

DOI: 10.7652/xjtuxb201412008

自适应动态功率控制的机会网络节能高效路由算法

李季碧, 李宾, 任智, 陈前斌
(重庆邮电大学移动通信技术重庆市重点实验室, 400065, 重庆)

摘要: 针对机会网络中基于跨层设计的能量高效路由算法(ERBC)存在的未考虑节点运动、部分数据消息传输时能耗偏大、矢量消息交换过程有冗余控制开销的问题,提出一种自适应动态功率控制的节能路由算法(ERAPC)加以解决。ERAPC 算法通过拓展确认字符(ACK)帧的使用改进了基于接收信号强度指示值(RSSI)的节点测距机制,将功率控制的范围从部分数据消息扩展到全部,以减少节点能耗;通过等待收发节点尽可能靠近后才传送数据,进一步减小节点能量消耗;通过提出一种更简捷的矢量消息交换新机制,减少网络控制开销。仿真结果表明,与 ERBC 算法相比,ERAPC 算法的比特能耗至少降低了 27.27%,控制开销则减小了 11.87%以上。

关键词: 机会网络;路由算法;节能;功率控制;开销

中图分类号: TP393.04 **文献标志码:** A **文章编号:** 0253-987X(2014)12-0049-08

An Efficient Energy-Saving Routing Algorithm for Opportunistic Networks with Dynamically Adaptive Power Control

LI Jibi, LI Bin, REN Zhi, CHEN Qianbin
(Key Laboratory of Mobile Communications Technology of Chongqing, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: An efficient energy-saving routing algorithm with dynamically adaptive power control (ERAPC) is proposed to address the problems that there exist large energy consumption for transmitting partial data packets, no consideration of node's mobility and high control overhead in vector exchange mechanism lying in the energy-efficient routing algorithm based on cross-layer design (ERBC). The ERAPC extends the usage of acknowledgement (ACK) frames to improve the RSSI-based ranging mechanism, to enlarge the range of power control to all data messages and to reduce nodal energy consumption. Until the sender and the receiver move as close as possible, the data messages are sent out to lower the transmit power further; and an efficient new mechanism of exchanging vectors is presented in ERAPC to decrease the control overhead. Simulation results and comparison with the ERBC algorithm show that the energy consumption of each bit in the ERAPC is reduced by at least 27.27% and the control overhead is reduced by at least 11.87%, respectively.

Keywords: opportunistic networks; routing algorithm; energy-saving; power control; overhead

路由技术是机会网络体系结构中的重要组成部分,对数据传输和网络整体性能有关键性的影响作

用,因此目前受到较多关注^[1-4]。节能路由算法的研究重点旨在减少数据传输过程中节点的能量消耗。机会网络中的节点多依靠电池供电,携带的能量有限,而在复杂的无线环境下电池的更换往往较为困难,这使得人们开始重视对机会网络节能路由算法^[5]的研究。

目前,根据节能策略的不同,可将机会网络节能路由算法分为间接节能和直接节能路由算法2大类。典型的间接节能的路由算法包括文献[6]提出的短期到达率估计算法(short-term arrival rate estimation, STAR)和文献[7]提出的 n -Epidemic 路由算法等,该类路由算法通过减少控制消息或数据发送次数来达到节能效果。直接节能的典型路由算法又可分为2类:一类如文献[8]提出的基于接触时间的休眠机制(sleep scheme based on contact time, SSCT),该类算法旨在通过让节点适当地休眠来达到减少能耗的目的;另一类采用功率控制的思路,如文献[9]提出的基于跨层设计的能量高效路由算法(energy-efficient routing algorithm based on cross-layer design, ERBC),该类算法使用接收信号强度指示值(received signal strength indication, RSSI)等技术调整节点发射功率,通过减少发送功率达到节能的目的。ERBC算法是一种代表性的采用功率控制、具有直接节能效果的路由算法,但在研究中发现ERBC算法在原理方面仍然存在以下不足:①在传输数据时只进行一次功率调整,在节点移动的情况下存在功率控制不准确的问题,影响节能效果和传送成功率;②在2个节点相互传送数据时只有一个节点能进行功率控制,导致部分数据消息传输能耗偏大;③在节点测距时忽略了媒体接入控制(media access control, MAC)层确认(acknowledgement, ACK)帧的使用,使得用于测距的消息种类和数量有限,导致测距开销偏大;④在矢量交换过程中存在冗余的矢量消息。本文为解决目前机会网络中具有代表性的节能路由算法ERBC所存在的上述4个问题,提出一种新的自适应动态功率控制的节能高效路由算法——ERAPC。

1 模型及相关技术描述

1.1 机会网络模型

机会网络的数学模型为 $G=(V, E)$, 其中 V 表示所有节点的集合, $V=\{v_1, v_2, \dots, v_n\}$ (n 表示网络节点数, v_n 代表第 n 个网络节点), E 表示所有链路的集合且 $E=\emptyset \cup \{e_1, e_2, \dots, e_k\}$ (e_k 为网络中第 k

