Ad Hoc网络中一种新的能量优化多播路由算法

邹力鹃 张其善 刘建伟

(北京航空航天大学 电子信息工程学院, 北京 100191)

摘 要: 探讨了 Ad Hoc网络中能量优化的多播路由问题. 在一次多播会话中节点功率不变的假设前提下,分析了 NJT(Node-Join-Tree)算法,针对 NJT算法所生成的多播树中可能存在冗余发送节点的问题,提出了一种新的能量优化多播路由算法 ——RMT(Refine-Multicast-Tree)算法,给出了 RMT算法的复杂性分析,并设计了 RMT算法的分布式实现方案. RMT算法在已有多播树中进行本地搜索,寻找是否存在冗余的发送节点,若存在冗余发送节点则将其从多播树发送节点集合中删除,重构多播树,从而达到减少多播树能量消耗的目的,生成能量更优的多播树. 仿真实验表明 RMT算法性能优于 NJT算法.

关 键 词: Ad Hoc网络; 能量优化; 多播; 本地搜索

中图分类号: TP 393. 04

文献标识码: A **文 章 编 号**: 1001-5965(2010)01-0065-06

New energy efficient multicast routing algorithm in Ad Hoc network

Zou Likun Zhang Qishan Liu Jianwei

(School of Electronics and Information Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract: The energy efficient multicasting was discussed. Based on the assumption that each node in the network has a fixed level of transmission power during a multicast session, node-join-tree (NJT) algorithm was analyzed. In order to solve the problem that there may be redundant transmitting nodes in multicast tree resulted from NJT algorithm, a new energy efficient algorithm — refine-multicast-tree (RMT) algorithm was proposed. The complexity analysis and distributed implementation of RMT algorithm were also presented. RMT algorithm locally searched existing multicast tree to find whether there are redundant transmitting nodes in tree. When redundant transmitting nodes exist in multicast tree, RMT algorithm deletes such nodes from transmitting node set and reconstruct multicasting tree. The total energy cost of multicast tree was decreased accordingly and thus a more energy efficient multicast tree was generated. Simulation result shows that RMT algorithm performances well compared with NJT algorithm.

Key words: Ad Hoc network; energy-efficient; multicast; local search

由于 Ad Hoc网络节点一般是用电池提供能量,限制了设备和网络的使用时间,对多播的应用造成了很大的影响. 在过去的十年间,电池容量仅仅增加了 2. 7倍,因此构造能量优化的多播树、设计相应多播路由协议得到了广泛的关注.

本文讨论的 Ad Hoc网络的能量优化问题是 MEM (Minimum Energy Multicast)问题[1]. MEM问题是指多播会话期间,多播需求所涉及节点的传输功率总消耗最小化,对应的度量指标是多播树总能耗 TEC (Total Energy Cost).

针对使用全向天线的 MEM问题,研究者提出了多种能量优化的多播路由算法. 文献[2-3]针对 MEB (Minimum Energy Broadcast)问题的 MLP模型进行扩展用于解决 MEM问题. 基于广播生成树的剪枝方法是根据已有生成树算法[4-7](如最小生成树 MST、最短路径生成树 SPT、广播

收稿日期: 2008-12-08

基金项目: 国家自然科学基金资助项目(606721202);国家 863计划资助项目(2006AA01Z422, 2009AA01Z418)

作者简介: 邹力鹃(1977-),女,湖南新化人,博士生, likun_zou@163. com.

增量功率 BIP)生成一个广播生成树,然后将没有孩子节点的非多播目的节点的树节点删除,从而获得相应的多播树. 由于发现能量最优的多播树问题实际上是最小 steiner树问题,因此文献[8]提出了基于最小 steiner树的 SPF算法和 MIPF算法. 本地搜索算法则使用本地搜索策略反复改进初始方案 (如由一个生成树算法得到的解决方案),从而得到最终的优化方案. 分布式算法 S-REMiT[9]和 MDDP算法[10]是本地搜索算法的典型代表.

