

文章编号:1009-3087(2014)04-0126-08

无线 Mesh 网络不相交组播路由的构建方法

包学才^{1,2},戴伏生^{1*},韩卫占¹

(1. 哈尔滨工业大学(威海校区), 山东 威海 264209; 2. 南昌工程学院 信息工程学院, 江西 南昌 330099)

摘要:为有效提高无线 Mesh 网络组播路由的拓扑容错能力以及资源利用率,描述了单棵组播树拓扑容错能力的不足,建立了优化模型,提出基于增广不相交路径的不相交组播树构建方法。该方法以无线链路干扰作为链路权重,求解源节点到目的节点以及目的节点之间的不相交路径,通过寻找增广不相交路径来保持各个组播树之间的不相交,最后得到不相交组播树。通过仿真对比分析,提出的方法在目的节点数为 3~10 时,求解多棵不相交组播树的成功比例平均提升 47.13%,而且平均吞吐量和平均时延性能也分别提升了 4.91% 和 5.16%,验证了该方法的正确性和有效性。

关键词:无线 Mesh 网络;不相交组播树;拓扑容错;增广不相交路径

中图分类号:TN915.2;TP393.0

文献标志码:A

Construction Method of Disjoint Multicast Routing for Multi-radio Multi-channel Wireless Mesh Network

BAO Xuecai^{1,2}, DAI Fusheng^{1*}, HAN Weizan¹

(1. Harbin Inst. of Technol., Weihai Campus, Weihai 264209, China;

2. School of Info. Eng., Nanchang Inst. of Technol., Nanchang 330099, China)

Abstract: In order to improve the topology fault-tolerance and resource utilization of multicast routing in wireless Mesh network, the fault-tolerant deficiency of a single multicast tree was described, the optimization model was established, and then the construction method of disjoint multicast tree was proposed based on augmentation disjoint paths. In this method, the wireless link interference was used as the link weights, by which node disjoint paths between the source node and the destination nodes and among the destinations nodes were calculated. Meanwhile, the disjoint among multicast trees was maintained by calculating the augmentation disjoint paths. By simulations and comparative analysis, the success ratio of the proposed method ascended 47.13% for solving the disjoint multicast trees, and the network performance of average throughput and delay also ascended 4.91% and 5.16%, respectively, which demonstrated that the proposed method is correct and valid.

Key words: wireless Mesh network; disjoint multicast trees; topology fault-tolerance; augmentation disjoint path

无线 Mesh 网络组播路由是提高下一代宽带无线 Mesh 网络资源利用率的关键技术之一,当前无线 Mesh 网络的组播路由算法主要是基于多接口多信道无线 Mesh 网络,即每个无线 Mesh 节点上安装多个无线收发器,每个收发器可以从一组载波信道中分配其中一个载波信道。研究方法大体包括 2 个方面:1)组播树构建与信道分配结合的方法^[1-3],主要是先构建一棵组播树,再进行信道分配,但没有考

虑具体的干扰情况。2)不考虑信道分配^[4-9],即在现有的信道分配方法的基础上进行组播树的构建,在构建过程中考虑了无线广播属性、链路干扰、信道数等情况,同时根据定义的链路代价来寻找一棵组播树。目前,这些方法大多是在多接口多信道无线 Mesh 网络上构建单一的组播树,没有考虑节点遭受破坏的情况下组播树的容错能力,即没有实现用于重构或保护路由的容错功能以及建立组播流负载均

收稿日期:2013-08-09

基金项目:通信网信息传输与分发技术国家重点实验室开放课题资助项目(ITD-U11002)

作者简介:包学才(1983—),男,讲师,博士。研究方向:可靠性约束下无线 Mesh 网络资源管理。E-mail:baoxuecai@sina.com

*通信联系人 E-mail:dfs1963@163.com

衡的机制。随着未来点到多点业务的快速发展,提供具有拓扑容错的组播路由算法就显得更为重要,但目前研究这种具有拓扑容错能力的组播路由方法较少,其中的难点是如何更加有效地从网络中找到最多的不相交组播树,目前,未见有效的算法。虽然文献[10-11]提出了针对视频业务的多个组播树方法,但求解的多个组播树是近似不相交的,没有给出多个完全不相交组播树的构建方法。文献[12]提出了几种具有保护措施的路由构建方法,但其中只有一种方法是求解完全节点不相交的组播树(即 NDT 算法),该方法先求出源节点到每个目的节点的最小代价路径,这些最小代价路径组合成一棵组播树,然后删除这棵组播树中的中间节点和链路,再重复计算源节点到每个目的节点的最小代价路径,依次组合成后续的组播树,从而保证每棵组播树是节点不相交的。虽然该方法计算简单,但很大程度上不能有效寻找到多棵不相交组播树,也就是说,任意给定一个网络,即使网络中存在多棵不相交组播树,该方法在很大程度上也只能找到一棵不相交组播树,从而影响了具有拓扑容错能力的组播树构建。所以,针对这种实际网络需求问题,作者提出了更有效的节点不相交组播树路由构建方法,从而能有效提高寻找多棵节点不相交组播树的效率,为建立具有具体拓扑容错能力的组播路由方法提供理论参考。

1 问题描述及相关理论

1.1 问题描述

研究的目的是设计具有容错和最小干扰的组播路由,建立具有一定拓扑容错能力的多接口多信道无线 Mesh 网络组播路由算法,即当组播树上某些节点遭到破坏后,尽可能减少对组播业务的影响,或者具备了备份组播路由的能力,而且建立的组播路由具有最小代价的特性。