条链路)。

当机会网络中2个节点相互进入对方的通信范围时,它们之间会生成1条链路,用 $\{e_i, (t_{si}, t_{ei})\}$ 表示该链路,其中 $1 \leq i \leq n(n-1)$, t_{si} 、 t_{ei} 分别表示该链路的生成时间和终止时间,且 $t_{si} < t_{ei}$, 则机会网络路由的数学模型为:寻找至少一个在逻辑上有序相连的链路组合 $\Sigma\{e_i, (t_{si}, t_{ei})\}$, 使该链路组合的首尾节点分别是数据分组的源节点和目的节点,且相邻2条链路中前一条链路的生成时间 t_s 必须小于后一条链路的终止时间 t_e , 即 $t_{si} < t_{e(i+1)}$ ^[10]。

根据上述机会网络路由模型,数据消息从源节点 S 传输至目的节点 D 的过程示意图如图1所示。

s : 源节点; D : 目标; A, B, C : 中继

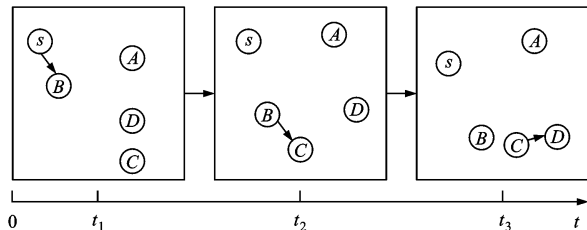


图1 机会网络数据消息传输示意图

在 t_1 时刻存在节点 S 到 B 的通信链路,记为 $\{e_1, (t_{s1}, t_{e1})\}$, 则 S 将携带的数据消息传输给节点 B ; 在 t_2 时刻存在节点 B 到 C 的通信链路,记为 $\{e_2, (t_{s2}, t_{e2})\}$, 则 B 将携带的数据消息传输给节点 C ; 在 t_3 时刻存在节点 C 到 D 的通信链路,记为 $\{e_3, (t_{s3}, t_{e3})\}$, 则 C 将携带的数据消息传输给节点 D 。可见,上述情形满足相邻2条链路中前一条链路的生成时间均小于后一条链路的终止时间。

1.2 能量模型

本文采用的节点能量消耗模型如下^[11]

$$E_{\text{trans}} = \begin{cases} kE_{\text{elec}} + k\epsilon_{\text{fs}}d^2, & d < d_0 \\ kE_{\text{elec}} + k\epsilon_{\text{amp}}d^4, & d \geq d_0 \end{cases} \quad (1)$$

$$E_{\text{RX}} = kE_{\text{elec}} \quad (2)$$

式中: k 为比特数; d 为节点间距离; E_{elec} 为发送和接收单位比特信息的损耗常量; ϵ_{fs} 、 ϵ_{amp} 分别是这2种模型中功率放大电路的功耗系数。式(1)为节点发送信息能耗,式(2)为节点接收信息能耗。

1.3 节点发射功率等级

将节点的固定发射功率 P 设置为最高等级,把发射功率从0到 P 平均分为 n 等份,每一个发射功率等级对应一个相应的通信范围。假设每个节点的通信范围是一个半径为 R 的圆形区域,将该圆形区域以半径 R 为基准分为 n 份,则发射功率等级 y 与距离 d 的对应关系为

$$y = \left\lceil \frac{nd}{R} \right\rceil \quad (3)$$

1.4 RSSI 测距模型

本文采用地面反射双线模型,用 P_S 表示节点 S 的发射信号强度,用 P_R 表示节点 R 的接收信号强度,则 2 个节点之间的距离为

$$D_{SR} = \left(c \frac{P_S}{P_R} \right)^{1/4} \quad (4)$$

式中: c 为传播影响因子,它是由天线增益和具体的无线信号传播环境等因素决定的一个常数。

2 ERAPC 算法

本文 ERAPC 算法在功率控制过程中引入 ACK 帧的使用,节点在发送数据消息时通过 RSSI 测距多次调整节点的发射功率;自节点利用 MAC 层的 ACK 帧的保留位捎带“有无其他邻居”信息给它节点,有条件地等待 2 个节点靠近后再进行消息传递;提出在 2 个节点相遇初期只进行一次摘要矢量(summary vector, SV)和请求矢量(request vector, RV)交互的矢量交换机制。

ERAPC 算法的主要操作步骤如下。

设节点 A 、 B 相互进入对方的通信范围并且都有数据要发送给对方;并且假设 B 收到 A 周期性广播的探寻(Hello)消息,记此时为 $T=0$ 时刻。 A 、 B 根据 ERAPC 算法进行操作的步骤如下。

(1)节点 A 以固定功率周期性广播 Hello 消息(其他节点也进行此操作)。

(2)节点 B 收到 A 广播的 Hello 消息,记该时刻为 t_{11} ,节点 B 以固定功率向 A 单播 SV 消息,并且用式(4)根据接收信号强度估算出此时 A 、 B 之间的距离

$$d_{AB} = \left(c \frac{P}{P_1} \right)^{1/4} \quad (5)$$

式中: P 是已知的节点的固定发射功率; P_1 是接收的 Hello 消息的信号强度值。

(3)节点 A 收到 SV,记此时为时刻 t_1 ,节点 A 用式(4)根据接收信号强度估算出此时 A 、 B 之间的距离

$$d_{AB1} = \left(c \frac{P}{P_2} \right)^{1/4} \quad (6)$$

式中: P_2 是接收的 SV 消息的信号强度值。

节点 A 根据计算出来的 d_{AB1} ,计算发送 ACK 帧的传输距离。为了保证对方节点 B 一定能够收到节点 A 发送的 ACK 确认帧,假定 B 以最大速度与本节点 A 作相反方向的运动的极端情况,则此时

