拓扑的不断变化,必须通过控制这些中继节点的运动速度来保持网络的容错性要求。考虑到每个中继节点的能量有限,为了延长其工作时间,在保证网络容错性要求的前提下还须使得中继节点总的运动路程最短。然而,现有的通过中继节点进行拓扑控制的方法大都是针对静态网络来合理配置中继节点的位置<sup>[7-8]</sup>,或是在网络拓扑缓慢变化的网络中控制中继节点的运动<sup>[9-10]</sup>。这些方法都无法应用于网络拓扑剧烈变化的 AANET 中。

## 1 问题描述

假设在二维平面上有  $w$  个飞机节点(Airborne Node, AN)和  $m$  个中继节点(Relay Node, RN)。AN 和 RN 都配备了最大通信距离为  $r$  的全向天线。令 AN 的集合记为  $V_A = \{a_1, a_2, \dots, a_w\}$ , 其中  $a_i (1 \leq i \leq w)$  表示第  $i$  个飞机节点。令 RN 的集合记为  $V_R = \{r_1, r_2, \dots, r_m\}$ , 其中  $r_j (1 \leq j \leq m)$  表示第  $j$  个中继节点。在本文中,某网络节点在二维平面中的位置坐标用表示该节点字符的黑体形式来表示。如,节点  $a_i$  的位置坐标用矢量  $\mathbf{a}_i = (x_{a_i}, y_{a_i})$  表示,节点  $r_j$  的位置坐标用矢量  $\mathbf{r}_j = (x_{r_j}, y_{r_j})$  表示。位置矩阵  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_w]$  表示 AN 集合  $V_A$  的位置坐标,位置矩阵  $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_m]$  则表示 RN 集合  $V_R$  的位置坐标。节点  $r_j$  在  $t$  时刻的瞬时速度用矢量  $\mathbf{v}_j(t)$  表示。所有

RN 在  $t$  时刻的速度用速度矩阵  $\mathbf{V}(t) = [\mathbf{v}_1(t), \mathbf{v}_2(t), \dots, \mathbf{v}_m(t)]$  表示。给定 AN 的集合  $V_A$ , RN 的集合  $V_R$ , 以及通信距离  $r$ , 其拓扑可以用无向图  $G = (V, E)$  来表示, 其中  $V = V_A \cup V_R$  为顶点集, 表示网络中的空中平台;  $E$  为边集, 表示网络中任意两个距离小于  $r$  的节点之间形成的通信链路的集合。网络拓扑的容错性可用图论中的顶点连通度来度量。如果图  $G$  中任意两个顶点之间都至少有  $k$  条内部顶点不相交的路径, 则称图  $G$  为顶点  $k$  连通。当  $k = 1$  时, 图  $G$  为简单连通; 当  $k \geq 2$  时, 称图  $G$  具有容错能力, 此时网络的抗毁性较好。由于 AANET 只要保证 AN 之间的通信具有容错性即可。因此, 在下文提到图  $G$  为顶点  $k$  连通时则表示任意两个 AN 之间至少有  $k$  条内部顶点不相交的路径。若网络总的运行时间为  $T$ , 网络的容错控制问题可以转化为求 RN 的速度矩阵  $\mathbf{V}(t), 0 < t \leq T$ , 使得在保证网络容错的前提下, 所有 RN 在时间  $T$  内总的移动距离最短。因此, 可以抽象为式(1)~(3)的优化问题。

$$\min_{\mathbf{V}(t)} \sum_{i=1}^m \int_0^T \|\mathbf{v}_i(t)\| dt \quad (1)$$

$$\text{s. t. } \lambda_i(u, v) \geq 2 \quad (2)$$

$$\forall u, v \in V_A, 0 < t \leq T$$

$$0 \leq \|\mathbf{v}_i(t)\| \leq v_{\max} \quad (3)$$

$$\forall i \in \{1, 2, 3, \dots, m\}, 0 < t \leq T$$

式中:  $v_{\max}$  表示 RN 的最大速度,  $\lambda_i(u, v)$  表示在  $t$  时刻点  $u$  和  $v$  之间内部顶点不相交路径的数量。