如图1所示, s 为组播树中的源节点, r_1 、 r_2 、 r_3 为组播树中的目的节点,当目的节点 r_1 遭破坏或出现故障后,那么后面所有目的节点都无法接收到信息。而构建多个不相交组播树路由在一定程度上具备容错能力,以及可以并行传输相应的业务,从而更好地提高网络资源利用率。同时,如果网络中任意求解的一棵组播树无法满足业务的带宽或时延要求,那么通过多棵组播树同时传输可以更好满足业务的服务质量要求,特别是在视频业务传输方面^[10]。

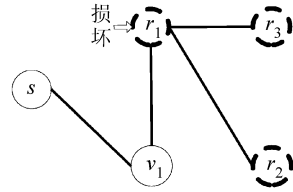


图1 单组播树的拓扑容错性

Fig.1 Performance of fault-tolerance for single multicast

因此,如何有效地构建节点不相交组播树是研究的关键,而目前求解节点不相交组播树的方法^[12]主要是先求出源节点到每个目的节点的最小代价路径,这些最小代价路径组合成一棵组播树,然后删除这棵组播树中的中间节点和链路,再重复相同的步骤计算后面的不相交组播树。但这种方法在很大程度上不能有效地计算多个不相交组播树。

从图2中可以看出,利用文献[12]求解节点不相交组播树时,当计算完第1棵组播树后,剩余网络中就不存在新的组播树,这种方法在很大程度上影响了多棵不相交组播树的构建。因此,如何更有效地构建多棵节点不相交的组播树是问题的关键。但由于是针对多接口多信道的无线 Mesh 网络架构,在求解多棵节点不相交组播树时,要考虑无线节点的干扰,由于每条链路分配的信道有可能不同,每条链路的干扰情况也不同,所以建立的多棵不相交组播树的总链路干扰应该最小化,从而建立最优的多棵不相交组播树。

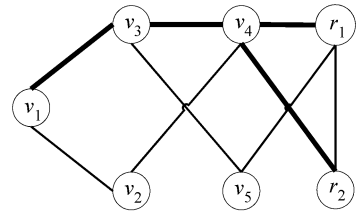


图2 基于最短路径求解的第1棵组播树

Fig.2 First multicast tree based on shortest path algorithm

为了设计有效的多棵不相交组播树构建方法,建立相应的优化模型如下:

$$\min \left\{ \max \left(\sum_{i \leq m, k \in C} I(T_i, k) \right) \right\} \quad (1)$$

$$\text{s. t.} \quad T_i \cap T_j = \{s, M\} \quad (2)$$

式(1)~(2)中:式(1)的优化目标是各组播树上最大干扰最小化,其中, $I(T_i, k)$ 为组播树上已分配信道 k 的链路干扰冲突集(具体计算见文献[18]), T_i 为第 i 棵不相交组播树, k 为信道标号, m 为不相交组播树的数量, C 为可分配信道集;式(2)表示除了源节点和接收节点之外,各组播树上的其他节点及链路是不相交的,其

中, s 为源节点, M 为接收节点的集合。

1.2 相关理论

定义1(拟阵^[13]) 一个拟阵 M 是一个有序对 (E, Γ) , 其中, E 为一个有限集合, $\Gamma \subseteq 2^E$ 为 E 中子集的集合, 满足以下条件:

- 1) $\emptyset \in \Gamma$.
- 2) 若 $I \in \Gamma$, 且 $I' \subseteq I$, 则 $I' \in \Gamma$.
- 3) 若 $I_1, I_2 \in \Gamma$ 且 $|I_1| < |I_2|$, 则存在 $e \in I_2 - I_1$, 使得 $I_1 \cup e \in \Gamma$.

定义2(拟阵划分^[13]) 给定 k 个定义在 E 上的拟阵 $M_i, 1 \leq i \leq k$, 对 E 的一个子集 J , 若能找到 J 的子集 $J_1, J_2, \dots, J_k$, 满足 $J = \bigcup_{i=1}^k J_i$, 并且对每个 $i = 1, 2, \dots, k, J_i \in \Gamma(M_i)$, 则称 J 是可划分子集。

定义3(增广不相交路径序列) 设 $DP = \{P_1, P_2, \dots, P_i, P_m\}$ 为目的节点 i 到源节点和其他目的节点间的不相交路径集合, 即从这些路径集合初始划分的不相交组播树为 $T = \{T_1, T_2, \dots, T_k\}$, 设路径 $x \in T_k, y \in P_i$, 如果 $|T_k - x + y| = |T_k|$ (其中, $|T_k|$ 为树中目前连接接收节点的个数), 则称路径序列 $\{x, y\}$ 为增广不相交路径序列。

