

无线传感器网络中基于自适应定时器策略的分簇算法

曹涌涛, 何 晨, 蒋铃鸽
(上海交通大学电子工程系, 上海 200240)

摘 要: 提出了一种适用于无线传感器网络的基于自适应定时器策略的分簇算法. 该算法通过自适应调整节点的等待时间来保证较多电池能量的节点有更大的机会成为簇首. 理论分析表明该算法不仅实现了系统的动态负载均衡, 并且能够通过参数的设置确保簇首的均匀分布. 仿真结果显示该算法有效地延长了系统生命, 提高了网络的能耗效率.

关键词: 无线传感器网络; 分簇算法; 系统生命

中图分类号: TP393 **文献标识码:** A **文章编号:** 0372-2112 (2007) 09-1719-05

A Distributed Timer-Based Clustering Algorithm for Wireless Sensor Networks

CAO Yong-tao, HE Chen, JIANG Ling-ge
(Department of Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240 China)

Abstract: We present a timer-based distributed clustering algorithm. By adaptively adjusting the wakeup rate of the exponential distribution, a node with higher residual energy is more likely to be elected clusterhead. Moreover, the algorithm is able to ensure that clusterheads are well scattered. Simulation experiments illustrate that our algorithm is able to significantly prolong system life compared with previous ones.

Key words: wireless sensor networks; clustering; system life

1 引言

无线传感器网络中分簇的概念最早是在分组无线网中提出的, 当时主要用于层次型路由^[1]. 随着研究的不断深入, 迄今为止, 已经提出了大量的分簇算法来构造和维护分簇网络结构. 基于分簇网络, 可以减少路由算法和洪泛广播的开销, 方便地管理节点和控制信道接入, 并可以提高网络资源的使用效率^[2, 5].

在现有的无线传感器网络分簇算法^[2~6]中, Heinzelman 等所提出的 LEACH 算法^[2]具有代表性. 该算法通过保证在一个工作周期中, 每一个节点都有一次机会且只有一次机会成为簇首来实现负载均衡. 尽管 LEACH 的初衷是均摊系统的负荷, 但仿真结果显示 LEACH 会导致某些节点过早地耗尽电池能量. 经过分析, 我们认为有两个原因可以解释这个现象:

由于 LEACH 算法的随机性, 一些没有成为簇首的节点被迫与基站直接通信. 尽管文献[2]在假定传感器节点均匀随机分布的情况下, 计算出了系统中簇首数目的最优值. 但由于 LEACH 算法中簇首产生是随机的, 所

以每一轮中产生的簇首数目不总是等于预先设定的最优值. 如果产生太少的簇首, 某个普通节点的附近将很有可能没有簇首的存在, 所以这个节点被迫与远方的基站直接通信. 即使是能够产生系统所期望数量的簇首, 但是由于簇首产生的随机性, 会导致簇首的分布不均^[6], 同样会使得某些节点的附近没有簇首的存在而成为簇首与基站直接通信, 我们称之为“被动型簇首”. 无线传感器网络采用低天线、近地的无线通信模式, 传送信号所需要的发射功率随通信距离 d 的指数倍 d^α 成正比, 其中 $2 \leq \alpha \leq 4$, 长距离的通信将会极大地消耗节点有限的电池能量, 从而导致某些节点过快地耗尽电池能量而“死亡”. 也就是说, 过多的“被动型簇首”的产生将会缩短传感器网络的系统生命.

LEACH 决定节点的“角色”并没有考虑节点的剩余电池能量, 这样的负载均衡策略是不完备的. LEACH 按照一个预先定义的概率随机地决定节点的“角色”. 尽管一个节点, 在一个工作周期中, 只有一次机会成为簇首, 但是其它原因会导致它的电池能量过度的消耗, 比如被迫与基站直接通信, 或者因为分布不均管理过多的普通

节点等等. 在 LEACH 的下一个工作周期中, 所有的节点还是具有同样的机会成为簇首, 那些电池能量已经在警戒线以下的节点会因为成为簇首而很快地耗尽电池能量.

