

MANET 分簇节点组通信功能的设计与实现

李冰¹, 金志刚¹, 张明阳^{1,2}

(1. 天津大学计算机科学与技术学院, 天津 300072; 2. 华东计算技术研究所, 上海 200233)

摘要: 提出一种将源路由(DSR)协议与自适应按需加权(AOW)分簇算法相结合的 Ad hoc 网络体系结构设计方案。利用 AOW 算法进行分簇, 将簇半径严格控制在 1 跳以内, 簇内成员无需路由功能, 大大减少了网络路由信息的冗余。簇头运行 DSR 协议充当路由转发器, 使不同簇节点之间的通信依赖于各簇的簇头。模拟结果显示该方案具有良好的性能, 并已在 Windows XP 中实现。

关键词: 分簇算法; 动态源路由协议; Ad hoc 网络; 自适应按需加权算法

Design and Implementation of Group Communication Function for Clustering Nodes in MANET

LI Bing¹, JIN Zhi-gang¹, ZHANG Ming-yang^{1,2}

(1. School of Computer Science & Technology, Tianjin University, Tianjin 300072;

2. East China Institute of Computer Technology, Shanghai 200233)

【Abstract】 This paper proposes a scheme of Ad hoc network architecture which combines Dynamic Source Routing (DSR) protocol and Adaptive On-demand Weighted(AOW) clustering algorithm. Using AOW algorithm, the cluster radius is strictly controlled within a hop and cluster-members do not need routing function, greatly reducing the redundancy of routing information. Each cluster-header implements the DSR Protocol, working as a router. The communication between nodes coming from different clusters relies on cluster-headers. Simulation result shows that the scheme has good performance. It has been implemented in Windows XP operating system.

【Key words】 clustering algorithm; Dynamic Source Routing(DSR) protocol; Ad hoc network; Adaptive On-demand Weighted(AOW) algorithm

1 概述

随着移动技术的飞速发展,人们已不再满足于有线网络,而希望可以在更自由的环境下使用网络;同时,在某些特殊场合,仅依靠传统的基于基站的无线网络无法满足人们的需求,例如,战场上部队快速地展开和推进,在临时地点召开共享桌面的办公会议,这些场合的通信不能依赖于任何预先架设的网络设施。为了能够在没有固定基站和有线网络的地方进行通信,Ad hoc 技术应运而生。但目前这种技术还不成熟,需要优化之处比较多。合适的分簇算法和高效的路由协议是其中 2 个研究方向。

Ad hoc 网络的体系结构可分为平面结构和分级结构^[1] 2 种。在平面结构中,网络较健壮,但可扩展性较差。利用分级结构可以很好地解决平面结构中存在的问题。在分级结构中,通常将整个 Ad hoc 网络进行分簇。经典的分簇算法有最小 ID 分簇算法^[1]、最高节点度分簇算法^[1]、最低节点移动性分簇算法^[2]等。但它们仅考虑了某一方面因素对网络的影响,没有考虑系统的负载平衡或系统的能量消耗。

Ad hoc 网络的路由协议大致可以分为先验式路由协议和反应式路由协议^[3]。目的序列距离矢量(Destination Sequenced Distance Vector, DSDV)路由协议^[4]是目前应用最广泛的先验式路由协议。每个节点维护一个路由表,周期性地与邻节点交换路由信息。但路由维护将产生较大的冗余。Ad hoc 按需距离矢量(Ad hoc On-demand Distance Vector, AODV)路由协议^[5]是 DSDV 的改进,但它是反应式路由。为了找到通往目的节点的路由,源端将广播路由请求分组,邻居节点依次广播此分组,直到它被送到目的节点或者缓存了到目的节点路由

的中间节点。但洪泛方式的路由请求可能使相邻节点之间发生传播冲突和重复广播。

动态源路由(Dynamic Source Routing, DSR)协议^[6]包含路由发现和路由维护 2 部分。源路由使得数据包可以根据已经获得的路径进行发送,中间节点只需根据携带在源路由包中的下一跳目的节点进行转发。同时,利用侦听到的路由信息,中间节点可以减少新的路由发现过程,大大提高性能。由于获得源路由后,通过比较可以很快发现重复节点,因此有效地避免了路由循环。