估算出来的节点 A 发送 ACK 确认帧的传输距离为

$$d_{ACK} = d_{AB1} + T_{\max}(V_{\max} + V_{\min}) + \frac{(L_{SV} + 2L_{ACK})V_{\max} + L_{ACK}V_{\min}}{B} \quad (7)$$

式中: $T_{\max}$ 为节点的最大处理时延; $V_{\max}$ 为网络中所有节点的最大移动速度; $V_{\min}$ 为节点自身的移动速度; L_{ACK} 为 ACK 确认帧的长度, L_{SV} 为 SV 消息的长度; B 为带宽;此处忽略传播时延。

为了告知对方节点 B 是否可以等待两节点靠近后再进行数据消息传递,本节点 A 采用跨层信息共享机制,如图 2 所示,节点 A 将自己有无其他邻居节点的信息装入 ACK 帧中,选取 IEEE802.11 标准^[12]规定的 ACK 帧的 Frame Control 域中的保留字段 To DS 记录邻居节点情况信息:当本节点 A 周围有其他邻居节点存在时则将该字段置“1”,否则置“0”。节点 A 装填好 ACK 帧的邻居信息后,节点 A 根据估算出来的传输距离 d_{ACK} 作如下判断:若 $d_{ACK} < R$ (R 是节点以固定功率 P 发送消息时的通信半径),则根据式(3)去选择适当的发射功率等级去发送 ACK 帧;若 $d_{ACK} \geq R$,则直接以固定功率去发送 ACK 帧给对方节点 B 。

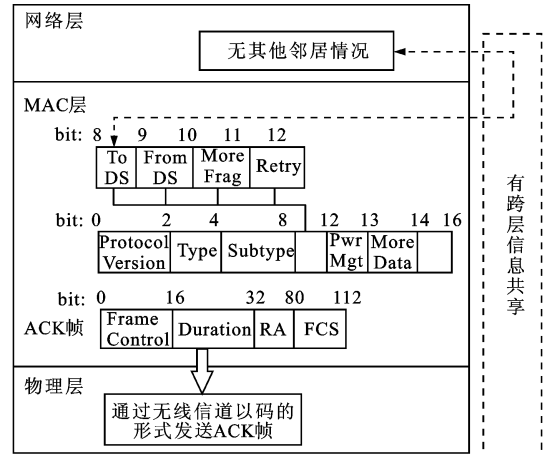


图 2 节点跨层信息共享示意图

(4)节点 A 发送完 ACK 帧后,以固定功率向 B 发送其请求矢量 RV。

(5)节点 B 收到节点 A 发送过来的 RV 消息后,记此时为 t_{22} ,再将自己有无其他邻居节点的信息装入对 RV 消息的 ACK 帧中并以固定功率发送 ACK 帧,并根据接收信号强度估算出此时 A 、 B 之间的距离

$$d_{AB22} = \left(c \frac{P}{P_3} \right)^{1/4} \quad (8)$$

式中: P_3 是接收的 RV 消息的信号强度。

节点 B 计算出 d_{AB22} 后,作如下判断。①若 d_{AB11}

$\leq d_{AB22}$, 表明2个节点有相互远离或者保持相对静止的趋势并且短时间内保持该趋势不变, 此时节点B根据计算出来的距离 d_{AB22} , 假设节点A以最大速度作相反方向运动的极端情况, 估算出来的发送 dataB1 数据消息的传输距离为

$$d_{dataB1} = d_{AB22} + T_{\max}(V_{\max} + V_{\min}) + \frac{(L_{RV} + 2L_{dataB1})V_{\max} + L_{dataB1}V_{\min}}{B} \quad (9)$$

式中: $V_{\min}$ 为节点自身的移动速度; L_{dataB1} 为 dataB1 数据消息的长度。节点B根据计算出来的 d_{dataB1} 去选择适当的发射功率等级即可。

节点B以发送 dataB1 数据消息的发射功率等级去发送节点B剩下的需要发送的数据, 直到等待时间 T_{wait2} 后再进行功率自适应调整。为了保证在等待时间 T_{wait2} 内节点使用同一发射功率等级发送消息时对方节点一定能够收到, 假设对方节点以最大速度与本节点作相反方向运动, 此时计算出来的等待时间 T_{wait2} 最小

$$\int_0^{T_{wait2}} V_B(t) dt + \int_0^{T_{wait2}} V_{\max} dt = d_0 - \text{mod}(d_{AB22}, d_0) \quad (10)$$

$$d_0 = \frac{R}{n} \quad (11)$$

式中: $V_B(t)$ 为节点B的运动速度; $V_{\max}$ 为节点的最大运动速度; R 为节点以最高发射功率等级发送消息时的通信半径范围; mod 表示取余数操作; n 是总的功率等级数; d_0 为2个相邻的发射功率等级对应的通信范围之差。等待时间 T_{wait2} 后, 节点B发送数据消息的功率自适应调整过程跟前面所述的一样。