2 基于自适应定时器策略的分簇算法

2.1 分簇算法

分簇算法如图 1 所示, 算法参数的说明见表 1. 算法也划分为“轮”, 每一个轮分为簇形成和稳定工作两个阶段. 在簇形成阶段, 节点 i 首先处于“等待”模式, 产生区间 $[0, 1]$ 上均匀分布的随机变量 x_i ,

表 1 算法的参数说明

$N(i)$	节点 i 邻居的集合
$E_{residual}^i$	节点 i 剩余电池能量
E_{max}	最大电池能量
λ_i	节点 i 的等待速率
λ_{max}	最大等待速率
λ_{min}	最小等待速率
T_{CF}	最长的簇形成时间
$Forced_CH$	被动型簇首的标志

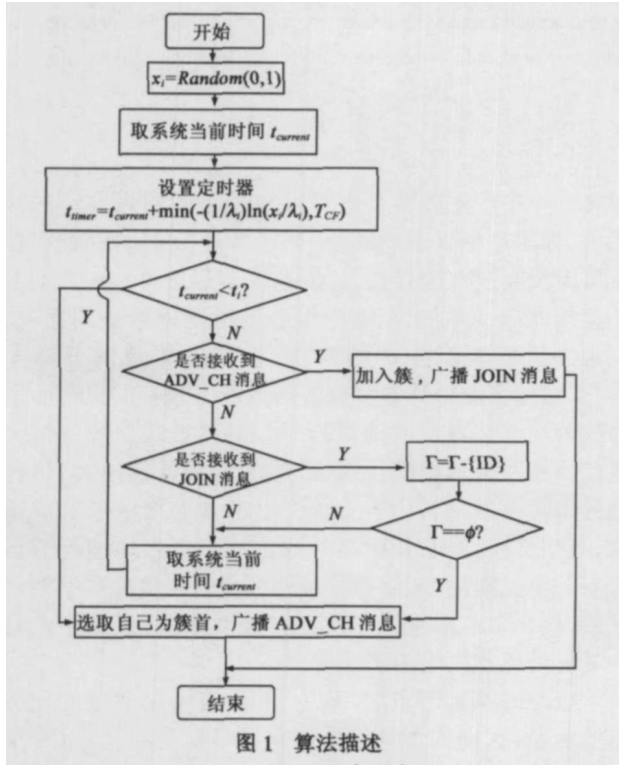


图 1 算法描述

$$x_i = \text{Random}(0, 1)$$

然后按照负指数分布生成一个定时长度 t_i :

$$x_i = \lambda_i e^{-\lambda_i t_i} \Rightarrow t_i = - (1/\lambda_i) \ln(x_i/\lambda_i) \tag{1}$$

其中 λ_i 为等待速率, 取值为

$$\lambda_i = \lambda_{min} + (\lambda_{max} - \lambda_{min}) \frac{E_{residual}^i}{E_{max}} \tag{2}$$

最后设置定时器 $t_{timer} = t_{current} + \min(t_i, T_{CF})$, $t_{current}$ 为当

前时钟.

当某一个节点 v 的定时到达时, 它首先竞争信道, 如果成功, 则选举自己成为簇首并广播 ADV_CH 消息通知其邻居. 这里根据节点的剩余电池能量自适应地调整参数, 希望具有较多剩余能量的节点较早地结束等待, 成为簇首. 当它的邻居接收到这个消息时, 就停止计时并加入到 v 的簇. 如果一个节点同时接收到多个 ADV_CH 消息, 则可以使用 ID 或者与簇首的距离 (信号的强弱) 来选择加入某一个簇, 然后广播 JOIN 消息并结束算法. 当节点 v 收到一个消息 JOIN(u, t), 首先检查自己之前是否广播过 ADV_CH 消息, 如果节点 v 之前没有广播 ADV_CH 消息, 则记录下 u 的决定. 如果 v 的所有的邻居在 v 预先设定的等待时间到达之前都已经决定了自己的“角色”, 即加入了其它的簇, 则 v 停止计时器并成为一个“被动型”簇首. 当 T_{CF} , 最大簇形成时间到达时, 所有没有决定自己“角色”的节点也自动成为“被动型簇首”.

3.3 算法性能分析

本文的算法有以下的特性:

(1) 算法是完全分布式的, 节点完全依赖本地信息做出决定, 不需要全网的拓扑信息.

(2) 簇形成的过程最多需要 T_{CF} 的时间. 由算法可知, 当最大簇形成时间 T_{CF} 到达时, 所有没有决定自己“角色”的节点将停止计时器, 结束算法, 所以 T_{CF} 是算法的时间上限.