2 多节点不相交组播树算法

从第1.1节的优化模型可知, 为构建最小干扰的多个不相交组播树, 主要描述多个不相交组播树的构建思路及具体的实现方法, 并给出算法复杂度分析。

2.1 算法的构建思路

目前, 把网络划分为几个不相交子集的理论, 主要依据来自拟阵理论。在拟阵划分理论中, Edmonds^[14]给出了拟阵划分的描述和算法, 但该算法只适合把网络划分为几个链路不相交的最小生成树, 没有给出子集的闭包算子求解的有效算法, 也未给出算法整体复杂度, 且最小生成树包括网络的所有节点。而对于组播树, 只包含网络部分节点, Edmonds 提出的算法不能用于不相交组播树的划分。由于在网络中直接构建最多的不相交组播树是非常困难的, 对此方面的数学理论分析目前还停留在描述一个 k 连通网络, 证明该网络存在多少链路不相交组播树的上下界问题(也称之为斯坦纳树问题)上, 同样没有给出具体求解的有效算法。而在很多情况下, 如网络中存在多棵节点不相交组播树时, 删除路径方法^[12]也只能求出一棵组播树, 该方法对于任何一个给定的网络, 能求出多少节点不相交组播

树是不确定的。如何依据划分理论来求解多个节点不相交组播树是问题的关键所在, 但直接求解非常困难, 目前也没有有效的方法。所以, 可换个思路解决问题, 分析构建多个不相交组播树与多个不相交生成树的不同点, 该不同点主要在于组播树中目的节点只包含网络中部分节点, 而最小生成树中目的节点包含网络的所有节点。在组播树中, 源节点到目的节点或目的节点之间的链路通常需要通过其他非目的节点, 且在不相交的多个组播树中这些非目的节点是不相交的, 而最小生成树的每条链路是本身不相交的。因此, 先求解网络源节点到其他各个目的节点的最大不相交路径, 再在这些不相交路径中, 依据1.2节的定理及增广不相交路径序列求解最大的节点不相交组播树。下面将详细介绍算法的实现思路和涉及的操作。

算法的总体思路是: 首先, 求解源节点到每个目的节点的最大不相交路径并保存。然后, 在保存的不相交路径中, 删除源节点, 求解目的节点之间的最大不相交路径并保存。接着, 依据拟阵划分理论^[13], 依次添加路径到节点不相交组播树。当出现新添加路径与已添加的路径出现相交情况时, 求解增广不相交路径集。当存在一条增广不相交路径序列时, 根据此增广不相交路径序列对当前构建的组播树进行添加或移除相应路径后, 当前路径可以添加到节点不相交组播树中且保持原有节点不相交组播树之间仍然不相交。

为了便于理解, 简要介绍算法中2种类型的增广不相交路径序列情况:

1) 新路径替换的增广情况

当前需要添加的路径与某棵组播树的一条或多条路径相交时, 依次对每条相交路径求解增广不相交路径序列, 此序列中增加或去除的路径是单条路径。为更清楚地说明路径增广情况, 如图3所示。

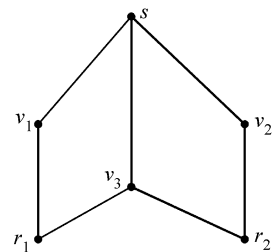


图3 增广不相交序列实例1

Fig.3 Example 1 of augmentation disjoint sequence

设源节点为 s , 2个目的接收节点为 $\{r_1, r_2\}$, s 到 r_1 的不相交路径有 $V_{s1} = \{v_{s11} = \{s, v_1, r_1\}, v_{s12} = \{s, v_3, r_1\}\}$, s 到 r_2 的不相交路径有 $V_{s2} = \{v_{s21} = \{s,$

$v_3, r_2\}, v_{s22} = \{s, v_2, r_2\}\}$ 。删除源节点, r_1 到 r_2 的不相交路径就只有一条即 $V_{12} = V_{21} = \{v_{121} = v_{211} = \{r_1, v_3, r_2\}\}$, $V_{11} = V_{22} = \emptyset$ 。由于不相交组播树的个数不会超过源节点到每个目的节点中最小不相交路径数, 这里不相交路径数为 2, 因此, 设不相交组播树的初始个数为 2, 分别用 T_1, T_2 表示。首先, 依次添加源节点到每个目的节点的一条路径到第 1 棵树中, 则 $T_1 = \{t_{11} = v_{s11}, t_{12} = v_{s21}\}$ 。当准备添加路径 v_{s12} 到 T_2 时, 发现路径 v_{s12} 与已建立的组播树 T_1 中的路径 v_{s21} 相交, 相交节点为 v_3 , 在 v_{s21} 所在的 V_{s2} 中寻找与 v_{s1} 不相交的路径, 通过比较可求解到增广不相交路径序列为 $\{v_{s21}, v_{s22}\}$, 即 $T_1 - v_{s21} + v_{s22}$ 。然后, 将 v_{s12} 添加到 T_2 中, 再将 v_{s21} 添加到 T_2 中, 最后, 得到 2 棵节点不相交的组播树即 $T_1 = \{t_{11} = v_{s11}, t_{12} = v_{s22}\}$, $T_2 = \{t_{21} = v_{s12}, t_{22} = v_{s21}\}$ 。这种单条路径增广情况包含 2 种类型: ①在与当前添加路径相交的路径上寻找增广不相交路径序列, 即上述描述情况; ②与当前添加路径具有相同目的节点路径间的增广不相交路径序列。