②若 $d_{AB11} > d_{AB22}$, 表明2个节点有相互靠近运动的趋势并且短时间内保持该趋势不变; 如果节点A有其他邻居节点存在, 则节点B根据式(9)估算功率等级, 然后发送 dataB1 数据消息; 如果节点A没有其他邻居节点, 则节点B等待2个节点靠近至最小发射功率等级所对应的通信范围后, 再进行跨层功率调整以发送数据。设等待时间为 t_B , 则有

$$\bar{v} = \frac{d_{AB11} - d_{AB22}}{t_{22} - t_{11}} \\ t_B = \frac{d_{AB22} - d_{\min}}{\bar{v}} = \frac{(d_{AB11} - d_{\min})(t_{22} - t_{11})}{d_{AB11} - d_{AB22}} \quad (12)$$

式中: $d_{\min}$ 为节点最小的发射功率等级所对应的通信范围; $\bar{v}$ 为节点B从 t_{11} 到 t_{22} 这段时间的平均相对速度。

如果节点在等待时间内的运动方向没有发生突变(远离或者静止), 则等待时间 t_B 后, 节点以最小

的发射功率等级发送 dataB1 数据消息; 如果节点在等待时间内的运动方向发生突变, 则节点立即根据式(9)估算功率等级, 然后发送 dataB1 数据消息; 如果没有数据要发送给对方节点, 则节点立即根据 d_{AB22} 估算功率等级, 然后单播发送 Hello 消息给对方节点, 下一个 Hello 消息延迟到正常的发送时刻发送, 并且节点跨层泛听 Hello 消息, 即在 MAC 层判断数据段的长度是否等于 Hello 消息的长度, 如果是的话, 则不管在 MAC 层是单播还是广播都上传到网络层, 这样可避免 Hello 消息丢失。

(6)节点A、B交换数据消息。在此过程中, 发送数据消息用估算功率, 发送 ACK 帧用固定功率。

节点A收到对 RV 消息的 ACK 确认帧后, 记此时为时刻 t_2 ; 节点A根据接收信号强度估算出 A、B 之间的距离

$$d_{AB2} = \left(c \frac{P}{P_{ACK}} \right)^{1/4} \quad (13)$$

式中: P_{ACK} 是接收的 ACK 确认帧的信号强度。

节点A计算出来 d_{AB2} 后, 作如下判断。①若 $d_{AB1} \leq d_{AB2}$, 则表明2个节点有相互远离或者保持相对静止的趋势, 此时假设节点B以最大速度作相反方向运动的极端情况, 估算出来的发送 data1 数据消息的传输距离为

$$d_{data1} = d_{AB2} + T_{\max}(V_{\max} + V_{\min}) + \frac{(L_{ACK} + 2L_{data1})V_{\max} + L_{data1}V_{\min}}{B} \quad (14)$$

式中: $V_{\min}$ 为节点自身的移动速度; L_{data1} 为 data1 数据消息的长度; L_{ACK} 为 MAC 层 ACK 帧的长度。节点A计算出来 d_{data1} 后, 根据 d_{data1} 选择适当的发射功率等级发送数据 data1。节点A以发送 data1 数据消息的发射功率等级去发送节点A剩下的需要发送的数据, 直到等待时间 T_{wait} 后再进行功率自适应调整。对于等待时间 T_{wait} 的计算, 与步骤(5)中所述的一样

$$\int_0^{T_{wait}} V_A(t) dt + \int_0^{T_{wait}} V_{\max} dt = d_0 - \text{mod}(d_{AB2}, d_0) \quad (15)$$

式中 $V_A(t)$ 为节点A的运动速度。等待时间 T_{wait} 后, 节点A发送数据消息的功率自适应调整过程跟前面所述的一样。②若 $d_{AB1} > d_{AB2}$, 则表明2个节点有相互靠近运动的趋势; 如果节点B有其他邻居节点存在, 则节点A使用发送对 SV 消息的 ACK 确认帧的发射功率等级去发送 data1 数据消息; 如果节点B没有其他邻居节点存在, 则节点A等待2个节点靠近至最小发射功率等级所对应的通信范围

后再进行跨层功率调整发送数据。假设等待时间为 t_A , 则有

$$\left. \begin{aligned} \bar{v}_2 &= \frac{d_{AB1} - d_{AB2}}{t_2 - t_1} \\ t_A &= \frac{d_{AB2} - d_{\min}}{\bar{v}_2} = \frac{(d_{AB2} - d_{\min})(t_2 - t_1)}{d_{AB1} - d_{AB2}} \end{aligned} \right\} \quad (16)$$