上述研究工作均假设在一次多播会话过程中,节点具有可以根据与接收节点的距离和环境背景调整传输功率的能力;因此,这些研究工作的重点是在一次多播会话过程中,如何利用节点功率可调的特性改变节点的传输功率以达到提高能量效率的目的. 然而在实际网络系统中,节点的传输级别很可能在一次多播需求中不会发生改变,而在不同的多播会话阶段其传输功率可以不同,这与已有研究工作的假设前提不同,因此原有的算法和路由协议不再适用. 针对多播会话阶段节点传输功率不变的情况,文献[11]首次进行了相应的能量优化的多播协议研究,提出了 NJT (Nodes-Join-Tree)算法. 由于根据 NJT算法生成的多播树中可能存 余的发送节点,本文提出了一个新的能量优化算法——RMT(Refine-Multi-cast-Tree)算法. RMT算法采取本地搜索策略,在已有多播树中寻找冗余发送节点,当多播树中存在冗余发送节点时,RMT算法将找到的冗余发送节点从发送节点集合中删除,重构多播树以减少多播树能量消耗.

1 系统模型

网络模型为有向图 $G=(V, E)$, V为有限顶点集合 (顶点对应 Ad Hoc网络中的节点)且 $|V|=n$, E为有向边的集合. 对于 V中的顶点 v_i,其传输功率为 $p(v)$,对于任意的 2个顶点 v_1和 v_2,若 v_2在 v_1的传输范围之内,则 $(v_1, v_2)\in E$,即存在一个从 v_1到 v_2的有向边,称 v_2是 v_1的邻居节点. 系统模型假设:节点使用全向天线,且在一次多播会话阶段是静止的;节点知道它的邻居以及自身的能量消耗情况;网络连接可以是不对称的.

对于多播需求 (s, D), s代表多播源节点, D是目的节点集合. 假设 T_s是一个根节点为 s的多播树,记为 $T_s(V_T, E_T)$,其中 $V_T\subseteq V$是树中节点的集合, $E_T\subseteq E$是树连接的集合. 可以将 T_s中的节点分类:只发送数据的节点 (源节点)、只需接收数据的节点 (叶子节点)、既需要接收数据又需要发送 /转发数据的节点 (不是源节点的非叶节点). 假设接收节点不消耗能量,那么多播树 T_s的全部能量消耗 TEC(T_s)可以计算为 $\sum_{v\in \text{Non-Leaf}(T_s)} p(v)$,其中 Non-Leaf($T_s$)表示树中非叶子节点的集合,即多播树中发送节点的集合. MEM问题是寻找一个多播树结构,使得该多播树的总能耗最小.

2 NJT算法

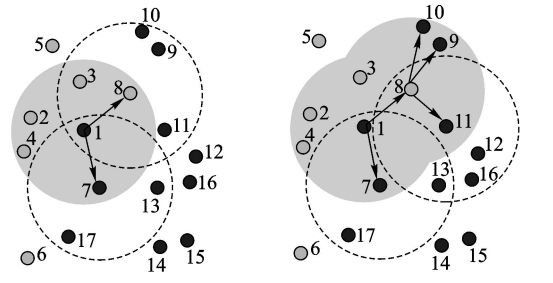
NJT算法的目的是创建总能耗 TEC(T_s)趋向最小的多播树 T_s. NJT算法中, C代表多播树中非叶子节点的集合,即发送节点集合,初始为 $\{s\}$, N为 C中节点邻居节点的集合, U为集合 C中节点所不能覆盖的节点集合,即 C, N外的节点.

NJT算法起始于 s初始时, C只包含 s,且 $U=D$,将 s的所有邻居节点从 U中删除,添加到 N中,接着从 N中选取一个节点加入 C,并将其邻居节点从 U中删除,加入到 N中,如此反复,直到 U为空为止. 此时意味着 D中所有节点已被 C覆盖, C中所得就是最终多播树中发送节点集合.

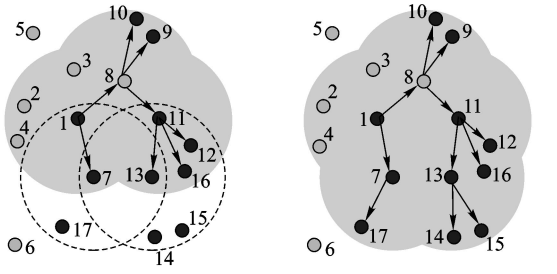
假设 neighbor(v_i)是 v_i的邻居节点集合,即 neighbor(v_i) $=\{v_j \mid (v_j, v_i)\in E\}$并且有 $v_j\neq v_i$,为了使 C中每增加一个新节点时,以尽量少的能量消耗覆盖尽可能多的目的节点,对 N中节点 v_i使用评价函数 $f(v_i)=|\text{neighbor}(v_i)\cap U|/p(v_i)$进行评估. $f(v_i)$表示候选节点在每个能量单位所能覆盖的目的节点的个数, $f(v_i)$的数值越大,说明节点能量效率越高. 每一次选取具有最大数值的节点加入到 C中,以控制多播树的总能量消耗尽可能的小.