2) 不相交组播树之间的增广情况

当在新路径增广情况中, 不存在增广不相交路径序列时, 就在不相交组播树之间求解多路径增广不相交路径序列, 如图 4 所示。

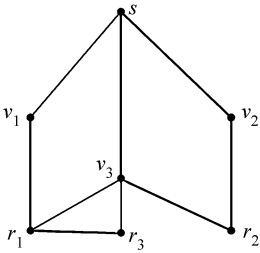


图 4 增广不相交序列实例 2

Fig. 4 Example 2 of augmentation disjoint sequence

图 4 中, 设源节点为 s , 3 个目的接收节点为 $\{r_1, r_2, r_3\}$, s 到 r_1 的不相交路径有 $V_{s1} = \{v_{s11} = \{s, v_1, r_1\}, v_{s12} = \{s, v_3, r_1\}\}$, s 到 r_2 的不相交路径有 $V_{s2} = \{v_{s21} = \{s, v_3, r_2\}, v_{s22} = \{s, v_2, r_2\}\}$, s 到 r_3 的不相交路径有 $V_{s3} = \{v_{s31} = \{s, v_1, r_1, r_3\}, v_{s32} = \{s, v_3, r_3\}\}$ 。删除源节点, r_1 到 r_2 的不相交路径有 $V_{12} = V_{21} = \{v_{121} = v_{211} = \{r_1, v_3, r_2\}\}$, r_1 到 r_3 的不相交路径有 $V_{13} = V_{31} = \{v_{131} = v_{311} = \{r_1, r_3\}\}$, r_2 到 r_3 的不相交路径有 $V_{23} = V_{32} = \{v_{231} = v_{321} = \{r_2, v_3, r_3\}\}$ 。设构建过程中, 当前树的状态是 $T_1 = \{t_{11} = v_{s12}, t_{12} = v_{s21}, t_{13} = v_{s31}\}$, $T_2 = \{t_{21} = v_{s11}, t_{22} = v_{s22}, t_{23} = -1\}$ 。当准备添加路径 v_{s32} 到 T_2 时, v_{s32} 与 T_1 中 v_{s12}, v_{s21} 相交且不存在新路径替换情况下的增广不相交路径序

列, 这时就需要先在不相交组播树之间寻找增广不相交路径序列, 这个序列就是 $\{\{v_{s12}, v_{s21}\}, \{v_{s12}, v_{s21}\}\}$, 即 $T_1 - \{v_{s12}, v_{s21}\} + \{v_{s11}, v_{s22}\}$, $T_2 - \{v_{s11}, v_{s22}\} + \{v_{s12}, v_{s21}\}$, 然后将 v_{s32} 添加到 T_2 , 最后, 得到 2 棵节点不相交的组播树 $T_1 = \{t_{11} = v_{s11}, t_{12} = v_{s22}, t_{13} = v_{s31}\}$, $T_2 = \{t_{21} = v_{s12}, t_{22} = v_{s21}, t_{23} = v_{s32}\}$ 。

2.2 多节点不相交组播树的构建方法

2.1 节对算法的思路及增广不相交路径序列进行了描述, 下面将对整个算法实现进行详细描述。在算法步骤描述前, 对算法中出现的变量进行相应的定义和说明。设源节点为 s ; 目的节点为 $\{r_1, r_2, \dots, r_m\}$, 其中, m 为目的节点的个数; V_{si} 为源节点到目的节点 r_i 的不相交路径集; V_{ij} 为目的节点 r_i 到目的节点 r_j 之间的不相交路径集; v_{ijk} 为目的节点 r_i 到目的节点 r_j 间第 k 条不相交路径; v_{sij} 为源节点 s 到目的节点 r_i 的第 j 条不相交路径; $T_i = \{t_{ik}\}$ 为第 i 棵不相交组播树, t_{ik} 为第 i 棵不相交组播树中连接第 k 个目的节点 r_k 的路径, 其中, $k \leq m$; $J(p, q)$ 为路径 p 与路径 q 的相交情况, 其中, $p, q \in V_{si} \cup V_{ij}$, 如果这 2 条路径相交, $J(p, q)$ 值为 1, 否则, 为 0; 设 $DP = \{V_{s1}, V_{11}, \dots, V_{1m}; V_{s2}, V_{21}, \dots, V_{2m}; \dots; V_{sm}, V_{s1}, \dots, V_{sm}; V_{sm}, V_{m1}, \dots, V_{mm}\} = \{P_1, P_2, \dots, P_i, P_m\}$ 用于表示源节点到每个目的接收节点之间的不相交路径以及接收节点之间的不相交路径集合; Cp, Cq, Cn 为从 P_i 中取出的一条路径。具体的算法步骤如下:

步骤 1: 初始化, $J(i, j) = 0, V_{si} = \emptyset, V_{ij} = \emptyset, T_i = \{t_{ik} = -1\}$ 。

步骤 2: 首先, 计算源节点 s 到每个目的接收节点 $r_1, r_2, \dots, r_m$ 之间的最大不相交路径, 并保存到 $V_{s1}, V_{s2}, \dots, V_{sm} \circ V_{s1}, V_{s2}, \dots, V_{sm}$ 组成一个新网络。接着, 从新网络中删除源节点 s 。然后, 计算 $r_1, r_2, \dots, r_m$ 之间的不相交路径, 并保存到 $V_{11}, \dots, V_{1m}; V_{21}, \dots, V_{2m}; \dots; V_{m1}, \dots, V_{mm}$ 。整理连接到目的节点的路径并保存到 $DP = \{P_1, P_2, \dots, P_i, P_m\}$, 其中, P_i 为能连接目的接收节点 r_i 的路径。

步骤 3: 计算 $DP = \{P_1, P_2, \dots, P_i, P_m\}$ 中每条路径间的相交情况, 并保存到 $J(i, j)$ 中, 用于标记路径间的相交路径, 然后进入步骤 4 进行节点不相交组播树的构建。

