

交替活跃的部分连接网络休眠调度策略研究^{*}

张天乐^{1,2}, 李忠诚¹, 殷昭印^{1,2}

(1. 中国科学院 计算技术研究所, 北京 100080; 2. 中国科学院 研究生院, 北京 100049)

摘要: 提出基于交替活跃的部分连接网络的休眠调度策略, 无须进行拓扑控制, 支持较高的休眠/活跃比率, 显著提高了网络平均寿命。给出了满足延迟要求的休眠时间表的确定方法。提出簇内休眠相位同步和簇间相位次序调整的主动调节方法。仿真表明, 该方法显著降低了端到端的延迟。

关键词: 部分连接网络; 休眠调度; 延迟; 仿真

中图分类号: TP393 文献标志码: A 文章编号: 1001-3695(2007)07-0270-03

Sleep Scheduling for Partially Connected Network in Alternately Active Mode

ZHANG Tian-le^{1,2}, LI Zhong-cheng¹, YIN Zhao-yin^{1,2}

(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China; 2. Graduate School, Chinese Academy of Sciences Beijing 100049, China)

Abstract: A scheme of sleep scheduling based on partially connected network in alternately active mode was proposed. Without need of topology control, the method had high sleep/wake ratio thereby significantly extending the lifetime of network. A method was given to set the sleep-wake cycles according to maintain the responsiveness. The proposed sleep scheduling synchronizes the phase and cycle of sleep within clusters while differentiates phase among clusters. Simulation shows that these methods can significantly reduce the end-to-end delay.

Key words: partially connected networks; sleep scheduling; delay; simulation

无线 Ad hoc 网络具有灵活部署的优势, 拥有巨大的应用潜力。组成网络的节点通常受到电源供给和硬件资源的限制, 但大量的节点协同工作, 却可以提供丰富的功能。因此在实际的 Ad hoc 网络应用中, 需要在节点(个体)的资源限制和网络(群体)的性能之间作出折中。尽管无线接口的硬件低功耗设计有了长足发展, 通信功耗仍然占功耗中的最大比例^[1]。在满足性能要求的前提下, 降低无线通信功耗是无线 Ad hoc 网络面临的巨大挑战。

最直接的节省通信功耗的方法就是对节点的通信状态进行调度。由于无线 Ad hoc 网络中, 发送、接收和侦听消耗的功耗相差不大, 只有彻底关闭无线收发电路, 才具有较好的节能效果。通过节点调度, 让节点交替地处于活跃和休眠状态的交替活跃模式是节能的有效方法。但是较低的活跃程度会造成丢包和响应速度降低, 甚至成为非连通的网络。

由于目前大多数的 Ad hoc 路由基于完全连接的拓扑假设, 当网络是非连通时无法工作。为避免节点交替活跃造成的连通性破坏, 同时避免部分节点能量过度支出, 休眠调度时, 需要进行拓扑控制。节能协同技术是一种有效的节点调度方法, 坚持完全连接的假设, 构造连通支配集。其中的节点保持活跃, 形成完全连接的骨干网。但是, 为了保证网络的完全连接, 休眠节点的数量和休眠时间均受到限制, 影响了节能效果, 而且拓扑控制的算法也会造成额外开销。

另外一种可行的方法就是抛弃完全连接的假设, 节点按照独立的休眠—活跃时间表进行交替切换, 所有节点可以得到公

平的休眠机会。其控制简单, 节省了调度的能量消耗。同时, 由于不需要保证网络的完全连接, 允许更多的休眠节点数量和更长的休眠时间, 网络整体的寿命得到延长。但是大量的节点随机地处于交替活跃模式, 会使网络的连通性无法保证, 成为部分连接的网络。为此, 本文提出能够适应部分连接的网络异步路由转发方式。当一跳休眠时, 包会在本节点延迟等待; 当一跳恢复活跃时, 继续转发。在连通性无法保证的部分连接网络中, 也能维持系统的性能。该方法虽然引入了一定的延迟, 但是由于无须拓扑控制, 允许更高程度节点休眠, 适用于强调节能而对实时性要求不高的领域, 如土壤监测、生态监测、气象监测等非实时的应用。