(3) 算法的消息复杂性是 $O(1)$ 的. 显然, 在算法的执行过程中, 每一个节点只需要广播一个 ADV_CH 或 JOIN 消息.

同时与 LEACH 算法相比, 本文算法具有更好的性能:

引理 1 与 LEACH 算法相比, 本算法能够获得更好的动态负载均衡.

证明 实现负载均衡, 平均分摊系统能耗到各个节点上是延长系统生命的有效手段. 因为簇首完成数据融合以及远距离通信等任务, 需要消耗更多的能量, 所以分簇算法应该选择较多电池能量的节点承担簇首的功能.

本文算法使用节点剩余电池能量作为参数来实现负载均衡, 希望较多电池能量的节点有更大的机会成为簇首, 也就是使这些节点有较短的等待时间. 根据式 (1)、(2), 为了保证这个方案的可行性, 等待时间 t_i 需要随参数 λ_i 的增加而单调减小. 由负指数分布 $x_i = \lambda_i e^{-\lambda_i t_i}$, 得到以 λ_i 为自变量关于 t_i 的函数 $t_i(\lambda) = -\frac{1}{\lambda} \ln \frac{x_i}{\lambda}$, 这个函数的一阶导数是

$$\frac{dt_i}{d\lambda_i} = \frac{1}{\lambda_i} (1 - \ln \frac{\lambda_i}{x_i})$$

因为 $0 < x_i < 1$, 显然当 $\lambda_i \geq x_i e$, $t_i > 0$ 同时有 $\frac{dt_i}{d\lambda_i} \leq 0$, 即函数 $t_i(\lambda_i)$ 在区间 $[0, +\infty)$ 上单调减小. 所以当我们选择

$$\lambda_i \geq \max(x_i e) = e \quad (3)$$

等待时间随参数 λ_i 的增加而单调减小.

同时由式(2)知 λ_i 是节点剩余能量 $E_{residual}^i$ 的单调增函数, 所以等待时间是节点剩余能量 $E_{residual}^i$ 的单调减函数. 等待时间的减少也意味着这些拥有较多电池能量的节点有更大的机会成为簇首, 从而实现整个系统的负载均衡. 而 LEACH 算法并没有考虑节点电池能量的信息, 所以它的负载均衡策略是不完备的.

引理 2 通过设置 λ_i 的上限, 本文算法能够保证簇首的均匀分布.

证明 不失一般性, 假定 i 和 j 是工作区域内任意两个相邻节点, $\lambda_k (k = 1, 2)$ 和 $t_k(i, j)$ 是这两个节点的等待速率和等待时间长度.

由式(1), 令

$$x_k = \lambda_k e^{-\lambda_k t_k} \quad (4)$$

得到

$$t_k = -\frac{1}{\lambda_k} \ln \frac{x_k}{\lambda_k}, \quad k = i, j \quad (5)$$

根据算法的描述, x_i 和 x_j 是两个独立的在区间 $[0, 1]$ 上均匀分布的随机变量. 因此, t_i 和 t_j 也是相互独立的随机变量, 其概率分布为

$$f_{T_k}(t_k) = \begin{cases} \lambda_k^2 \exp(-\lambda_k t_k), & t_k \geq (1/\lambda_k) \ln \lambda_k \\ 0, & \text{otherwise} \end{cases}, \quad k = i, j \quad (6)$$

由式(3)知 $\lambda_k \geq e$, 得到

$$\lambda_k^2 \exp(-\lambda_k t_k) > 0 \quad (7)$$

和

$$(1/\lambda_k) \ln \lambda_k > 0 \quad (8)$$

用 T_c 表示节点间最大时延, 则这两个节点同时成为簇首的概率为

$$F(|t_i - t_j| \leq T_c) = \iint_{|t_i - t_j| \leq T_c} f_{T_i}(t_i) f_{T_j}(t_j) dt_i dt_j$$

利用(6)、(7)和(8)独立随机变量的特性, 得到

$$\begin{aligned} F(|t_i - t_j| \leq T_c) &\leq \int_0^{T_c} \int_0^{T_c} f_{T_i}(t_i) f_{T_j}(t_j) dt_i dt_j \\ &+ \int_{T_c}^{\infty} \int_{t_j - T_c}^{t_j} f_{T_i}(t_i) f_{T_j}(t_j) dt_i dt_j \\ &= \lambda_i \lambda_j - \frac{\lambda_i \lambda_j}{\lambda_i + \lambda_j} (\lambda_i e^{-\lambda_i T_c} + \lambda_j e^{-\lambda_j T_c}) \end{aligned}$$