步骤 4: 在 $T_i = \{t_{ik}\}$ 中取 t_{ik} 值为 -1 且标号最小的 i, k (表明 i 棵树中与目的节点 r_k 路径还未添加), 确定 i, k 后进入步骤 5; 如果不存在 $t_{ik} = -1$ 的情况, 则转到步骤 9 输出已建立的不相交组播树路径。

步骤5:根据步骤4确定的 k ,在 DP 中取出 P_k ,依次取 P_k 中每一条路径,设取得的路径用 C_p 表示,即 $C_p \in P_k$,接着,判断与 $T_j = \{t_{jp}\} (j \neq i)$ 中每一条路径是否相交(根据 $J(C_p, t_{jp})$ 可直接判断)。如果路径 C_p 与 t_{jp} 相交,进入步骤6;否则, $t_{ik} = C_p$,并返回步骤4。

步骤6:确定 C_p 与 t_{jp} 相交中的 p 值,然后在 DP 中取出 P_p ,在 P_p 上寻找新路径替换增广序列(2.1节有相关描述),如果 P_p 中找到的新路径 C_n 只与树 T_j 的路径相交,与其他树里的路径不相交,检查该路径在树中如果不是孤立路径(即 C_p 与树中其他路径相连并能达源节点),则添加 $T_i = \{t_{ik} = C_n\}$,然后返回步骤4;否则,在 P_p 取一条与当前路径 C_p 不相交的路径 C_q ,并令 $C_p = C_q, j = i$,接着,判断与 $T_j = \{t_{jp}\} (j \neq i)$ 中每一条路径进行是否相交(根据 $J(C_p, t_{jp})$ 可直接判断)。如果路径 C_p 与 t_{jp} 相交,递归返回步骤6,直到找到新路径替换增广不相交序列,并按照序列对相交的树进行添加或去除路径;如果不存在,则进入步骤7。

步骤7:当寻找新路径替换增广序列不成功时,对树 $T_i = \{t_{ik}\}$ 与树 $T_j = \{t_{jp}\}$ 之间进行不相交组播树之间增广不相交序列计算,如果存在,完成操作后,返回步骤4;否则,进入步骤8。

步骤8:如果 C_p 是 P_k 中最后一条取的路径,则进入最后步骤9,否则,返回步骤5。

步骤9:结束上述操作,检查所有不相交组播树 T_i ,对 T_i 中包括所有目的接收节点的树进行输出。

步骤10:算法结束。

2.3 算法实现

无线 Mesh 网络作为下一代宽带无线接入网络,整个无线网络节点是相对静止的^[5]。因此,该不相交组播树的构建方法可以由2种方式实现。第1种实施方式比较简单,就是网络中源节点集中式收集全网的节点状态信息,然后采用集中计算方式来构建,构建结束后,采用类似于源路由方式进行路由。第2种不需要先集中收集全网的节点状态信息,采用类似于分布式方式构建,在2.1节算法思路描述中已经介绍,只需要计算源节点到每个目的节点的不相交路径,这些不相交路径的计算可采用分布式来计算得出,具体的实现方法通过以下描述来实现。首先每个节点建立节点分裂操作后模式,这种分裂操作运算见文献[15],存储与各邻居节点之间的关系,源节点则通过广播方式依次去寻找到达目的节点的增广路径,当到达节点不是目的节点时,

保存源节点当前是求解第几条不相交路径标号,以及第1次到达该节点的邻居节点标号,然后再以广播形式进行转发。当到达节点是目的节点时,按照每个节点保存第1次到达该节点的邻居节点标号,反向到达源节点并保存反向路径,然后网络的源节点保存这些不相交路径。当网络中源节点收集了到达所有目的节点的不相交路径,通过2.2节描述的方法进行不相交组播树的构建,接来源节点根据计算的不相交组播树,再依次通知组播树中的各个节点,使组播树中的各个节点建立下一跳路由。当该业务结束时,源节点再通知组播树中各个节点删除该业务建立的下一跳路由信息。

2.4 复杂度分析

在计算复杂度方面,以 k 连通网络为例,在计算过程中,首先计算源节点 s 到各个目的节点的不相交路径,设目的节点的数量为 m ,在计算过程中,是以链路干扰为权重,通过最短路径方法进行不相交路径的求解,按邻接链表方式存储连接关系,则其复杂度为 $km |E| |V|$ (其中, $|E|$ 为网络中的链路数, $|V|$ 为网络中节点数),然后删除源节点,计算目的节点的不相交路径,计算复杂度为 $m(m-1)/2 |E| |V|$ (其中, m 为目的节点的数量)。计算各路径间的相交情况为 $(m(m-1)/2 + km)^2$,接下来在组播树构建过程中,依次添加每棵树中连接每个目的接收节点的路径,设最后生成不相交组播树数量为 $|T|$,每添加一次路径要判断与其他已建立的路径是否相交来找到增广不相交路径序列,其比较次数为 $(m(m-1)/2 + km)$,最多递归次数为 $|T| m$,其复杂度为 $|T| m(m(m-1)/2 + km)$ 。因此,通过累加后,总复杂度 $(|T| m + |E| |V|) \cdot (m(m-1)/2 + km) + (m(m-1)/2 + km)^2$,其最坏复杂度为 $O(m^4 + km^2(|T| m + |E| |V|))$ 。在空间复杂度方面,以矩阵存储为例,存储网络连接关系需要 $|V|^2$,存储源节点到其他目的节点的不相交路径以及目的节点之间的不相交路径需要最多存储空间为 $km^2 |V|$,存储最后不相交树的空间最多需要 $|T| m |V|$,因此,最后的空间复杂度为 $|V| \cdot (|V| + km^2 + km)$ 。