通常情况下, 网络中 AN 的运动不可预测, 无法通过离线的方式通过求解式(1)~(3)得出 RN 的速度矩阵  $\mathbf{V}(t)$ 。因此, 需要采用在线的方式根据网络在  $t$  时刻的瞬时状态求出 RN 在  $t$  时刻的速度矩阵  $\mathbf{V}(t)$ 。AN 和 RN 在  $t$  时刻的位置分别用矩阵  $\mathbf{A}(t)$  和  $\mathbf{R}(t)$  表示。矩阵  $\dot{\mathbf{A}}(t)$  表示 AN 在  $t$  时刻的瞬时速度矩阵。则  $t$  时刻的网络状态可以用  $\{\mathbf{A}(t), \dot{\mathbf{A}}(t), \mathbf{R}(t)\}$  三个变量表示。AANET 网络拓扑的在线控制问题可以抽象为图 1 所示的闭环控制模型。控制器根据网络的连通性要求(容错网络)和  $t$  时刻的网络状态求出  $t$  时刻 RN 的速度矩阵  $\mathbf{V}(t)$ , 使得网络满足容错性的前提下, 所有 RN 在网络运行时间内总的运动路程最短。

## 2 在线拓扑控制

要实现在线拓扑控制, 就需要根据网络的实时状态得出 RN 速度矩阵的最优解。用采样间隔时间  $\Delta t$  将时间  $t$  离散化为时间序列  $n = 0, 1, \dots,$

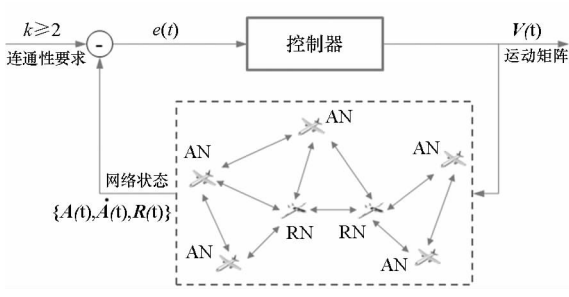


图 1 在线拓扑控制

Fig. 1 Online topology control

$K$ , 其中,  $K = \lfloor T/\Delta t \rfloor$ 。从而式(1) ~ (3)可离散化为

$$\min_{\{V[0], V[1], \dots, V[K]\}} \sum_{n=0}^K \sum_{i=1}^m \|v_i[n]\| \Delta t \quad (4)$$

$$\text{s. t. } \lambda_n(u, v) \geq 2 \quad (5)$$

$$\forall u, v \in V_A, n = 0, 1, 2, \dots, K$$

$$0 \leq \|v_i[n]\| \leq v_{\max} \quad (6)$$

$$\forall i \in \{1, 2, 3, \dots, m\}, n = 0, 1, 2, \dots, K$$

式中:  $V[n]$  和  $\lambda_n(u, v)$  分别为  $V(t)$  和  $\lambda_i(u, v)$  在时刻  $t = \Delta t \cdot n$  的采样值。对于任意  $n$ , 若求出  $V[n] = [v_1[n], v_2[n], \dots, v_m[n]]$  使得  $\sum_{i=1}^m \|v_i[n]\| \Delta t$  值最小, 则式(4) 可以取得较小的值。因此, 可以推出一个在线拓扑控制的近似算法: 根据采样时刻  $n$  时 AN 的位置和速度, 估计下一个采样时刻  $n + 1$  时 AN 的位置, 并根据 RN 在时刻  $n$  的位置求出 RN 的速度矩阵  $V[n]$ , 使得在  $n + 1$  时刻网络为容错网络, 并且  $\sum_{i=1}^m \|v_i[n]\| \Delta t$  的值最小。

**定义:** 对于四元组  $(V_A, A, V_R, R)$ , 当 RN 以移动矩阵  $D = [d_1, d_2, \dots, d_m]$  移动到新的位置  $R + D$ , 若此时 RN 和 AN 构成的无向图  $G = (V_A \cup V_R, E)$  具有容错性, 则称矩阵  $D$  为四元组  $(V_A, A, V_R, R)$  的可行移动矩阵。若某个可行移动矩阵总移动距离  $\sum_{i=1}^m \|d_i\|$  的值最小, 则该移动矩阵称为最小成本可行移动矩阵 (Minimum Cost Feasible moving Matrix, MCFM)。求解四元组  $(V_A, A, V_R, R)$  最小成本可行移动矩阵的问题记作  $MCFM(V_A, A, V_R, R)$ 。

令  $A[n] = [a_1[n], a_2[n], \dots, a_w[n]]$  和  $R[n] = [r_1[n], r_2[n], \dots, r_m[n]]$  分别表示 AN 和 RN 在采样时刻  $n$  时的位置矩阵。AN 在采样时刻  $n$  时的速度用矩阵  $\dot{A}[n] = [\dot{a}_1[n], \dot{a}_2[n], \dots, \dot{a}_w[n]]$  表示。若在采样时刻  $n$ , 网络当前的状态