因为 $\lambda_k < \lambda_{\max} (k = 1, 2)$, 所以

$$F(|t_i - t_j| \leq T_c) \leq \lambda_{\max}^2 (1 - e^{-\lambda_{\max} T_c})$$

利用欧拉公式,

$$F(|t_i - t_j| \leq T_c) \leq \lambda_{\max}^2 \left[\lambda_{\max} T_c - \frac{(\lambda_{\max} T_c)^2}{2!} + \frac{(\lambda_{\max} T_c)^3}{3!} - L \right]$$

选择

$$\lambda_{\max} < 1/T_c \quad (9)$$

得到

$$F(|t_i - t_j| \leq T_c) \leq \lambda_{\max}^3 T_c \quad (10)$$

ε_2 是一个预先设定的极小值, 如果期望

$$F(|t_i - t_j| \leq T_c) \leq \varepsilon_2 \quad (11)$$

由式(9)、(10), 选择

$$\lambda_{\max} \leq \min \left\{ \sqrt[3]{\varepsilon_2 / T_c}, 1/T_c \right\} \quad (12)$$

则式(11)将得到满足. 换句话说, 通过为等待速率设置上限, 算法能够保证产生相邻节点同时成为簇首的概率低于某一个设定的极小值, 即簇首是均匀分布的.

例如, 当 ADV_CH 消息的长度为 100 比特, 无线链路的传输速率设为 1Mbps 的时候, 得到 $T_c \approx 10^{-4}$ 秒 (忽略短距离通信中的传播时延). 当设置 $\lambda_{\max} = 5$, 相邻节点同时成为簇首的可能性为 0.0125. 式(12)是一个相当松的上界, 所以实际应用中这个概率会更小. 在该算法的仿真实验中, 没有相邻的节点被选为簇首.

根据前文的分析, 簇首分布均匀将会大大降低普通节点成为“强迫型簇首”的可能, 减少节点与基站进行远程通信的次数, 节约有限的系统能量.

3 仿真实验和结果

通过仿真实验来评价本文所提出的算法. 实验模型由 100 个节点组成, 它们随机均匀分布在 $(x = 0, y = 0)$ 至 $(x = 100, y = 100)$ 的这个正方形区域内, 每个节点的初始电池能量为 2 焦耳, Sink 节点位于 (50, 175) 的位置. 与 LEACH 相似, 每个节点有功率控制功能: 两个传输电平分别用于簇间通信和远程 (与基站) 通信, 对应的无线覆盖半径分别是 25m 和 200m.

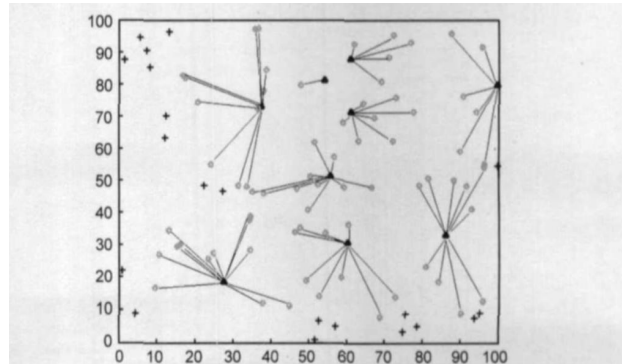


图 2 LEACH 的仿真结果输出

图 2 和图 3 是 LEACH 和本算法一次典型的仿真结

果输出, 图中三角形代表簇首, 圆圈代表普通节点, 十字代表“强迫型簇首”. 两种算法都产生了 9 个簇首, 但是产生的“强迫型簇首”的数目相差很大: LEACH 有 20 个而本文的算法只有 3 个, 显然这是我们的算法所产生的簇首分布均匀的结果. 我们也通过多次仿真实验, 计算了两种算法产生“强迫型簇首”的平均值, 两种算法分别是 28 和 2.4, “强迫型簇首”的减少将会降低节点与基站直接通信的次数.