NJT算法的例子如图 1所示,多播需求为 (1, {7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17}) (图 1中黑色节点),各节点的传输功率相同. 根据 NJT算法,初始设置为 $C=\{1\}$, $N=\{2, 3, 4, 7, 8\}$, $U=\{9, 10, 11, 12, 13, 14, 15, 16, 17\}$,图 1a中灰色区域代表源节点的传输范围. 首先加入第 1个新节点,如图 1a所示,比较节点 7和节点 8的能量效率,由于 $f(8)>f(7)$,因此 $C=\{1, 8\}$, $N=\{2, 3, 4, 7, 8, 9, 10, 11\}$, $U=\{12, 13, 14, 15, 16, 17\}$. 然后加入第 2个新节点,比较 7和 11的能量效率 (见图 1b), $f(11)>f(7)$,此时 $C=\{1, 8, 11\}$, $N=\{2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 16\}$, $U=\{14, 15, 17\}$. 接着,加入第 3个新节点,如图 1c所示,由于 $f(13)>f(7)$,因此 C变化为 $\{1, 8, 11, 13\}$, $N=\{2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$, $U=$

{17}.最后,节点 7的能量效率最大,所以 $C = \{1, 8, 11, 13, 7\}$, $N = \{2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}$, $U = \{\}$,算法终止,结果多播树见图 1d.该多播树的能量消耗为 $p(1) + p(8) + p(11) + p(13) + p(7)$.



a 比较节点 7,8的能量效率　b 比较节点 7,11的能量效率

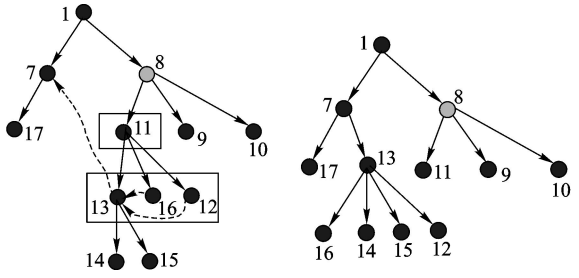


c 比较节点 7,13的能量效率　d 结果多播树

图 1　NJT算法示例

3　RMT算法

在 NJT算法生成的多播树中,可能存在多余的发送节点.如图 2a所示,节点 11的孩子节点为 {12, 13, 16},其中 12与 16可以被 13覆盖,而 13可以被节点 7覆盖;因此在多播树中,如果节点 11不再作为发送节点仅作为接收节点,并改变其孩子节点在树中的父节点,如图 2b所示,多播树的总能耗可以降低 $p(11)$.基于这个事实,提出了一个本地搜索算法——RMT算法.



a 节点 11是冗余的发送节点　b 将节点 11变非发送法节点

图 2　转换冗余发送节点的身份

假设根据 NJT算法得到的生成树 T_s以及树中发送节点集合 C,对于多播树 T_s中的节点 v, v的孩子节点记为 children(v, T_s),而所有经过 v可达的树节点记为 descendant(v, T_s),显然 children(v, T_s) ⊆ descendant(v, T_s).当 children(v, T_s)中的节点满足条件:

1) 对于任意 $x \in$ children(v, T_s), x能够被 $C \setminus \{v\}$所覆盖;

2) 至少存在一个 $x \in$ children(v, T_s)且 $x \in C$, x能够被 $(C - \{v\}) \setminus$ descendant(v, T_s)覆盖.