值  $\{A[n], \dot{A}[n], R[n]\}$  可以精确得到。则图 1 所示的 RN 运动控制问题如算法 1 所示。

算法 1 RN 运动的在线控制

Alg. 1 Online approach for movement control of RN

1. 在采样时刻  $n$  获取 AN 的位置矩阵  $A[n]$ , 速度矩阵  $\dot{A}[n]$  以及 RN 的位置矩阵  $R[n]$  的值
2. 估计 AN 在下一采样时刻  $n + 1$  时的位置矩阵  $\bar{A}[n + 1] = A[n] + \Delta t \times \dot{A}[n]$
3. 计算  $n$  时刻 RN 的移动矩阵  $D[n] = MCFM(V_A, \bar{A}[n + 1], V_R, R[n])$
4. 计算  $n$  时刻 RN 的速度矩阵  $V[n] = D[n] / \Delta t$ , 若  $V[n]$  中存在速度大于  $v_{\max}$  的分量  $v_i[n]$ , 则设置  $\|v_i[n]\| = v_{\max}$
5. 设置  $n = n + 1$ , 并转至步骤 1

3 MCFM 问题的求解算法

3.1 算法提出

解决 MCFM 问题的基本思路是“Steiner 化”。Steiner 化的概念由 Lin 和 Xue 首次提出<sup>[12]</sup>, 并广泛应用于通信网络中继节点配置问题中<sup>[13]</sup>。对于平面上的两个点  $x$  和  $y$ , 如果两点之间距离大于最大通信距离  $r$ , 则它们之间无法直接通信。令  $\|x - y\|$  表示点  $x$  和  $y$  之间的欧氏距离。若  $\|x - y\| > r$ , 则可以在边  $(x, y)$  上以相等间隔  $\lceil \|x - y\|/r \rceil$  设置  $\lceil \|x - y\|/r \rceil - 1$  个中继节点, 使得点  $x$  和  $y$  可以通过这些中继节点相互通信。这些中继节点称为 Steiner 点 (Steiner Point, SP), 并将上述过程称作对边  $(x, y)$  Steiner 化。通过对边  $(x, y)$  Steiner 化, 得到 SP 的集合记为  $SP(x, y)$ 。显然, 若  $\|x - y\| \leq r, SP(x, y) = \emptyset$ 。

若  $s_i$  表示某个 SP, 为了体现 RN 移动  $s_i$  的位置所要运动的平均距离, 定义其成本  $c(s_i)$  为所有 RN 到  $s_i$  的平均距离, 即:

$$c(s_i) = \sum_{r_j \in V_R} \|s_i - r_j\| / |V_R| \quad (7)$$

构造完全图  $G_C(V_A, E_C)$ , 其中边集  $E_C$  表示任意两个  $V_A$  中的点所形成边的集合。对于  $E_C$  中的边  $(a_i, a_j)$  定义其权值  $c(a_i, a_j)$  为:

$$c(a_i, a_j) = \sum_{s_i \in SP(a_i, a_j)} c(s_i) \quad (8)$$

因此 MCFM 问题可以转化为求加权完全图  $G_C(V_A, E_C)$  最小成本顶点 2 连通子图  $G_S(V_A, E_S)$  的问题。对  $E_S$  中所有的边 Steiner 化, 得到 SP 的集合记为  $V_S$ 。如果允许 SP 和其他在通信范围  $r$  内的顶点也可以形成边, 则可构成图  $G'(V_A \cup V_S,$

$E(G')$ )。然而,  $V_S$  中的一些顶点对于实现图  $G'$  ( $V_A \cup V_S, E(G')$ ) 顶点 2 连通并不是必要的。因此需要对  $V_S$  内每个 SP 进行必要性检测, 并优先去除成本大的 SP。此时, 若得到的  $V_S = \{s_1, s_2, \dots, s_h\}$ , 则 RN 可以移动到  $V_S$  中的 SP 位置来实现网络的容错性。而且 RN 需要移动到  $V_S$  中合适的 SP 位置, 使得所有 RN 总的移动距离最短。将该问题抽象为集合  $V_R$  与  $V_S$  中点的最优分配问题: 为每一个 RN 分配一个 SP, 并使得  $V_R$  中的 RN 移动到各自所分配到的 SP 位置时总移动距离最短。这可以用图论中的最大权匹配问题来解决, 建立完备加权二部图  $K_{|V_R|, |V_S|}$ , 如图 2 所示。令  $E(K)$  为  $K_{|V_R|, |V_S|}$  的中边的集合,  $E(K)$  中边  $(r_i, s_j)$  权值函数  $W(r_i, s_j)$  定义为