式中: $d_{\min}$ 为节点最小的发射功率等级所对应的通信范围; $\bar{v}_2$ 为节点 A 从 t_1 到 t_2 这段时间的平均相对速度。

如果节点在等待时间内的运动方向没有发生突变(远离或者静止), 则节点在等待时间 t_A 后, 以最小的发射功率等级发送 data1 数据消息; 如果节点在等待时间内的运动方向发生突变, 则节点立即根据式(14)估算出来的发射功率等级去发送 data1; 如果没有数据要发送给对方节点, 则节点立即根据 d_{AB2} 估算功率等级, 然后单播发送 Hello 消息给对方节点, 具体操作与步骤(5)中所述的相同。

节点 A 和 B 收到对方发来的数据消息后, 都以固定功率发送对应的 ACK 帧。在 2 个节点随机争用信道发送数据消息的过程中, 如果出现其中一个节点在等待时间 T_{wait} 或者 $T_{\text{wait}2}$ 内一直没有机会发送自己缓存的数据消息, 则在等待时间 T_{wait} 或者 $T_{\text{wait}2}$ 后, 节点有机会发送数据消息时, 该节点将以固定功率去发送自己将要发送的第一个数据消息, 之后的功率自适应调整过程与前面所述的一样。

此外, 在 ERAPC 算法中, 节点也采用数据消息有条件地广播策略, 即当条件 {有多个邻居节点} $\cap$ {邻居中无目的节点} $\cap$ {有多个邻居节点同时需要接收该数据消息} 同时满足时, 当前节点对该数据消息进行广播。并且规定, 此时当前节点选择的发射功率等级必须要保证需要接收数据消息的多个邻居节点一定能够接收到该数据消息。

3 ERAPC 算法的能耗分析

定理 1 ERAPC 算法相比于固定发射功率的能耗平均减少 $0.56\epsilon_{\text{fs}}r^2$ (r 为节点以固定发射功率等级发送数据消息时的通信半径)。

证明 设网络中的任意一个节点位于通信半径为 r 的圆形区域中心, 邻居节点可以位于该圆形区域的任一点, x 表示 2 个节点相遇后的距离, 则 x 的分布函数为

$$F(x) = P(X \leq x) = \frac{\pi x^2}{\pi r^2} = \frac{x^2}{r^2} \quad (17)$$

得出上述分布函数的概率密度函数为

$$f(x) = \begin{cases} \frac{2x}{r^2}, & 0 < x < r \\ 0, & \text{其他} \end{cases} \quad (18)$$

得出 2 个节点相遇后的平均距离为

$$E(x) = \int_0^r f(x)x dx = \frac{2r}{3} \quad (19)$$

根据式(1)可知, 节点以固定发射功率发送 1 bit 信息所需的发送能耗为

$$E_{\text{TX1}} = E_{\text{elec}} + \epsilon_{\text{fs}}r^2 \quad (20)$$

式中: ϵ_{fs} 为功率放大电路的功耗系数。

节点以估算功率发送 1 bit 信息的平均发送能耗为

$$E_{\text{TX2}} = E_{\text{elec}} + \epsilon_{\text{fs}}\left(\frac{2r}{3}\right)^2 = E_{\text{elec}} + \frac{4}{9}\epsilon_{\text{fs}}r^2 \quad (21)$$

故得出节点以固定发射功率发送 1 bit 信息和以估算功率发送 1 bit 信息的能耗差为

$$E_{\text{TX0}} = \frac{5}{9}\epsilon_{\text{fs}}r^2 \approx 0.56\epsilon_{\text{fs}}r^2 \quad (22)$$

由此可知, 相比于以固定发射功率发送消息时的情况, ERAPC 算法每发送 1 bit 信息, 平均可减少大概 $0.56\epsilon_{\text{fs}}r^2$ J 的发射能耗, 从而命题得证。

定理 2 相比于 ERBC 算法, ERAPC 算法的能耗平均减少为 $\frac{5}{18}pTK_{\text{data}}\epsilon_{\text{fs}}r^2 + \lambda aN\left((K_{\text{req}} + K_{\text{SV}})\left(\frac{4\epsilon_{\text{fs}}r^2}{9} + E_{\text{elec}}\right) + \frac{5K_{\text{ACK}}\epsilon_{\text{fs}}r^2}{9}\right)$, 其中 p 为网络中数据消息的平均产生速率, T 为网络运行时间, K_{data} 、 K_{req} 、 K_{SV} 和 K_{ACK} 分别为单个数据消息、单个 RV 消息、单个 SV 消息和单个 ACK 帧的比特数, λ 为网络场景中节点的度, N 为网络场景中总的节点数, a 为 2 个节点相遇后相互之间进行消息交互的概率, E_{elec} 为发送和接收单位比特信息的损耗常量。