则节点 v是冗余的发送节点,可以从结果多播树中的发送节点集合中删除,同时 children(v, T_s)中的节点也需要改变父亲节点.判断 T_s中的节点 v是否为冗余发送节点的过程记为 Is_redundant(v, T_s, C),其伪码如下:

```
C1 = (C - {v}) \descendant(v, Ts);
C2 = C \{v};
scoverflag = false;
wcoverflag = true;
    For each node x in children(v, Ts)
        If(x is forwarder and can be covered by C1)
            scoverflag = true;
        End-if
        If(x cannot be covered by C2)
            wcoverflag = false;
        End-if
    End-for
    Return(scoverflag && wcoverflag)
```

RMT算法以 NJT算法生成的多播树 T_s及发送节点集合 C为输入,对 $C \setminus \{s\}$中每一个节点 v进行判断:v是否为多余发送节点,如果 v是冗余发送节点,记录该节点的发送功率,将具有最大功率的冗余发送节点删除;反复执行寻找冗余发送节点、删除节点操作,直到没有冗余的发送节点为止;最后根据 C重构多播树. RMT算法的流程如下:

```
Input:  Ts and C which are resulted from NJT algorithm
Output: refined Ts
changeflag = true;
While (changeflag)
d_power = 0;  n = -1;
    For each node v in C \{s}
        If (Is_redundant(v, Ts, C) and d_power < p(v))
            n = v;  d_power = p(v);
        End-if
    End-for
    If (n! = -1)
        Eliminate(n, Ts);
```

```
    Else
        Changeflag = false;
    End-if
End-while
Reconstruct tree according to refined C
```

RMT算法的时间复杂度是 $O(Kn^3)$,其中 K 是 While循环执行的次数,表示删除冗余发送节点的次数.由于 C中至多有 n个节点,因此 For循环最多执行 n次,而 Is_redundant过程的复杂性为 $O(n^2)$,由于 $K \ll n$,所以 RMT算法的时间复杂度为 $O(n^3)$.

4 分布式 RMT算法实现

RMT算法也可以分布式方式在网络节点上运行.除了路由表 R_v和网络连接的邻居节点表 N_v,节点 v还需要维护的数据结构包括:表明树节点状态的标志(forwarder/receiver),一个节点可以既是 receiver也是 forwarder;从源节点 s经过树连接到达节点 v的直接路由 π_v;记录节点 v在树中的邻居节点信息的树邻居节点表 B_v.为了方便算法的实现,树邻居节点表 B_v不仅记录节点 v的父节点和孩子节点,还记录其它树中发送节点 u的信息.B_v中的条目包括(节点 id、关系标识 child/father/scover/wcover)信息.

例如树节点 x的状态为 forwarder,x不是 v在树中的父节点且 v在 x的传输范围内,若 x不是 v的后代节点,则 B_v中记录 $(x,$ scover)的条目,若 x是 v的后代节点,则 B_v中记录 $(x,$wcover)的条目.scover和 wcover表示不同的覆盖程度.分布式 RMT算法中使用的消息类型如表 1所示.

表 1 分布式 RMT算法中的消息类型

消 息	含 义
R_REQ(i, s, π)	由节点 i转发的源节点 s寻找冗余发送节点的请求,π记录消息传递的路由.
R_REP(i, s, c, p_c, π)	节点 i收到 R_REQ后的回复信息,c是在节点 i的后代节点中功率最大的冗余发送节点,π记录到达 c的路由.
D_REQ(i, s, v, π)	由节点 i转发的源节点 s将节点 v从多播树中发送节点中删除的请求.
D_REP($i, s, v,$ flag)	节点 i收到 D_REQ后的回复,flag取值为 ture或 false.
C_REQ($i, j,$ type)	节点 i向节点 j发送的变更父节点的请求,type取值为 check或 opertion.
C_REP($j, i, s_{\text{flag}}, w_{\text{flag}},$ type)	节点 j向节点 i发送的关于变更父节点的回复.

多播树的优化搜索按轮次进行,每一轮的优化搜索由多播树的源节点 s发起,R_REQ消息在多播树内沿树连接传播,s根据多播树节点回复,选择要删除的冗余发送节点 v,s发出 D_REQ后,完成节点 v在多播树中删除操作,本轮优化完成.源节点 s作为多播树的根节点,在收到 R_Reply后无法找到可删除节点时,终止优化操作.