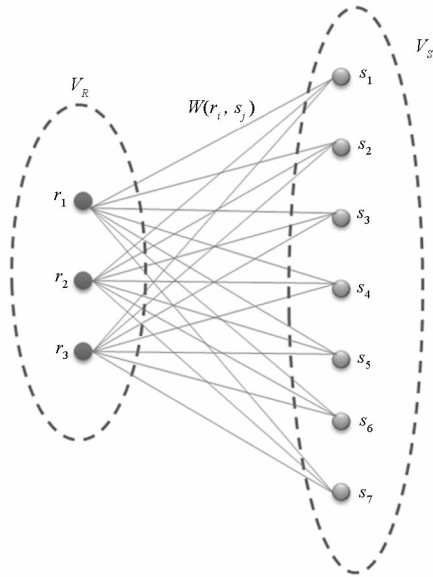


图 2 完备加权二部图  $K_{|V_R|, |V_S|}$

Fig. 2 Weighted complete bipartite graph  $K_{|V_R|, |V_S|}$

$$W(r_i, s_j) = \left( \max_{(r_p, s_q) \in E(K)} \|s_q - r_p\| \right) - \|s_j - r_i\| \quad (9)$$

应用 Kuhn-munkres 算法<sup>[15]</sup> 计算出  $K_{|V_R|, |V_S|}$  的最大权匹配  $M$ 。如果  $r_i$  在  $M$  中有匹配  $s_j$ , 则移动向量  $d_i = s_j - r_i$ , 否则  $d_i = \mathbf{0}$ 。从而根据匹配  $M$  可以求出 RN 的具有较小总移动距离的可行移动矩阵  $D = [d_1, d_2, \dots, d_m]$ 。因而, 可以得到如算法 2 所示的 MCFM 问题的快速求解算法。

算法 2 所示的 MCFM 问题求解算法能够求出具有较小总移动距离的可行移动矩阵  $D$ , 然而在某些情况下, 算法 2 并不能获得较好的性能。如图 3 所示, 3 个 AN ( $a_1, a_2$  和  $a_3$ ) 组成边长为  $1.72r$  的等边三角形, 则三角形的中心位置坐标  $s$  距 3 个顶点的距离都为  $r$ 。三角形 3 条边的中点

算法 2 MCFM 问题的快速算法

Alg. 2 Fast algorithm for MCFM problem

- 输入: RN 的集合  $V_R$  和位置矩阵  $R$ ; AN 的集合  $V_A$  和位置矩阵  $A$
- 输出: RN 的移动矩阵  $D = [d_1, d_2, \dots, d_m]$
1. 构建加权完全图  $G_C(V_A, E_C)$
  2. 计算  $G_C$  的最小成本顶点 2 连通子图  $G_S(V_A, E_S)$
  3. 对  $E_S$  中的边 Steiner 化得到 SP 的集合  $V_S$
  4. **for**  $s \in V_S$  按照权值  $c(s)$  由大到小顺序 **do**
  5.      $G' = (V_A \cup V_S - \{s\}, E(G'))$
  6.     **if**  $G'$  是顶点 2 连通 **then**
  7.          $V_S \leftarrow V_S - \{s\}$
  8.     **end if**
  9. **end for**
  10. 构建完备加权二部图  $K_{|V_R|, |V_S|}$
  11. 在二部图  $K_{|V_R|, |V_S|}$  上执行最大权匹配算法得到匹配  $M$
  12. **for**  $r_i \in V_R$  **do**
  13.     **if**  $r_i$  在  $M$  中有匹配的点  $s \in V_S$  **then**
  14.         设置  $d_i = s - r_i$
  15.     **else**
  16.         设置  $d_i = \mathbf{0}$
  17.     **end if**
  18. **end for**

位置分别记为  $s_1, s_2$  和  $s_3$ 。三个 RN ( $r_1, r_2$  和  $r_3$ ) 在相同的位置  $p$  处, 且距中点的距离为  $d$ , 即  $\|s - p\| = d$ 。可以看出, 此时网络并不连通。执行算法 2 后,  $r_1, r_2$  和  $r_3$  将分别移动到位置  $s_1, s_2$  和  $s_3$  使得网络顶点 2 连通, 即移动矩阵  $D = [s_1 - p, s_2 - p, s_3 - p]$ 。若  $d$  的值很小, 则总的移动代价为  $\|s_1 - p\| + \|s_2 - p\| + \|s_3 - p\| \approx 1.5r$ 。而最优解为三个 RN 任意两个移动到中点位置  $s$  即可, 即移动矩阵  $D = [s - p, s - p, \mathbf{0}]$ , 移动代价为  $2d$ 。由于  $d$  的值可以无限小, 所以算法产生的解的代价和最