证明 假设在一个 $L \times W$ 的矩形区域中, 每个采样时刻 t 所对应的网络可表示为一个简单无向图 $G_{p(r)}(V, E, t)$, 其中 V 表示场景中所有的节点, 其网络规模为 $N = |V|$, $p(r)$ 表示当通信半径为 r 时场景中任意 2 个节点存在无线通信链路的概率, 当且仅当 $|v_j - v_i| \leq r$ 时, 2 个节点 $\{v_j, v_i\}$ 互为邻居, 即组成链路集合 E , 则由条件可知, 场景中节点的度为

$$\lambda = \pi r^2(N/L)W - 1 \quad (23)$$

节点的到达数满足泊松分布

$$P_r\{X = n\} = \frac{e^{-\lambda}\lambda^n}{n!} \quad (24)$$

假设 2 个节点相遇后, 相互之间进行消息交互的概率为 a , 则平均每个节点与其相遇节点进行消息交互的次数 Y 服从强度为 λa 的泊松分布, 则得

出总的节点相遇并进行信息交互的次数为 λaN 。

根据算法原理可知,在 ERAPC 算法中所有发送的数据消息都能以估算功率发送,而在 ERBC 算法中,只能有一半的数据消息以估算功率发送,则相比于 ERBC 算法,ERAPC 算法平均可减少的数据消息发送能耗为

$$P_{data} = \frac{1}{2}pTK_{data}\frac{5\epsilon_{fs}r^2}{9} = \frac{5}{18}pTK_{data}\epsilon_{fs}r^2 \quad (25)$$

ERAPC 算法在每次节点相遇时减少了一个 RV 消息和一个 SV 消息的发送,并且减少的这个 RV 消息和 SV 消息在 ERBC 算法中都是以估算功率发送的,因此相比于 ERBC 算法,ERAPC 平均减少的总的控制消息发送能耗为

$$P_{control} = (\lambda aNK_{req} + \lambda aNK_{sv})\left(\frac{4\epsilon_{fs}r^2}{9} + E_{elec}\right) = \lambda aN(K_{req} + K_{sv})\left(\frac{4\epsilon_{fs}r^2}{9} + E_{elec}\right) \quad (26)$$

另外,在 ERBC 算法中,节点每次以固定发射功率发送 ACK 确认帧,而在 ERAPC 算法中,节点在每次相遇交互消息时,可以以估算功率发送 ACK 确认帧,则 ERAPC 算法平均可减少的 ACK 确认帧的发送能耗为

$$P_{ACK} = \lambda aNK_{ACK}\frac{5\epsilon_{fs}r^2}{9} = \frac{5}{9}\lambda aNK_{ACK}\epsilon_{fs}r^2 \quad (27)$$

综上,即得出在相同的网络条件下,ERAPC 算法比 ERBC 算法平均减少的发射能耗为

$$P_{EERPC} = P_{data} + P_{control} + P_{ACK} = \frac{5}{18}pTK_{data}\epsilon_{fs}r^2 + \lambda aN\left((K_{req} + K_{sv})\left(\frac{4\epsilon_{fs}r^2}{9} + E_{elec}\right) + \frac{5K_{ACK}\epsilon_{fs}r^2}{9}\right) \quad (28)$$

显然, $P_{EERPC} > 0$,即 ERAPC 算法的能耗相对更小。证毕。

4 仿真分析

使用仿真软件平台 OPNET Modeler 14.5 进行仿真。参考经典的 Epidemic 算法^[13]和 ERBC 算法在验证过程中采用的仿真参数设置,同时考虑目前常用的 IEEE802.11a 标准定义的指标,对主要仿真参数的设置如表 1 所示。

根据节点初始通信范围的不同,设置了 5 个不同的仿真场景,在每个仿真场景中都将节点的发射功率等级设置为 8 级。节点的不同初始通信范围所对应的 8 个功率等级通信范围如表 2 所示。

在每个仿真场景下分别运行 ERAPC、Epidemic、

表 1 仿真参数设置

参数	数值
网络面积/m ²	1 500×600
节点运动速度/m·s ⁻¹	1~20
分组产生间隔/s	1
节点个数	100
仿真时间/s	3 600
节点通信半径/m	10,25,50,75,100
节点初始能量/J	10
Hello 消息广播周期/s	3
发射功率等级	8
功耗系数 ϵ_{fs}	1×10^{-11}
损耗常量 E_{elec}	5×10^{-8}
数据传输速率/Mb·s ⁻¹	54
节点死亡能量阈值/J	0.005

表 2 能量等级对应的通信范围

节点初始通信半径/m	8 个功率等级对应的通信半径/m
10	10,9,8,7,6,5,4,3
25	25,22,19,16,13,10,7,4
50	50,44,38,32,26,20,14,8
75	75,66,57,48,39,30,21,12
100	100,87,74,61,48,36,23,10

n -Epidemic 和 ERBC 算法,并分别统计这 4 个算法在不同仿真场景下的节点比特能耗、归一化控制开销、网络寿命和数据消息传送成功率,每组实验分别进行 5 次,实验结果取平均值。所得到的实验结果如图 3~图 6 所示。

图 3 显示了在不同的仿真场景下 1 bit 数据消息从源节点成功送达目的节点所消耗的平均能量。从图 3 中可以看出,随着节点通信范围的扩大,统计出的节点比特能耗增大,这是因为节点通信范围的扩大使得节点之间的接触机会增多,节点之间交互次数增多,节点能耗增大的缘故。Epidemic 算法和 n -Epidemic 算法因为没有使用发射功率自适应调整策略,故其节点比特能耗较大。