1) 寻找冗余发送节点:s发送 R_REQ(s, s, π_s)消息,R_REQ消息在多播树内沿树连接传播.当节点 j收到来自树中非父节点 i的 R_REQ(i, s, π_i),j在 B_j中增加关于 i的条目后丢弃消息.当节点 j收到来自树中父节点 i的 R_REQ(i, s, π_i),j首先更新源路由,如果 j是 forwarder,则更新消息为 R_REQ(j, s, π_j)后转发,否则回复 R_REP($j, s, \Phi, 0, \Phi$)给其父节点 i 节点 i只处理来自树中孩子节点 j的 R_REP(j, s, c, p_c, π_c),根据孩子节点的 R_REP,获取功率最大的删除候选节点 $c_{\max}$及其功率 $p_{c_{\max}}$和源路由 $\pi_{c_{\max}}$.若 $p_{c_{\max}} > p_i$,则 i释放所有接收的 R_REP,回复父节点 R_REP($i, s, c_{\max}, p_{c_{\max}}, \pi_{c_{\max}}$);若 $p_{c_{\max}} < p_i$,i向孩子节点发出 C_REQ($i, \Phi,$ check),i的孩子节点 j在收到 C_REQ后检查 B_j后回复 C_REP($j, i, s_{\text{flag}}, w_{\text{flag}},$ check)(若 B_j中存在 scover的条目,$s_{\text{flag}}, w_{\text{flag}}$均为 true;没有 scover的条目,但有 wcover的条目,则 $s_{\text{flag}}, w_{\text{flag}}$分别为 false, true;否则均为 false).根据孩子节点的 C_REP,i判断自己是否为冗余发送节点.若 i是冗余的节点,i释放所有接收的 R_REP,回复父节点 R_REP(i, s, i, p_i, π_i),否则向父节点回复 R_REP($i, s, c_{\max}, p_{c_{\max}}, \pi_{c_{\max}}$).当源节点 s收到所有孩子节点的 R_REP后,选择功率最大的冗余节点,记为 $v_{\max}$.

2) 删除节点、重构多播树:s执行删除操作,即发送 D_REQ($s, s, v_{\max}, \pi_{v_{\max}}$)消息.消息 D_REQ沿 $\pi_{v_{\max}}$在多播树中传播,即节点 j收到来自节点 $i \in \pi_{v_{\max}}$转发的消息 D_REQ,当且仅当 j是 i在 $\pi_{v_{\max}}$中的下一跳节点才转发消息.当 x收到来自节点 $v_{\max}$的 D_REQ时,说明 x是 $v_{\max}$的孩子节点,x在 B_x选择标识为 scover的节点 y(没有 scover标识时选择 wcover标识)发送 C_REQ($x, y,$ operation)请求,y收到 C_REQ($x, y,$ operation)请求后,在 B_y增加 $(x,$ child)条目,回复 C_REP,x收到 C_REP,删除 B_x中 $v_{\max}$的条目,将 y的状态改为 father.x回复 D_REP给 $v_{\max}$.节点 $v_{\max}$收到所有孩子节点的 D_REP后修改 $B_{v_{\max}}$,将所有 child标识条目删除,除去 forwarder状态后,$v_{\max}$向树中父节点回复消息为 D_REP($v_{\max}, v_{\max}, s,$ true).D_REP沿多播树连接单播,直到源节点收到 D_REP后,说明一次优化操作完成.

当源节点在发出 R_REQ后所收到的 R_REP

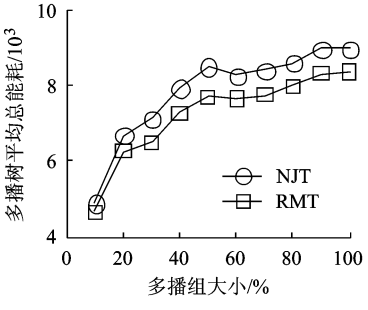
表明没有冗余发送节点时,RMT算法结束.

5 仿真和分析

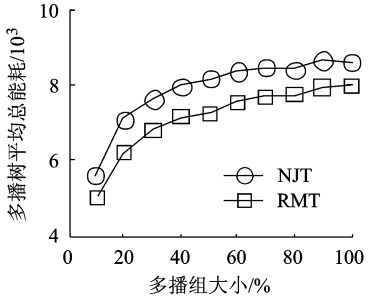
为了评价 RMT算法的性能,使用 Java实现 NJT,RMT算法,并对 2个算法的性能进行了比较.网络规模分别为 N(50,100)节点随机产生分布在一个 100 ×100的 2D矩形区域中.假设 R的单位是区域对角线距离,即当 R =1时,一个节点的传输距离可以覆盖整个区域.仿真中采用的功率模型为 $P = r^2$,其中 P是传输功率, r是信号可达的半径.每个节点的传输半径服从均值和方差均为 $R = 1/\sqrt{N}$的正态分布.任何导致不连通的拓扑结构都将被丢弃,以保证仿真中网络是连通的.