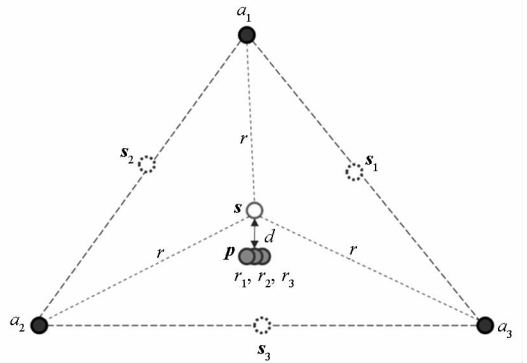


图 3 算法 2 具有较差性能的场景

Fig. 3 An example of poor performance by Alg. 2

优解代价的比值  $0.75r/d$  可以趋于无穷大。

### 3.2 算法的复杂度分析

完全图  $G_c(V_A, E_c)$  中边的数量  $|E_c| = O(|V_A|^2)$ 。利用文献[14]求完全图  $G_c$  的最小成本顶点 2 连通子图的时间复杂度为  $O(2|V_A|^4)$ 。假设网络的运行在边长  $L$  的正方形区域,完全图  $G_c(V_A, E_c)$  的 2 连通子图最多有  $2(|V_A| - 1)$  条边<sup>[16]</sup>。由于每条边 Steiner 化后 SP 的数量为  $O(L/r)$ ,则第 3 步得到的候选 SP 的集合大小  $|V_S| = O(L|V_A|/r)$ 。考虑到对具有  $m$  个顶点的无向图进行连通性检测的时间复杂度为  $O(m^2)$ ,则算法 2 中步骤 4~9 的剪枝过程时间复杂度为  $O(L|V_A|/r)^3$ 。在具有  $n$  个顶点的二部图上执行最大权匹配的时间复杂度为  $O(n^{2.5})$ ,则第 11 步的时间复杂度为  $O(|V_R|)^{2.5}$ 。步骤 12~18 计算移动矩阵  $D$  所花费时间为  $O(|V_R|)$ 。所以,算法 2 是具有时间复杂度为  $O(2|V_A|^4 + (L|V_A|/r)^3 + |V_R|^{2.5})$  的多项式时间算法。算法中存储空间开销主要用于存储图  $G' = (V_A \cup V_S, E(G'))$  和加权二部图  $K_{|V_R|, |V_S|}$  的邻接矩阵。图  $G'$  的二维邻接矩阵是一个  $(|V_A| + |V_S|) \times (|V_A| + |V_S|)$  的二维数组,而图  $K_{|V_R|, |V_S|}$  的二维邻接矩阵是一个  $|V_S| \times |V_S|$  的二维数组。考虑到  $|V_S| = O(L|V_A|/r)$ ,所以算法的空间复杂度为  $O(|V_A|^2)$ 。

### 4 仿真验证

假设  $w$  个 AN 以速度  $v$  在边长为 400km 的矩形区域运动。用随机路点 (Random Way Point, RWP)<sup>[17]</sup> 运动模型作为 AN 的运动模型,并将停止时间设为 0,来逼真地模拟实际飞机的运动模式。为了提高网络的连通性,在网络中加入  $m$  个最大速度  $v_{max} = 800m/s$  的 RN,并设置速度更新间隔  $\Delta t = 8s$ 。网络中所有节点的最大通信距离  $r = 100km$ 。总的仿真时间设置为 30min。如图 4 所示,基于网络仿真平台 OMNeT++<sup>[18]</sup> 建立仿真模型。在图 4 所示的网络模型中,“relay”模块代表 RN,“host”模块代表 AN。图 5 描绘了当  $v = 300m/s, w = 12, m = 5$  时 3 个不同仿真时刻的网络的拓扑图。

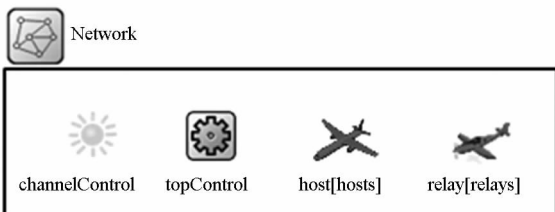
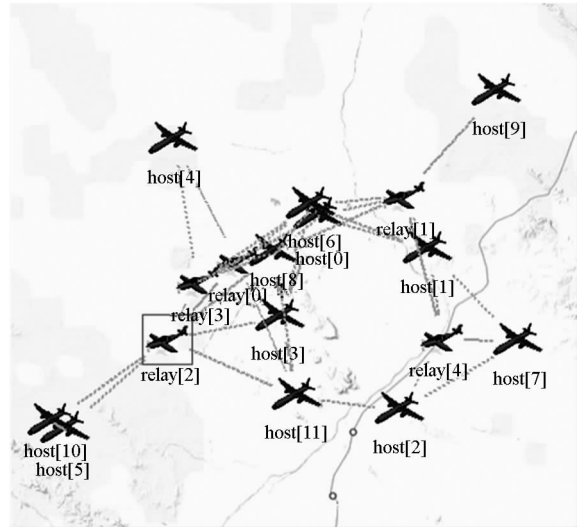