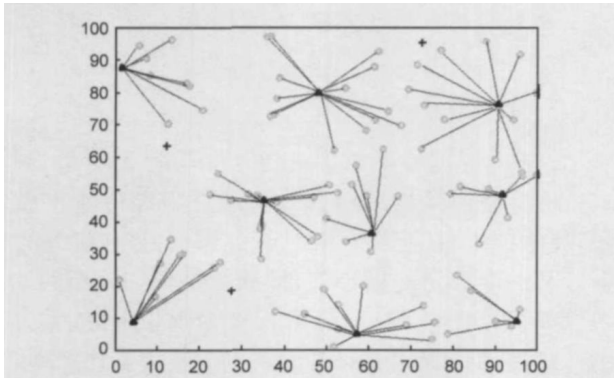


图 3 本文算法的仿真结果输出

为了分析系统的能耗, 我们引入 Heinzelman^[2]所提出的一种简单的无线信道模型, 分组比特长度 $L = 10000\text{bit}$. 在每一个稳定工作阶段, 每一个普通节点收集环境参数生成一个原始的数据分组并向簇首发送, 簇首将簇内成员(包括簇首本身)的所有数据分组进行数据融合, 重新封装生成一个新的分组再传送到 Sink 节点. 如果一个节点的附近没有簇首的产生, 它将直接把原始数据分组传递给远方的 Sink 节点. 这里我们把传递的原始数据分组称为“有效数据分组”. 对于 LEACH, 设置期望的簇首数目为 $k = 5$, 对于本文算法, 设 $\lambda_{\text{max}} = 5$ 以及 $\lambda_{\text{min}} = 2.8$. 同时为了进一步验证本文所提出算法在降低系统能耗方面的有效性, 我们引入另外一种考虑节点剩余电池能量的分簇算法-HEED^[5]加以比较, 参数 p_{min} 和 CH_{prob} 分别设为 0.0005 和 5%.

首先我们测试了传感器网络的系统生命. 从图 4 可以看出, 与 LEACH 以及 HEED 算法相比, 我们的算法明

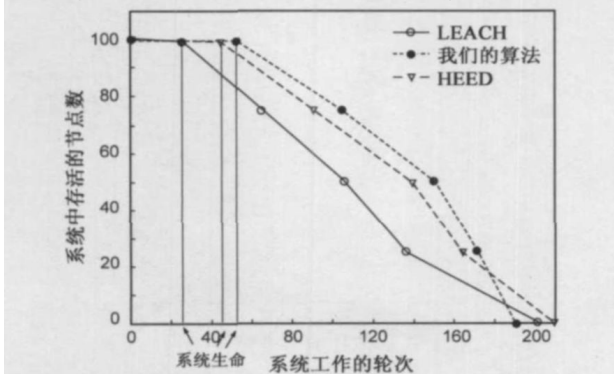


图 4 系统生命的比较

显地延长了系统生命. 正如前文所述, 这是因为 LEACH 的随机性将会导致某些节点过快地耗尽电池能量. 而我们的算法因为考虑了每个节点的剩余电池能量, 能量越多的节点越有可能成为簇首, 所以能够更好地保证负载均衡. HEED 算法尽管考虑了节点的剩余电池能量, 但是在每一个轮次都要广播节点权值的机制会带来额外的能耗, 特别是在一个高分布密度的网络中.

图 5 的仿真结果显示了系统生命和传递的有效数据分组的关系. 当第一个节点电池能量耗尽的时候, 我们的算法与 LEACH 相比, 系统将会多传递 50% 的“有效数据分组”. 这是因为 LEACH 会导致节点将有限的电池能量消耗在与基站的长距离通信上而我们的算法则避免了这一点, 从而可以传递更多的有用分组.