3 仿真实验及对比分析

3.1 算法合理性验证

为验证算法的正确性及有效性,采用随机的图模型作为无线网络模型,利用VC++建立生成无向网络图,具体做法就是在设定的一块平面区域内,设

定在区域放置的节点数量(这里以 60 节点为例),然后建立单位圆盘图 UDG(unit disk graph),即每个节点按设定的发射半径与其他节点建立网络拓扑图,根据文献[16]描述,当网络为 $2k$ 连通时,存在 k 棵链路不相交的生成树。由于目前网络通常以 2 棵不相交组播树作为容错要求,所以为能更好地验证算法性能,利用文献[17]的拓扑控制方法产生一个 3 连通拓扑,来验证不同算法寻找 2 棵以上的不相交组播树的性能。设需要建立不相交组播树的源节点为 2,随机产生 10 个目的接收节点,分别为 {9, 10, 59, 27, 7, 0, 1, 23, 32, 34}。利用目前主要采用的 NDT 方法^[12]与提出的算法计算不相交组播树,具体结果见图 5(a)~(c)。

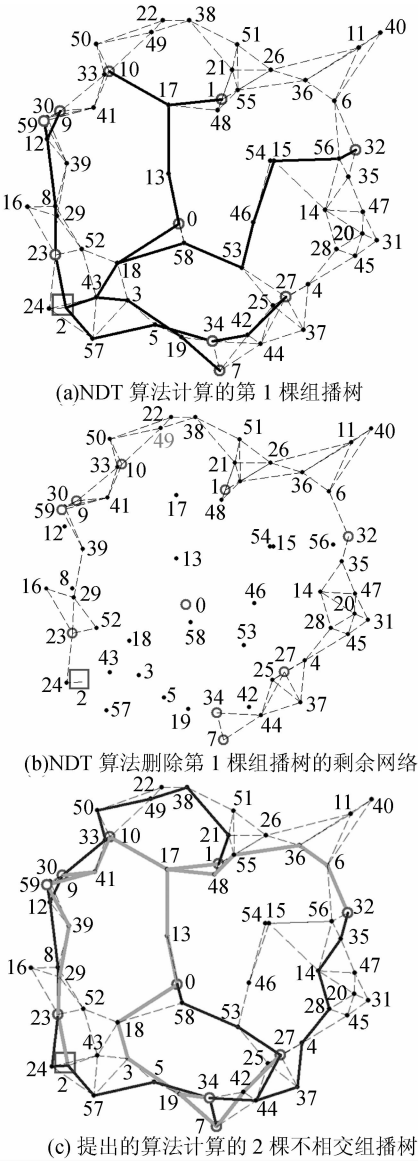


图5 提出的算法与 NDT 算法计算不相交组播树示例
Fig.5 Examples of calculating disjoint multicast trees based NDT algorithm and proposed algorithm

从图 5(a)和(b)可以看出,NDT 算法计算完第 1 棵组播树后(粗线表示计算的组播树),删除组播树的中间节点和链路,部分目的节点就成了孤立节点,过早结束了不相交组播树的计算。而对于图 5(c)的提出的算法计算结果可以看出,能很好地计算出 2 棵不相交组播树(分别用浅粗线、深粗线表示 2 棵不相交组播树),这也同时验证了算法设计的正确性和有效性。

3.2 算法性能对比分析

为验证提出算法的性能,同样按照上述网络模型进行设置,来验证不同算法求解最多的不相交组播树。在上述网络中,分别设置 8 组不同的目的节点数,即目的节点数从 3 到 10 依次递增,对于不同目的节点数,源、目的节点都随机产生,每组都仿真 100 次。分别比较这 2 种算法能找到的不相交组播树数,在相同数量组播树的情况下,对算法的平均路径跳数及平均链路数进行对比分析,这 2 个参数分别能反映出业务的端到端时延及组播树的资源利用效率。具体比较结果见表 1 及图 6 和 7。

表 1 不同接收节点数下成功求解相应数量组播树的次数百分比对比分析

Tab.1 Comparative analysis for the successful percentage of the corresponding number of multicast trees in different target nodes

		%							
不相交组播树个数		接收节点数							
		3	4	5	6	7	8	9	10
提出的算法	1	0	0	0	0	0	0	1	1
	2	97	100	100	100	100	100	99	99
	3	3	0	0	0	0	0	0	0
NDT 算法 ^[12]	1	10	39	46	34	60	68	51	71
	2	81	61	54	66	40	32	49	29
	3	9	0	0	0	0	0	0	0

从表 1 可以看出,在节点规模为 60 且 3 连通的情况下,提出的算法在求解 2 棵以上不相交组播树的成功比例远远高于 NDT 算法,并且随着目的节点数的增加,NDT 算法在当前网络规模下,寻找 2 棵不相交组播树的成功比例呈明显下降趋势,而提出的算法只是略微有所下降,通过计算,提出方法寻找 2 棵以上不相交组播树的成功比例平均提升 47.13%。