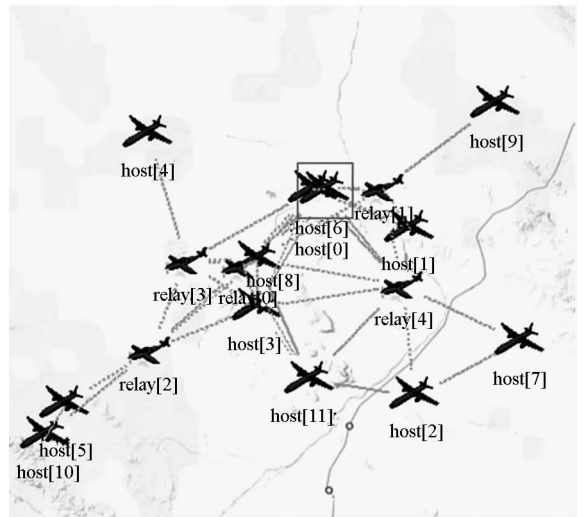
图 4 仿真模型

Fig. 4 Simulation model



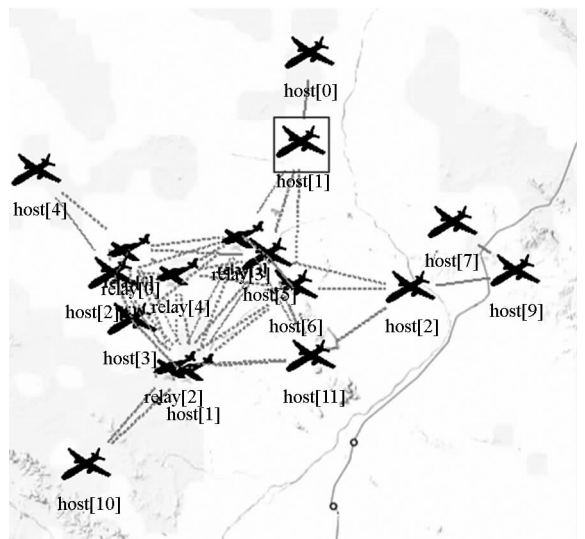
(a) 仿真时刻为 200s

(a) At simulation time of 200s



(b) 仿真时刻为 260s

(b) At simulation time of 260s



(c) 仿真时刻为 600s

(c) At simulation time of 600s

图 5 网络拓扑

Fig. 5 Network topology

为了方便研究算法性能,定义以下几个性能指标。

1) RN 总的运动路程 (Total Travel Distance, TTD):  $TTD$  值反映了 RN 在网络运行期间所消耗的总能量。为了延长 RN 的工作时间,要求  $TTD$  要尽量的小。

2) 2 连通率 (Biconnectivity Rate, BR):  $BR$  定义为网络为顶点 2 连通的时间与网络总的运行时间比值。 $BR$  值越大则说明网络的容错性越好。

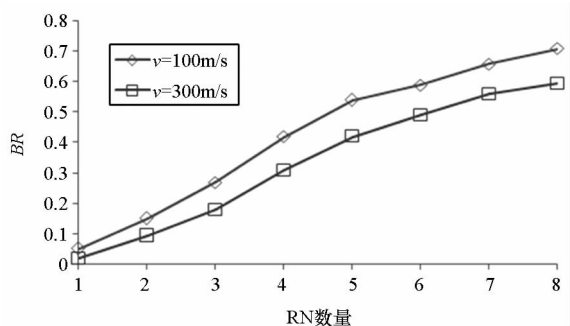
3) 相对 2 连通率 (Relative Biconnectivity Rate, RBR):  $RBR$  值反映了 RN 的运动对网络 2 连通率的影响。若向具有  $w$  个 AN 的网络中加入  $m$  个 RN, 当前网络的 2 连通率和原网络相比所提升的部分除了通过 RN 的运动所提升的部分外还包括由于网络节点密度增加所提升的值。 $RBR$  值可以由网络当前的  $BR$  值减去当网络中只有  $w + m$  个 AN 时的值得到。若令  $BR_0$  表示当网络中只有  $w + m$  个 AN 时的 2 连通率, 则有