1 相关研究

能量问题是无线 Ad hoc 关心的重要问题。降低通信功耗是主要途径。目前对于通信产生的功耗分析的研究中, 建立了很多无线通信的能量模型。节点在不同状态下的能量消耗具有差异。文献[2]给出了传感器网络射频电路 RFM 的功耗模型, 接收、发送和空闲的功率在 9~12 mW, 休眠消耗功率仅为 0.016 mW。文献[3]给出了 802.11 无线网卡的功耗模型, 接收、发送和空闲的消耗功率分别为 1 000 mW、1 400 mW、830 mW, 休眠状态的功率为 130 mW。可见, 即使在空闲状态, 能量的消耗也相当大。有效控制通信能量开销就是要让尽量多的节点在尽可能多的时间内处于休眠状态。

通过精心设计的算法自动选择一部分节点保持活跃, 而让另一部分节点休眠是有效的途径。文献[4]给出了节点协同算法, 构造最小连通支配集; 这一部分关键节点组成连通骨干, 完成路由转发, 而让其他节点休眠, 既缩小了路由查找范围, 又能够显著降低能量消耗。但是过少的关键节点会降低网络的响应速度和性能。文献[5]给出了扩大的连通支配集生成算法, 可以在不降低性能的情况下让网络节点交替休眠, 比较适合节点密度较大的网络。为了避免选择的活跃节点一直工作, 节点需要交替休眠, 每次交替均需要重新生成支配集, 造成较大开销。

另外一种休眠方式就是每个节点维护一个活跃—休眠时间表, 所有节点都均等地进行自主休眠。AFECA 方法^[6]采用节点随机休眠的策略, 休眠时间与邻居节点的数量呈正相关。Stemm 和 Katz 提出了利用应用层的指示信息确定节点的时间表^[7]。但是自主的分布式休眠调度会使网络的连通性被破坏, 对于需要完全连接拓扑的同步方式路由是十分不利的。Ramanathan 和 Rosales-Hain 等人提出调节发送功率的方法维持连通性^[7], 拓扑控制算法会受到网络的支持限制。而文献[8]则放弃了完全连接的假设, 让更多的节点休眠, 不需要维持网络的连通性; 虽然能够显著降低能量开销, 但是会造成延迟的增加。

采用延迟转发的异步方式进行通信, 不对网络的连通性有要求, 无须拓扑控制, 允许更多节点休眠, 延长了网络寿命。对异步通信方式的延迟进行估计, 依据计算结果, 设计了主动和被动方式的休眠调度算法, 显著降低了延迟。

2 交替活跃网络延迟估计

2.1 交替活跃网络模型

在二维平面区域中的 Ad hoc 网络可以表示为一个元组 $M = (N, P, S)$ 。其中 N 为节点的集合, $|N| = n$; P 为节点的位置函数: $N \times T \rightarrow (0, l)^2 (l > 0)$; S 为节点的状态函数: $N \times T \rightarrow A$ 。其中, 指示变量 $A = \{0, 1\}$, 0 表示节点休眠, 1 表示活跃。特殊地, 对于只考虑交替活跃特性的网络, 位置函数表示为 $P: N \rightarrow (0, l)^2$, 网络表示为 $MF = (N, P, S)$ 。

用 ON/OFF 模型表示节点的交替活跃特性: 在 ON 阶段, 正常通信, 处于活跃状态; 在 OFF 阶段, 节点关闭通信, 处于休眠状态。初始为 ON 状态, 当有数据时保持 ON 状态。假设 ON 事件和 OFF 事件分别为强度为 μ 和 λ 的泊松输入流。

2.2 端到端延迟时间的分布函数

采用异步延迟转发的过程如下: 当下一跳节点处于 OFF 状态时, 不会丢弃包, 而是将其缓存在缓冲区中, 并等待下一跳的出现。当包经过一个节点, 若该节点的下一跳已经处于 ON, 则无须等待, 直接转发; 若下一跳为 OFF, 则等待其变为 ON 时转发。当节点重复该过程, 包到达目的节点, 记做一次成功转发事件。该过程总共用时 T , 记做包异步转发的端到端延迟。