在图 6、7 中,提出方法计算的不相交组播树平均跳数及平均链路数整体上都略低于 NDT 算法,反映出提出算法不仅在计算多棵不相交组播树的效率上有明显优势,而且在平均跳数和平均链路数上也

优于 NDT 算法,对业务的端到端时延性能及资源的有效利用都有所提高。

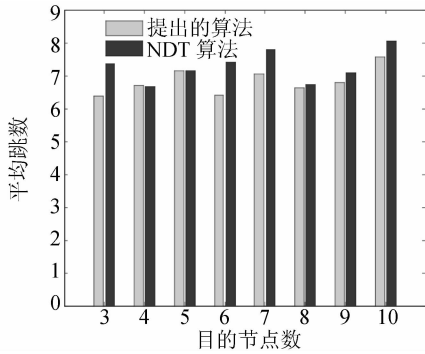


图 6 不相交组播树的平均跳数对比分析

Fig.6 Comparative analysis of average hops in multicast trees

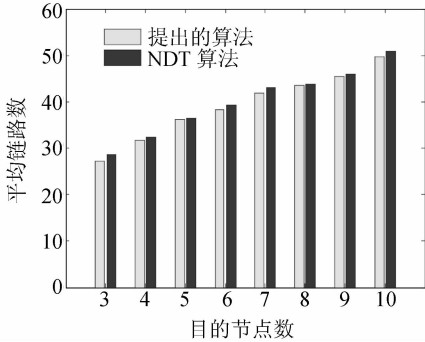


图 7 不相交组播树的平均链路数量对比分析

Fig.7 Comparative analysis of average links in multicast trees

为进一步验证不相交组播树构建方法的有效性,及不相交组播树的平均跳数和平均链路数对网络性能的影响,分别在不同目的节点数情况下,计算不相交组播树,再利用 NS2 网络仿真平台进行网络性能仿真对比分析。由于无线 Mesh 网络的标准不统一,存在基于 802.11、802.15、802.16 的 MAC 协议的平台,但仿真验证环境大部分还是采用 802.11 协议来验证网络路由和信道分配的可行性^[6-8]。为与目前的验证平台一致,同样建立一个与实现区域 $1\,000 \times 1\,000\text{ m}^2$ 的映射平面,节点位置在该区域随机分布,在区域内依次随机产生 60 个节点的网络,设定节点最大发射半径为 250 m,干扰距离为 500 m,利用文献[17]建立 3 连通网络拓扑。在接口及信道数方面由于设置过多的无线接口数和信道数不符合实际要求,这里设定每个节点的接口数为 3,可用信道数为 5(802.11a 中最多有 12 个非重叠的传输信道,因此选取 5 个信道是可行的),利用文献[18]的信道分配方法对建立的 3 连通拓扑进行信道分配。然后,设定目的节点数为 3~10(不同目的

节点数的节点都是随机产生),通过计算信道分配后的链路干扰。最后,分别利用 NDT 算法和提出方法计算 2 棵不相交组播树,源节点之间设置 UDP 传输的 CBR 数据流,数据包长为 1 024 Bytes,路由根据产生的组播树在网络中进行静态设置,MAC 协议为 802.11,发射速率为 2 Mbps。利用上述场景设置,分别验证在不同目的节点数时,不同方法求解的不相交组播树对网络性能的影响,具体见图 8~9。

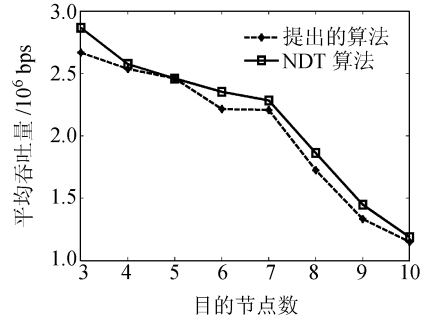


图 8 吞吐量对比分析

Fig.8 Comparative analysis of throughput

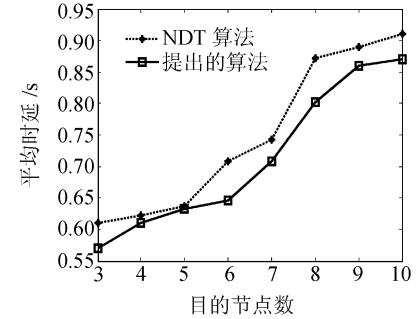


图 9 平均时延对比分析

Fig.9 Comparative analysis of average delay

从图 8、9 可以看出,在同一网络以及相同源、目的节点的情况下,提出的方法计算的 2 棵不相交组播树的网络性能都在一定程度上优于 NDT 算法,平均吞吐量及平均时延分别提升了 4.91% 和 5.16%。因此,提出的方法不仅在计算多棵不相交组播树数量的比例上有大幅提升,而且求得的不相交组播树对网络性能也有一定程度的提高,也进一步验证了方法的有效性。从算法构建思想和上述对比分析可知,提出的不相交组播树构建方法具有以下优势特征:1)利用源节点到目的节点以及目的节点之间的最多不相交路径进行组播树构建,最大限度地扩大了构建不相交组播树的路径数量,避免了类似于 NDT 算法中只是首先保证第 1 棵组播树的性能,使剩余网络中目的节点成为孤立节点,从而限制了后续不相交组播树的计算。2)先计算网络中源、目的节点的最多不相交路径,能够避免不相交组播树的路径长度,因为计算过程中要考虑到多条路径,不会