$$RBR = BR - BR_0 \quad (10)$$

#### 4.1 性能分析

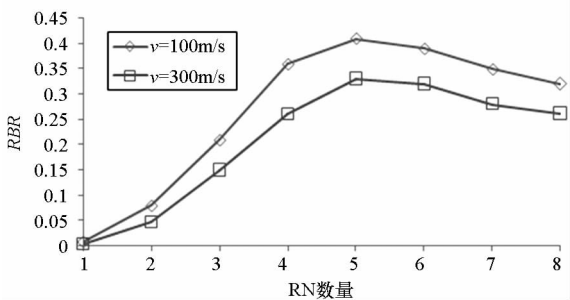
为了便于对算法性能进行分析,网络中 AN 数量要选择合适的值以便能产生显著的控制效果。由于网络  $BR$  值与网络中节点密度有关,为了方便研究 RN 运动对网络性能的影响,网络的初始  $BR$  值不能太大。因此,AN 的数量太大会导致较大的初始  $BR$  值,则算法不会产生显著的控制效果。而 AN 的数量过少则无法证明算法在实际 AANET 中的有效性。实验证明,当网络中 AN 数为 12 左右时,网络具有较小的  $BR$  值(0.05 左右)。因此,在下面的仿真实验中设置  $w = 12$ 。由于当前飞机的巡航速度一般在 1 马赫以下,所以在对算法性能分析时只考虑 AN 以亚音速巡航的场景。研究不同数量的 RN 在低动态环境 ( $v = 100\text{m/s}$ ) 和高动态环境 ( $v = 300\text{m/s}$ ) 下对网络性能的影响。RN 的数量与各个性能指标的关系如图 6 所示。如图 6(a)所示,网络的 2 连通率随着 RN 的增加而显著增大,而低动态的网络环境更容易获得较高的  $BR$  值。这也证明该 RN 运动控制算法在 AANET 的容错控制方面的有效性。从图 6(b)可以看出,网络的相对 2 连通率并不是随着 RN 数量的增加而单调增加,而是在 RN 的数量大于 5 后出现减少的趋势。所以,当 RN 数量增加到一定值之后,网络  $BR$  值的提高更多的是依赖网络节点密度的增加。根据式(10)可知,当网络的节点总数达到一定数量之后, $BR_0$  值的增加占主要部分,使得  $RBR$  值出现减少的趋势。而

$RBR$  值依赖于 RN 的运动,所以当 RN 的数量大于 5 后, $TTD$  的值随着 RN 数目增加的趋势也变得不显著,如图 6(c)所示。



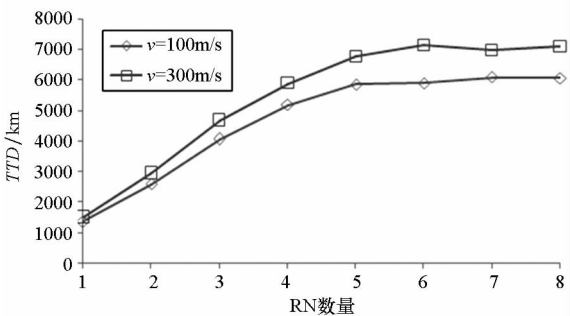
(a) 网络 2 连通率

(a) Biconnectivity rate of the network



(b) 网络相对 2 连通率

(b) Relative biconnectivity rate of the network



(c) RN 总的运动路程

(c) Total travel distance of RNs

图 6 性能分析

Fig. 6 Performance analysis

#### 4.2 性能比较

令  $w = 12, m = 6$ , 比较本文的算法和文献[9]中提出的算法的性能。只考虑  $TTD$  和  $BR$  两个性能指标。令  $TTD$  和  $TTD^*$  分别表示本文算法和文献[9]中提出的算法的 RN 移动总距离; $BR$  和  $BR^*$  分别表示本文算法和文献[9]中提出的算法的网络 2 连通率。比较比值  $TTD/TTD^*$  和  $BR/BR^*$  在不同网络动态中的值。从图 7 可以看出,随着 AN 运动速度的提高,本文的算法可以取得较好的性能。这是由于文献[9]中提出的算法是

针对当前时刻的网络状态来确定中继节点的最新位置。然而,在高动态的网络中,RN 移动到目标位置时,网络的状态早已改变,所以该算法在高动态的航空网络环境中并不能获得较好的性能。