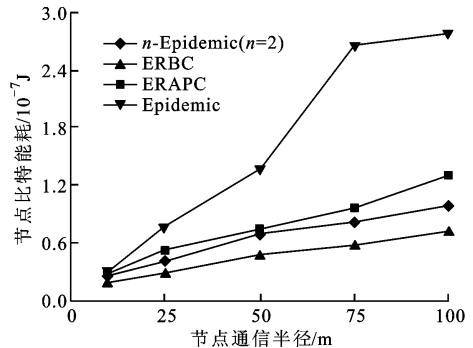


图 3 4 种算法的节点比特能耗比较

此外,相比于 ERBC 算法,新算法 ERAPC 的节点比特能耗最多减少了 31.92%。这是由于以下 2 个原因造成的:一是新算法 ERAPC 优化了功率自适应调整过程的节能策略,有条件地等待 2 个节点靠近后再进行跨层功率调整发送数据消息,增加了估算等待时间 T_{wait} ,在节点发送消息时多次动态调整节点发射功率,使得新算法减少了能量浪费现象;二是新算法 ERAPC 减少了 RV 消息和 SV 消息的发送个数,从而减少了节点发送多余消息造成的能量消耗。

实验中的统计量归一化控制开销是所有节点发出的控制消息包含的比特数与所有节点发送的控制消息和到达目的节点的数据消息包含的比特数之和的比值。图 4 为 4 种算法的归一化控制开销比较。由图 4 可见,本文 ERAPC 算法的归一化控制开销在每个场景中均低于 ERBC、Epidemic 和 n -Epidemic 算法,这主要是因为 ERAPC 算法减少了控制消息 SV 和 RV 的发送个数。

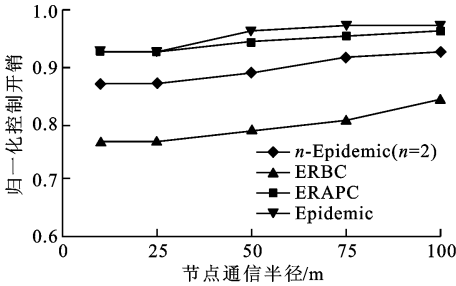


图 4 4 种算法的归一化控制开销比较

图 5 显示的是不同仿真场景下节点网络寿命比较情况。由图 5 可见,随着节点通信范围的扩大,第一个节点死亡的时间越来越早,这是因为随着节点通信范围的扩大,节点间的接触机会增多,消息交互频繁程度加大,这使得节点的能量消耗增大,进而导致节点容易过快死亡。另外,如图 5 中所示,使用了 ERAPC 算法的网络寿命要长于另外 3 种算法的网

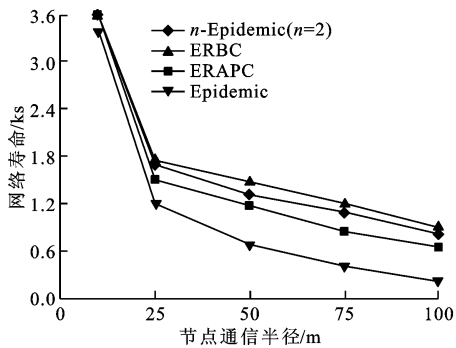


图 5 4 种算法的网络寿命比较

络寿命,这是因为 ERAPC 减少了 RV 消息和 SV 消息的发送,有效地减少了节点的发射能量消耗,从而延长了网络寿命。

图 6 显示的是不同的仿真场景中成功接收的数据消息数占发送数据消息总数的百分比的比较情况。由图 6 可见,随着节点通信范围的扩大,数据消息传送成功率出现了一定程度的降低,这是因为随着节点通信范围的扩大,节点间的接触机会增多,会出现导致部分节点的能量消耗过快而死亡的情况,进而造成数据消息传送成功率的降低。另外,ERAPC 算法的数据消息传送成功率要高于另外 3 种算法,这是因为 ERAPC 算法比其他 3 种算法更加节能的缘故。

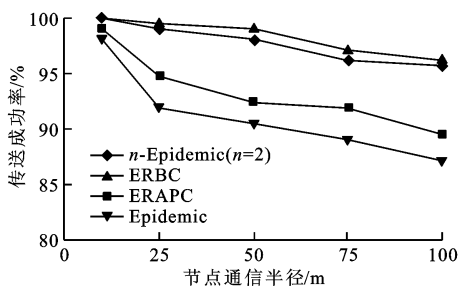


图 6 4 种算法的数据消息传送成功率比较

5 结 论

本文针对基于功率控制的机会网络节能路由算法存在的节能效果不足和控制开销偏大等问题,提出一种自适应动态功率控制的机会网络节能高效路由算法 ERAPC。新算法重新设计了节点的自适应动态发射功率调节机制,调整了用于测距的消息种类,有条件地等待 2 个节点靠近后再用更小的功率发送数据消息,并且减少了 RV 和 SV 消息的冗余发送。仿真结果表明,与 ERBC 算法相比,ERAPC 算法至少降低了 27.27% 的数据传输能耗和 11.87% 的控制开销。

参考文献:

[1] POONGUZHARSELVI B, VETRISSELVI V. Survey on routing algorithms in opportunistic networks [C]// Proceedings of International Conference on Computer Communication and Informatics. Piscataway, NJ, USA: IEEE, 2013: 1-5.