仿真模拟的模拟次数为 100次,每一次仿真中,给定 N,M和 R,随机在区域生成 N个节点,选择的 M%的节点作为多播组以及节点的传输半径.算法性能的度量包括:①AT(Average TEC),多播树的平均总能耗;②NT(Normalized TEC),多播树的标准总能耗 NT是 TEC_{alg}和 TEC_{best}的比值,其中 $alg \in A$, $A = \{NJT, RMT\}$, $TEC_{best} = \min\{TEC_{alg} \mid alg \in A\}$;③发送节点数量.

图 3a和图 3b分别表示 N = 50和 N = 100时,算法构造的多播树总能耗 TEC的平均值 AT与多播组大小之间的关系.当多播组增大时,AT也呈现增加的趋势,显然 RMT算法构造的多播树的 AT小于 NJT算法构造的多播树的 AT.



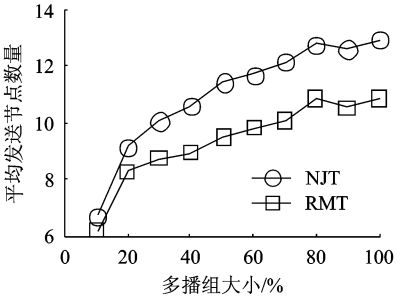
a　50个节点



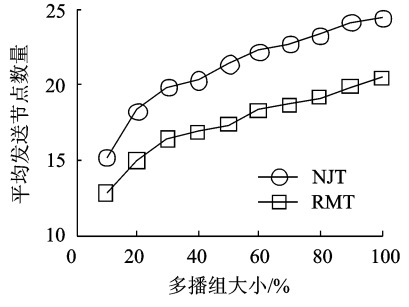
b　100个节点

图 3　多播树总能耗平均值 vs多播组大小

图 4a和图 4b分别表示 N = 50和 N = 100时,不同算法构造的多播树中发送节点数量与多播组大小之间关系.由图 4可知 NJT算法构造的多播树的所需的发送节点数量多于 RMT算法构造的多播树所需发送节点数量.



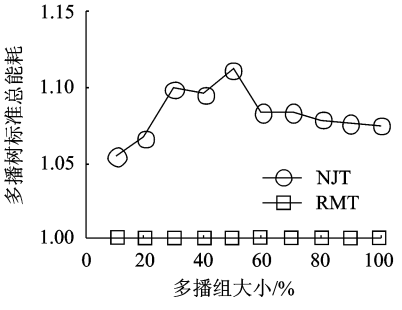
a　50个节点



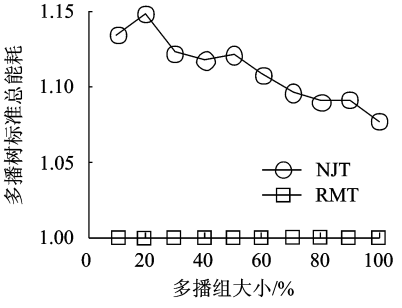
b　100个节点

图 4　多播树中发送节点数量 vs多播组大小

图 5a和图 5b分别表示 N = 50和 N = 100时,不同算法构造的多播树的 NT之间的比较.由图 5可知,NT与多播组大小没有明显的关联,与图 3、图 4结论一样,RMT算法的 NT小于由 NJT算法构造的多播树的 NT.



a　50个节点



b　100个节点

图 5　NT vs多播组大小

6 结束语

在多播会话阶段节点传输功率不变的前提下,本文提出了一种新的能量优化多播路由算法——RMT算法构造能量优化的多播树. RMT算法是对已有的多播树进行本地优化搜索,将冗余的发送节点转换为接收节点、重构多播树,从而达到降低多播树总能耗的目的. 文中不仅给出了集中式算法的流程,理论分析了算法的最坏时间复杂度,而且给出了分布式实现的方案. 仿真结果显示,RMT算法优于 NJT算法,可以极大降低多播树的总能耗.