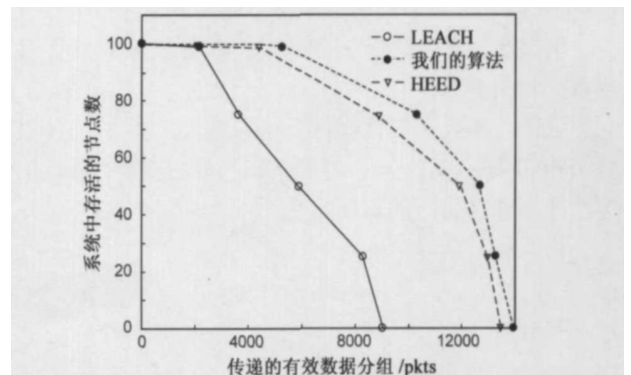


图 5 传递的有效数据分组与系统中存活节点数量的关系

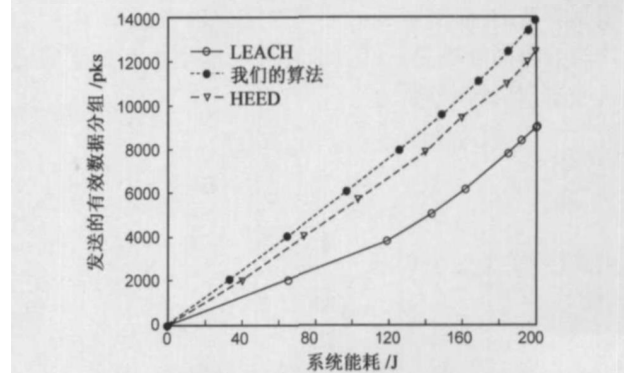


图 6 系统能耗与传递的有效分组数的关系

最后, 我们比较了算法的能耗效率 (energy efficiency). 图 6 给出系统能耗与传递的有效数据分组数的关系. 可以看出在耗能相同的情况下, 我们的算法与其他两种算法相比能够传递更多的有效数据分组, 这说明我们的算法具有更高的能耗效率.

4 结论

分簇算法的研究是无线传感器网络中的一个基本问题. LEACH 作为一个代表性的分簇算法在资源受限的无线传感器网络得到了广泛的应用. 但是 LEACH 算法的随机性会破坏系统的负载均衡, 导致某些节点过

快地耗尽电池能量而“死亡”。我们提出了一种基于自适应定时器策略的分布式分簇算法,有效延长了系统生命并且提高了能耗效率。我们将来的工作是通过求解算法参数的最优值来进一步提高算法的性能。

参考文献:

- [1] Akyildiz I F, Su W, Sankarasubramaniam Y. A survey on sensor networks[J]. IEEE Communications Magazine, 2002, 8(2): 102-114.
- [2] 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.
- [3] Hou T C, Tsai T J. An access based clustering protocol for multihop wireless ad hoc networks[J]. IEEE Journal of Selected Areas in Communications, 2001, 19(7): 1201-1210.
- [4] Mhatre V, Rosenberg C. Design guideline for wireless sensor networks: communication, clustering and aggregation[J]. Ad Hoc Networks Journal, 2004, 1(2): 45-63.
- [5] O Younis, S Fahmy. Distributed clustering in ad hoc sensor networks: a hybrid, energy efficient approach[A]. Proceedings of IEEE INFOCOM 2004[C]. Hong Kong: IEEE press, 2004. 629-640.
- [6] Zhao L, Hong X, Liang Q. Energy-efficient self organization for wireless sensor networks: A fully distributed approach[A]. Proceedings of IEEE GLOBECOM 2004[C]. Boston: IEEE press, 2004. 1345-1351.

作者简介:



曹涌涛 男, 1975 年 11 月生于江苏省盐城市, 上海交通大学电子工程系博士研究生, 1997 年毕业于南京通信工程学院, 获工学学士, 2000 年毕业于解放军理工大学通信工程学院, 获工学硕士。目前的主要研究方向为移动计算、Ad hoc 网络与无线传感器网络。

E-mail: ytao@sjtu.edu.cn



何 晨 男, 1952 年 5 月生于江苏省苏州市, 上海交通大学教授, 博士生导师, 上海交通大学现代通信研究所副所长, 1982 年毕业于南京工学院无线电系, 获工学学士, 1985 年毕业于南京工学院通信与电子系统专业, 获工学硕士。1994 年毕业于日本国立德岛大学研究生院通信与电子系统专业, 获工学博士。目前的主要研究方向为新一代无线通信系统理论、智能信息处理

以及自适应信号处理在通信中的应用、信息论与编码理论等。

E-mail: chenhe@sjtu.edu.cn



蒋铃鸽 女, 1959 年 9 月生于江苏省南京市, 1982 年 1 月毕业于东南大学无线电工程系无线电技术专业, 1993 年 9 月和 1996 年 9 月分别获得日本国立德岛大学电子系统工学硕士和工学博士学位; 现为上海交通大学电子工程系教授, 博士生导师。主要研究领域为无线通信系统中的智能信息处理、移动 IP 技术、数字水印及混沌理论在现代通信中的应用等。

E-mail: lgjiang@sjtu.edu.cn