只顾及一条路径而忽略其他路径的计算。而 NDT 算法^[12]在计算第 1 棵组播树的源、目节点间的最短路径后网络中离目的节点最近的大部分节点包含在第 1 棵组播树中,从而增加了第 2 棵不相交组播树中路径的长度,究其原因是 NDT 算法中第 1 棵组播树占据了所有的最短路径,没有考虑 2 棵树路径长度的一定均衡性。

4 结 论

在描述构建拓扑容错组播树的重要性基础上,通过分析目前计算节点不相交组播树的局限性,建立了计算节点不相交组播树的优化模型,然后对构建思路进行了详细描述,提出了基于增广不相交路径的不相交组播树构建方法。通过模拟的网络环境进行算法验证和对比分析,虽然算法复杂度高于 NDT 算法,但提出的不相交组播树构建方法成功寻找 2 棵以上不相交组播树的比例远高于 NDT 算法^[12],而且求得的不相交组播树平均路数及平均链路数均低于 NDT 算法。除此之外,在网络性能对比方面,提出的方法求得不相交组播树的平均吞吐量及平均时延还分别提升了 4.91% 和 5.16%。从而表明了提出的方法的正确性和有效性,为构建具有拓扑容错的不相交组播树的方法提供了理论参考。未来的工作是进一步分析网络拓扑与不相交组播树数量的关系,进一步改进算法复杂度和算法性能。

参考文献:

- [1] Yuan J, Li Z, Yu Wei, et al. A cross-layer optimization framework for multihop multicast in wireless mesh networks[J]. IEEE Journal on Selected Areas in Communications, 2006, 24(11): 2092–2013.
- [2] Zeng G, Wang B, Ding Y, et al. Efficient multicast algorithms for multichannel wireless mesh networks[J]. IEEE Transactions on Parallel and Distributed Systems, 2010, 21(1): 86–98.
- [3] Nguyen H L, Nguyen U T. Channel assignment for multicast in multi-channel multi-radio wireless mesh networks[J]. Wireless Communications and Mobile Computing, 2009, 9(4): 557–571.
- [4] Nguyen U T, Xu J. Multicast routing in wireless mesh networks: Minimum cost trees or shortest path trees? [J]. IEEE Communications Magazine, 2007, 45(11): 72–77.
- [5] Chiu H S, Yeung K L. Maximizing multicast call acceptance rate in multi-channel multi-interface wireless mesh networks[J]. IEEE Transactions on Wireless Communications, 2010, 9(8): 2622–2631.
- [6] Liu T, Liao W. Multicast routing in multi-radio multi-channel wireless Mesh networks[J]. IEEE Transactions on Wireless Communications, 2010, 9(10): 3031–3039.
- [7] Chiu H S, Yeung K L, Lui K S. Bandwidth-guaranteed multicast in multi-channel multi-interface wireless mesh networks[C]//Proceedings of the 2009 IEEE International Conference on Communications. Dresden: IEEE, 2009: 4656–4660.
- [8] Zhao X, Guo J, Chou C T, et al. A high-throughput routing metric for reliable multicast in multi-rate wireless mesh networks[C]//Proceedings of IEEE International Conference on Computer Communications. Shanghai: IEEE, 2011: 2042–2050.
- [9] Lin K C J, Yang D N. Multicast with intra-flow network coding in multi-rate multi-channel wireless Mesh networks[J]. IEEE Transactions on Vehicular Technology, 2013(99): 1–14.
- [10] Wei W, Zakhor A. Multiple tree video multicast over wireless ad hoc networks[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2007, 17(1): 2–15.
- [11] Chow C O, Nishikawa H, Ishii H. Multiple tree video multicast over mobile ad hoc networks[C]//Proceedings of the IEEE Region 10 Technical Conference on Computers, Communications, Control and Power Engineering. Taipei: IEEE, 2007: 1–4.
- [12] Zhao X, Chou T C, Guo J, et al. Protecting multicast sessions in wireless mesh networks[C]//Proceedings of 31st IEEE Conference on Local Computer Networks. Tampa: IEEE, 2006: 467–474.
- [13] 赖虹建. 拟阵论[M]. 北京: 高等教育出版社, 2002.
- [14] Edmonds J. Minimum partition of a matroid into independent subsets[J]. Journal of Research of the National Bureau of Standards B: Mathematics and Mathematical Physics, 1965, 69B(1): 67–72.
- [15] Bao Xuecai, Dai Fusheng, Han Weizhan. Evaluation method of network invulnerability based on disjoint paths in topology[J]. Systems Engineering and Electronics, 2012, 34(1): 168–174. [包学才, 戴伏生, 韩卫占. 基于拓扑的不相交路径抗毁性指标与评估方法[J]. 系统工程与电子技术, 2012, 34(1): 168–174.]
- [16] Kriesell M. Local spanning trees in graphs and hypergraph decomposition with respect to edge connectivity [C]//Proceedings of the 6th Twente Workshop on Graphs and Combinatorial Optimization. Enschede: Elsevier, 1999(3): 110–113.
- [17] Li N, Hou J C. Localized fault-tolerant topology control in wireless ad hoc networks[J]. IEEE Transaction on Parallel and Distributed Systems, 2006, 17(4): 307–320.
- [18] Tang J, Xue G, Zhang W. Interference-aware topology control and qos routing in multi-channel wireless Mesh networks[C]//Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc). Urbana-Champaign: ACM, 2005: 68–77.