假设整个转发过程平均路径长度为 n 。其中平均经过 i 个已经处于 ON 状态的节点, 而需要等待 $n - i$ 个处于 OFF 状态的节点跳变为 ON 状态。设互不相容事件 ON_i 为: 在路径上转发过程中, 平均有 i 个节点已经处于 ON 状态。设 T 时间内, 成功转发事件为 PP; 事件 $PP | ON_i$ 为直接通过了 i 个已经处于 ON 状态的节点条件下成功转发事件, 即包还需再等待 $n - i$ 个

OFF 节点出现。包在某个节点等待下一跳由 OFF 变为 ON 状态, 可以看做是等待一个强度为 μ 的泊松输入流上的一个顾客到达, 则等待 n 个处于 OFF 状态的节点可等价看做是在同一个泊松输入流上等待 n 个顾客到达。其概率为 $1 - \sum_{k=0}^{n-1} p_k(T)$ 。其中, $p_k(T) = (T\mu)^k / k! e^{-T\mu}$ 。因此, 由全概率计算得到 T 时间内包能成功到达的概率为

$$Ap_n(T) = P(PP) = \sum_{i=0}^n P(ON_i) \times P(PP | ON_i) = \sum_{i=0}^n \{ C_n^i \times i! (1 + \lambda)^{-i} \} \times (1 - \sum_{k=0}^{n-i-1} p_k(T)) \quad (1)$$

其中, $\lambda = \mu / \rho$ 。

假设包按照异步延迟转发的过程到达目的节点的端到端延迟为随机变量 T 。记其分布函数为 $F(t) = P(T \leq t) = 1 - P(T > t) = 1 - P_0(t)$ 。其中, $P_0(t)$ 表示在 t 时间内包未能成功转发的概率, 显然 $P_0(t) = 1 - Ap_n(t)$, 因此 $F(t) = Ap_n(t)$ 。例如当 $\mu = 1/10$, $\lambda = 1/100$ 时, $n = 13$, 延迟 T 的分布函数如图 1 所示。经过拟合后 $F(t)$ 近似看做 $1 - e^{-0.085t}$, T 的平均值为 11.7 s。当网络的时间表确定后, 可以通过式(1)估计网络的平均端到端延迟。相反地, 也可以按照应用的延迟需求设计节点的休眠活跃时间表。

2.3 端到端延迟时间的均值

数据包在节点等待下一跳出现, 等待时间设为 T_i , 则端到端延迟为 $\sum_{k=1}^n T_i$ 。当包到达某个节点后, 在路由选择时, 下一跳可能已经处于 ON 状态(概率为 $\lambda / (\lambda + \mu)$), 或者处于 OFF 状态(概率为 $\mu / (\lambda + \mu)$)。前者可以直接转发, 后者则需要等待其变为 ON 状态。由 2.2 节中所述, 等待节点由 OFF 到 ON 的跳变为一个等待按照泊松过程顾客到达的事件, 等待平均时间为 μ^{-1} 。因此, 平均每跳等待时间为 $1 / [\mu(1 + \lambda/\mu)]$, 端到端总延迟为

$$n / [\mu(1 + \lambda/\mu)] \quad (2)$$

当 $\mu = 1/10$, $\lambda = 1/100$ 时, $n = 13$ 时, 总延迟为 11.82 s, 与 2.2 节中按照延迟分布函数计算结果基本吻合。

根据端到端的延迟分布函数, 可以估计在容忍的延迟时间内获得的包递交率, 也可以按照应用的递交率的性能需求, 推算延迟等待的时间; 根据端到端的平均延迟的计算可以得到节点按照当前休眠时间表进行调度, 也可以获得系统的响应时间。这两个计算结果对强调整能休眠的 Ad hoc 网络的节点调度具有参考意义和指导作用, 也是本文休眠调度方法的理论依据。

3 休眠—活跃调度的方法

与协同调度方法不同的是, 在交替活跃网络中不需要拓扑控制让活跃节点构成路由骨干网, 而是让节点平等地按照各自的时间表进行休眠—活跃交替。休眠活跃调度的主要问题就是确定和调整节点的时间表。

时间表的调整有两种方式, 即主动和被动。在主动方式中, 节点通过局部信息自适应地调整时间表, 使得邻近节点的时间表尽量重合, 即同时活跃或同时休眠; 而距离较远的节点休眠的交替相位按照一定的顺序, 如包转发方向依次滞后。这种局部同步、全局排序的方法, 能够有效地适应异步延迟转发方式。