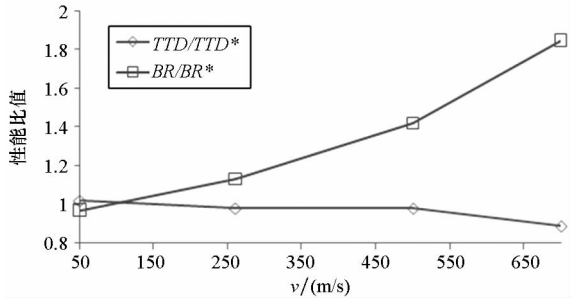


图 7 性能比较

Fig. 7 Performance comparison

### 5 结论

1) 针对 AANET 的特点提出一种基于中继节点的网络容错控制方法。该控制方法采用在线控制 RN 运动速度的方式,利用较小的运动总路程实现网络的容错性。

2) 将 RN 的运动速度控制问题抽象为求解最小成本可行移动矩阵的问题,并给出了适用于在线控制的快速计算方法。

3) 仿真结果表明该拓扑控制方法在 AANET 的容错控制方面具有潜在的应用前景。

### 参考文献 (References)

[1] Sakhaee E, Jamalipour A. The global in-flight internet [J]. IEEE Journal on Selected Areas in Communications, 2006, 24(9): 1748 - 1757.

[2] 郑博, 张衡阳, 黄国策, 等. 航空自组网的现状与发展[J]. 电信科学, 2011, 27(5): 38 - 47.  
ZHENG Bo, ZHANG Hengyang, HUANG Guoce, et al. Status and development of aeronautical ad hoc networks [J]. Telecommunications Science, 2011, 27(5): 38 - 47. (in Chinese)

[3] Fall K. A delay-tolerant network architecture for challenged internets [C]//Proceedings of ACM International Conference on the Applications, Technologies, Architectures, and Protocols for Computer Communication, Karlsruhe, Germany: ACM, 2003: 27 - 34.

[4] Aziz A A, Sekercioglu Y A, Fitzpatrick P, et al. A survey on

distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks [J]. IEEE Communications Surveys & Tutorials, 2013, 15(1): 121 - 144.

[5] Nishiyama H, Ngo T, Ansari N, et al. On minimizing the impact of mobility on topology control in mobile ad hoc networks [J]. IEEE Transactions on Wireless Communications, 2012, 11(3): 1158 - 1166.

[6] Das S, Liu H, Nayak A, et al. A localized algorithm for bi-connectivity of connected mobile robots [J]. Telecommunication Systems, 2009, 40(3 - 4): 129 - 140.

[7] Nigam A, Agarwal Y K. Optimal relay node placement in delay constrained wireless sensor network design [J]. European Journal of Operational Research, 2014, 233(1): 220 - 233.

[8] Kashyap A, Khuller S, Shayman M. Relay placement for fault tolerance in wireless networks in higher dimensions [J]. Computational Geometry, 2011, 44(4): 206 - 215.

[9] Kashyap A, Shayman M. Relay placement and movement control for realization of fault-tolerant ad hoc networks [C]//Proceedings of 41st Annual Conference on Information Sciences and Systems, Baltimore, MD: IEEE, 2007: 783 - 788.

[10] Senturk I F, Akkaya K, Yilmaz S. Relay placement for restoring connectivity in partitioned wireless sensor networks under limited information [J]. Ad Hoc Networks, 2014, 13: 487 - 503.

[11] Rohrer J P, Jabbar A, Cetinkaya E K, et al. Highly-dynamic cross-layered aeronautical network architecture [J]. IEEE Transactions on Aerospace and Electronic System, 2011, 47(4): 2742 - 2765.

[12] Lin G H, Xue G L. Steiner tree problem with minimum number of Steiner points and bounded edge-length [J]. Information Processing Letters, 1999, 69(2): 53 - 57.

[13] Degener B, Fekete S P, Kempkes B, et al. A survey on relay placement with runtime and approximation guarantees [J]. Computer Science Review, 2011, 5(1): 57 - 68.

[14] Khuller S, Raghavachari B. Improved approximation algorithms for uniform connectivity problems [J]. Journal of Algorithms, 1996, 21(2): 434 - 450.

[15] West D B. Introduction to graph theory [M]. 2nd ed. UK: Pearson Education, 2000: 125 - 130.

[16] Khulle S, Vishkin U. Biconnectivity approximations and graph carvings [J]. Journal of the ACM, 1994, 41(2): 214 - 235.

[17] Camp T, Boleng J, Davies V. A survey of mobility models for ad hoc networks research [J]. Wireless Communications and Mobile Computing, 2002, 2(5): 483 - 502.

[18] OMNeT++ [EB/OL]. OMNeT++ Community, [2014 - 08 - 17]. <http://www.omnetpp.org>.