[2] CAO Y, SUN Z, WANG N, et al. Converge-and-diverge: a geographic routing for delay/disruption-tolerant networks using a delegation replication approach [J]. IEEE Transactions on Vehicular Technol-

- ogy, 2013, 62(5): 2339-2343.
- [3] NIU J, GUO J, CAI Q, et al. Predict and spread: an efficient routing algorithm for opportunistic networking [C]// Proceedings of 2011 IEEE Wireless Communications and Networking Conference. Piscataway, NJ, USA: IEEE, 2011: 498-503.
- [4] LIU H, CHEN Y. A moving scope aware routing approach for opportunistic networks [C]// Proceedings of 2010 International Conference on Wireless Communications and Signal Processing. Piscataway, NJ, USA: IEEE, 2010: 1-5.
- [5] CHOI B J, SHEN X. Adaptive asynchronous sleep scheduling protocols for delay tolerant networks [J]. IEEE Transactions on Mobile Computing, 2011, 10(9): 1283-1296.
- [6] WANG W, MOTANI M, SRINIVASAN V. Opportunistic energy-efficient contact probing in delay-tolerant applications [J]. IEEE/ACM Transactions on Network, 2009, 17(5): 1592-1605.
- [7] LU X, HUI P. An energy-efficient n-epidemic routing protocol for delay tolerant networks [C]// Proceedings of the 5th International Conference on Networking, Architecture, and Storage. Piscataway, NJ, USA: IEEE, 2010: 341-347.
- [8] 付凯, 夏靖波, 尹波. DTN 中一种基于接触时间的休眠机制 [J]. 计算机科学, 2013, 40(2): 87-90.
FU Kai, XIA Jingbo, YIN Bo. Sleep scheme based on contact time in DTN [J]. Computer Science, 2013, 40(2): 87-90.
- [9] YAO Y K, ZHENG W X, REN Z. An energy-efficient routing algorithm for disruption tolerant networks [C]// Proceedings of the 2012 2nd International Conference on Computer and Information Application, Paris, France: Atlantis Press, 2012: 895-898.
- [10] 任智, 索建伟, 陈红. 基于相遇节点跨层感知的机会网络高效低时延路由算法 [J]. 通信学报, 2013(10): 1-8.
REN Zhi, SUO Jianwei, CHEN Hong. Efficient low-delay routing algorithm for opportunistic networks based on cross-layer sensing of encountered nodes [J]. Journal on Communications, 2013(10): 1-8.
- [11] HEINZELMAN W B, CHANDRAKASAN A P, BALAKRISHNAN H. An application-specific protocol architecture for wireless microsensor networks [J]. IEEE Transactions on Wireless Communications, 2002, 1(4): 660-670.
- [12] GU D, ZHANG J. Qos enhancement in IEEE 802.11 wireless local area networks [J]. IEEE Communications Magazine, 2003, 41(6): 120-124.
- [13] VAHDAT A, BECKER D. Epidemic routing for partially connected ad hoc networks, CS-2000-06 [R]. Durham, NC, USA: Duke University, 2000.

(编辑 刘杨)

(上接第7页)

- [4] KONAK A, KULTUREL-KONAK S, AZİZOĞLU M. Minimizing the number of tool switching instants in flexible manufacturing systems [J]. International Journal of Production Economics, 2008, 116(2): 298-307.
- [5] 边培莹. PSO 算法在 FMS 刀具可复用调度中的应用与仿真 [J]. 机械设计与制造, 2012(3): 76-78.
BIAN Peiying. Application and simulation of PSO in FMS based on tools reusable scheduling [J]. Machinery Design & Manufacture, 2012(3): 76-78.
- [6] 张亮, 楼佩煌, 胡武茹, 等. 一种改进型遗传算法在 FMS 刀具调度中的应用 [J]. 工业控制计算机, 2009, 22(9): 75-78.
ZHANG Liang, LOU Peihuang, HU Wuru, et al. Application of an improved genetic algorithm for tool scheduling in FMS [J]. Industrial Control Computer, 2009, 22(9): 75-78.
- [7] PRABAHARAN T, NAKKEERAN P R, JAWAHAR N. Sequencing and scheduling of job and tool in a flexible manufacturing cell [J]. The International Journal of Advanced Manufacturing Technology, 2006, 29: 729-745.
- [8] 赵丹, 张家泰, 舒海生, 等. 基于双重遗传算法的工件流与刀具流综合调度优化 [J]. 西南交通大学学报, 2010, 45(6): 926-931.
ZHAO Dan, ZHANG Jiatai, SHU Haisheng, et al. Double layer genetic algorithm for integrated scheduling optimization of part and tool flows [J]. Journal of Southwest Jiaotong University, 2010, 45(6): 926-931.
- [9] 陈世哲, 刘国栋, 浦欣, 等. 基于优势遗传的自适应遗传算法 [J]. 哈尔滨工业大学学报, 2007, 39(7): 1021-1024.
CHEN Shizhe, LIU Guodong, PU Xin, et al. Adaptive genetic algorithm based on superiority inheritance [J]. Journal of Harbin Institute of Technology, 2007, 39(7): 1021-1024.

(编辑 杜秀杰)