通过分簇算法将网络分为若干邻接区域; 簇首领周期性地广播时间表以同步区域内部节点的时间表。其中包含一个簇

首领时间戳,表示其已经保持活跃的时间。簇内节点收到后,会根据自己已经活跃的时间与该时间戳进行比较,并使具有相同的活跃交替相位。簇内时间表同步可以使节点同时活跃的概率增加。

证明:假设路径长度为 n , 路径穿过 $k(k < n)$ 个簇。由于簇内节点同时启动关闭,等待同一簇内节点出现等效于等待一个节点,从而 AP 的计算公式中相当于 n 减少为 k 。由式(1)得到,对于同样的 T , AP 会增大, T 的分布函数增长快,则 T 的均值降低。从图2可以看出簇内调整后, AP 曲线左移, T 均值降低。经过簇内调整后,长度为 n 的路径延迟接近于长度为 k 的路径延迟。从式(2)看出,延迟降低 k/n 倍。

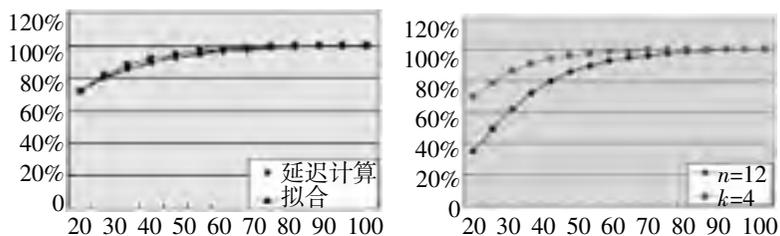


图1 延迟 T 的分布函数 图2 簇内调整前后延迟分布函数

另外,由于采用异步转发方式,应该尽量让节点的休眠次序符合转发次序。对于 k 个簇,如果按照各自的休眠表调度,则相当于路径长度为 k 的转发过程,总延迟为 $k[\mu(1 + \dots)]$ 。如果将 k 个簇的休眠相位均匀地分布于 $1/\mu$ 长度内,则总延迟为 $1[\mu(1 + \dots)]$ 。可见,簇间的休眠时间表相位按照转发次序进行调整,可以改善网络的延迟时间。

对于簇间调整过程如下:簇 A 首领周期性地广播时间表,当簇 B 的节点收到时,查找路由表。如果 A 包含在 B 的路由中,且处于上游链路,则使自己的相位滞后于簇 A 的相位,否则相位超前于簇 A 的相位,滞后的幅度为簇 A 和簇 B 平均休眠时间 $(1/\mu)$ 的 $1/k$; 如果不在路由表中,或者存在两条路由条目且 A、B 互为上游链路,则不调整。经过调整后,在包的转发方向上,每跳的平均等待延迟从 $1/\mu$ 降低为 $1/(k \times \mu)$,端到端延迟显著降低。

被动调度方式的调节方法目的是使网络在不同情况下呈现不同的活跃特性。网络的应用层对响应的需求会随时变化(如平时和战时区别、通常时期和应急时期的区别),节点的休眠表需要进行相应的调整。例如在应急情况时,网络需要具有较好的响应速度,网络节点的活跃程度应当提高;而对于平时情况,网络可以延长休眠。当应用层对于性能和响应时间的需求发生变化,被动调节方法可以动态修改节点休眠—活跃时间表,以调节网络性能和时间特性。根据第2章的计算结果,可以通过调节 μ 和 \dots 达到调节延迟时间和递交率的目的。另外,被动调度还包括依据应用对网络的延迟时间要求,在同步转发方式和异步转发方式之间动态选择。当应用需要较高实时性时,节点根据延迟计算得到较高的 μ 。这时,会触发被动调度过程,选择同步转发方式;反之,采用异步转发方式。