参考文献 (References)

[1] Guo S, Yang O W W. Energy-aware multicasting in wireless ad hoc networks: a survey and discussion[J]. Computer Communications, 2007, 30(9): 2129 - 2148

[2] Guo S, Yang O W. Minimum-energy multicast routing in static wireless ad hoc networks[C]// Proceeding of IEEE 60th Vehicular Technology Conference. Los Angeles: IEEE Press, 2004: 3989 - 3993

[3] Di Y, Bauer J, Haugland D. Minimum-energy broadcast and multicast in wireless networks: an integer programming approach and improved heuristic algorithms[J]. Ad Hoc Networks, 2008, 6(5): 696 - 717

[4] Wieselthier J E, Nguyen G D, Ephremides A. Energy-efficient broadcast and multicast trees in wireless networks[J]. Mobile Networks and Applications, 2002, 7(6): 481 - 492

[5] Kang I, Poovendran R. A novel power-efficient broadcast routing algorithm exploiting broadcast efficiency[C]// Proceeding of 58th IEEE Vehicular Technology Conference. Orlando: IEEE Press, 2003: 2926 - 2930

[6] Wan P J, Calinescu G, Li X Y, et al. Minimum-energy broadcast routing in static ad hoc wireless networks[C]// Proceeding of 20th Annual Joint Conference of the IEEE Computer and Communications Societies. Anchorage: IEEE Press, 2001: 1162 - 1171

[7] Wieselthier J E, Nguyen G D, Ephremides A. On the construction of energy-efficient broadcast and multicast trees in wireless networks[C]// Proceeding of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies. Tel Aviv: IEEE Press, 2000: 585 - 594

[8] Wan P, Calinescu G, Yi C. Minimum-power multicast routing in static ad hoc wireless networks[J]. IEEE/ACM Transactions on Networking, 2004, 12(3): 507 - 514

[9] Wang B, Gupta S K S. S-REMiT: a distributed algorithm for source-based energy efficient multicasting in wireless Ad Hoc networks[C]// Proceeding of IEEE Global Telecommunications Conference. San Francisco: IEEE Press, 2003: 3519 - 3524

[10] Guo S, Yang O. A dynamic multicast tree reconstruction algorithm for minimum-energy multicasting in wireless ad hoc networks[C]// Proceeding of the 23rd IEEE International Performance, Computing, and Communications Conference. Phoenix: IEEE Press, 2004: 637 - 642

[11] Li D, Liu Q, Hu X, et al. Energy efficient multicast routing in ad hoc wireless networks[J]. Computer Communications, 2007, 30(18): 3746 - 375

(编 辑:娄 嘉)

(上接第 64页)

4 结 论

HBDS采用了一种具有综合性和开放式的新的软硬件架构,软件架构按功能划分为 5个层次,硬件架构中充分考虑了数字裁剪加工和飞行仿真与试验等实际应用要求,采用面向对象的整合方法,包含了自定义开发、多 Agent基线比选等开放性设计功能,使用者可以根据不同的设计方案和任务要求对系统中的模块进行自主设计和开发,一定程度上增加了系统的开放性并提高了设计者的工作效率.

参考文献 (References)

[1] Said M A. A review on the recent progress in materials selection development and characterization for ultra long duration balloon (ULDB) missions[J]. Advances in Space Research, 2002, 30(5): 1173 - 1182

[2] Kreith F A, Kreider J. Numerical prediction of the performance of high altitude balloons[R]. NCAR-IN/STR-65, 1974

[3] Runnels S R, Smith M S, Fairbrother D A. High-altitude balloon thermal trajectory analysis database system[R]. AIAA-2003-6821, 2003

[4] Carlson L A, Horn W J. A new thermal and trajectory model for high-altitude balloons[R]. AIAA-81-1926, 1981

[5] 黄无量,徐春娴,吴枚,等. 高空科学气球的应力分析与球形设计[J]. 空间科学学报, 1986, 6(1): 59 - 62
Huang Wuliang, Xu Chunxian, Wu Mei, et al. On the stress analysis and the shape design of the scientific balloon at high altitude[J]. Chinese Journal of Space Science, 1986, 6(1): 59 - 62 (in Chinese)

[6] Izutsu N, Yajima N, Ohta S, et al. Flight demonstration of a superpressure balloon by three-dimensional gore design[J]. Advances in Space Research, 2002, 30(5): 1221 - 1226

(编 辑:李 晶)