4 仿真和结果分析

在仿真中,考察场景:100个节点随机抛撒均匀分布于 $10 \text{ km} \times 10 \text{ km}$ 的平面区域,所有节点均按照交替活跃模型以间歇性方式工作。ON 和 OFF 状态到达时间间隔服从负指数分布,参数分别为 $\mu = 1/10$ 和 $\dots = 1/100$ 。位于区域上方的源节点 node_0 ~ node_5 的应用层的 CBR 业务源,分别向位于区域下方的 node_94 ~ node_99 发送数据包,发送速率为 2 packet/s,

包到达容忍延迟时间为 T , 发包持续时间为 2 h, 平均包长度 500 Bytes, 平均路径长度为 13 跳。

当所有节点按初始化的时间表进行交替活跃的工作,平均延迟为 12.11 s (计算估计值为 11.82 s)。当进行簇划分(簇直径为 2 跳),平均每条路径经过 5.5 个簇,经过簇内调节后,平均路径长度降低为 5.5 跳。根据计算模型得知,平均延迟计算估计值为 5 s, 仿真结果如图3所示。稳态后平均为 5.2 s。当启动簇间调节过程后,理论延迟 k 降低 5.5 倍。仿真结果表明延迟降低 3.8 倍。

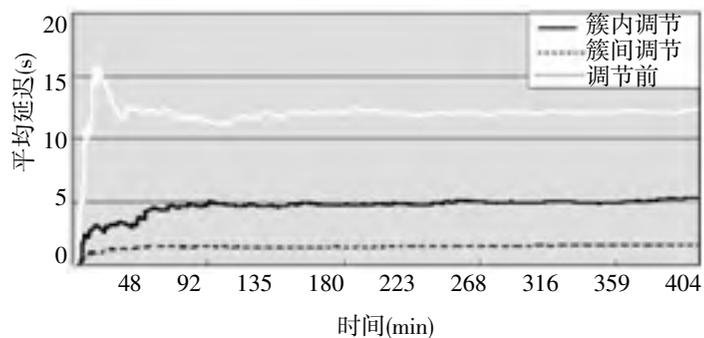


图3 节点调度对延迟的改善

5 结束语

对交替活跃模式网络的延迟进行建模,给出延迟的分布函数和平均延迟的计算方法,为节点调度策略提供理论依据和方法指导。采用了主动调度和被动调度两种节点调度方法对休眠—活跃时间表进行动态调整,降低了端到端延迟。理论分析和仿真结果表明,本文提出的节点调度方法能够有效适应交替活跃工作模式的部分连接网络,能够显著改善网络的延迟时间,并能随应用需求的变化自动选择最佳的交替活跃模式和路由方式,具有良好的适应性和实用性。

参考文献:

- [1] HIH E, CHO S H, ICKES N, *et al.* Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks [C] // ACM MOBICOM. [S. l.] : [s. n.], 2001: 272-287.
- [2] FEENEY L M. An energy-consumption model for performance analysis of routing protocols for mobile Ad hoc networks [J]. Mobile Networks and Applications, 2001, 3(6) : 239-249.
- [3] FEENEY L, NILSSON M. Investigating the energy consumption of a wireless network interface in an Ad hoc networking environment [C] // Proc of IEEE INFORCOM. Anchorage, AK: [s. n.], 2001.
- [4] DAS B, BHARGHAVAN V. Routing in Ad hoc networks using minimum connected dominating sets [C] // IEEE International Conference on ICC. [S. l.] : [s. n.], 1997: 376-380.
- [5] CHEN Benjie, JAMIESON K, BALAKRISHNAN H, *et al.* Span: an energy-efficient coordination algorithm for topology maintenance in Ad hoc wireless networks [J]. ACM Wireless Networks Journal, 2002, 8(5) : 481-494.
- [6] XU Ya, HEIDEMANN J, ESTRIN D. Adaptive energy-conserving routing for multihop Ad hoc networks, Research Report 527 [R]. [S. l.] : USC/Information Sciences Institute, 2000.
- [7] RAMANATHAN R, ROSALES-HAIN R. Topology control of multihop wireless networks using transmit power adjustment [C] // Proc of the IEEE Conference on Computer Communications (INFOCOM). Tel Aviv, Israel: [s. n.], 2000: 404-413.
- [8] SCHURGERS C, TSIATIS V, GANERIWAL S, *et al.* Topology management for sensor networks [C] // ACM MOBIHOC. [S. l.] : [s. n.], 2002: 135-145.