本文综合 Ad hoc 网络技术以及网络分簇技术,提出了一种将 DSR 源路由协议与自适应加权(Adaptive On-demand Weighted, AOW)分簇算法相结合的组通信方案,并取得了较好的实验结果。

2 DSR 协议与 AOW 算法相结合的设计方案

2.1 DSR 协议与 AOW 算法相结合的设计构架

根据节点所处地位的不同,本文分别采用平面结构和分簇结构进行设计。

处于同簇的节点间的关系是平面结构,即所有节点是平等的。即使是簇头节点,对于簇内节点而言,它的簇头属性也是透明的。每个簇的半径严格限制在 1 跳以内,保证了簇内节点完全可以通过点到点的方式实现通信,簇内节点(除了

基金项目: 国家自然科学基金资助项目(90604013);天津市应用基础及前沿技术研究计划基金资助项目(08JCYBJC14200)

作者简介: 李冰(1982-),女,博士研究生,主研方向:计算机网络及其安全;金志刚,教授、博士生导师;张明阳,硕士研究生

收稿日期: 2007-11-20 **E-mail:** libingce19@yahoo.com.cn

簇头外)不需要具有路由功能,从而大大减少了网络路由信息的冗余。

处于不同簇的节点间的关系是分簇结构,即不同簇的节点之间的通信依赖于各簇的簇头。每个簇的簇头要实现 DSR 协议,充当路由转发器,更新和保持整个 Ad hoc 网络的拓扑结构。由于采用了分簇结构,因此在保证各个簇的半径为 1(即可以在簇内进行点到点通信)的同时,又为节点的组通信提供了可扩展的功能。理论上,像这样由平面结构和分簇结构结合而成的组通信模型具有无限的扩展性。

在本文设计中,簇头在组织一个簇时用由该簇头设置的 SSID(Service Set Identifier)来标识簇,即簇内的节点都通过将自己的 SSID 设置成为簇头所规定的 SSID 来加入簇。不同簇之间的 SSID 不同,因此,不同簇之间的簇内节点不能直接通信,只能通过簇头转发数据。用 SSID 来标识一个簇可以使该设计方案与有固定设施的无线网络保持兼容,因为无线接入点也是通过 SSID 来标识一个子网的。

2.2 平面结构中节点的通信

在同一个簇内处于平面结构的节点可以进行点到点的通信。本文将路由由仅仅限制在簇间,所以,簇半径严格控制在 1 跳以内,每个节点都由簇头分配一个 IP,整个簇形成一个子网,簇内节点的通信都在子网内部进行。簇间的通信要经簇头中转,簇内和簇间的区别对于簇内节点(除去簇头)而言透明。

簇内节点在通信时,首先扫描自身的网卡,得到一个自身网卡列表。然后扫描周围可见范围内的 SSID,得到一个 SSID 的列表。若 SSID 列表为空,则该无线节点所在范围内没有任何簇存在,该节点将自己成为簇头,开始侦听其他节点的加入。若 SSID 列表不为空,则该无线节点遍历 SSID 列表。对于列表中的每个值,无线节点都把自己的 SSID 设置成与之相同的值,然后广播一个 IP 请求包。若该 SSID 对应的无线节点支持分簇组通信,则返回一个 IP 应答包。请求 IP 的节点在收到这个应答后,将自身的 IP 设置为对应值,并开启 2 个线程,一个是用来监听其他节点发送来的请求包,另一个线程周期性地广播包给簇内其他节点,从而周期性地更新簇内节点列表和共享资源列表。

2.3 分簇结构的设计

2.3.1 AOW 算法

AOW 算法的提出是为了改善分簇结构网络的总体性能,最大限度地发挥分簇结构的优势。它认为簇头的选举应考虑多种因素,并根据实际需要和应用环境做出合理的折中。该算法为每个节点分配一个权值 $Weight$,用于表示节点适合充当簇头的程度,权值越小,节点越适合充当簇头。权值的计算如下:

$$Weight = a \times mobility + b \times degree + c \times power + d \times energy + x$$

其中, $mobility$ 表示节点移动性; $degree$ 表示节点度; $power$ 表示节点传输功率; $energy$ 表示节点剩余能量; x 表示其他可能影响权值的因素,如处理能力和存储空间;参数 a, b, c 和 d 为权重因子,由具体的应用和网络环境决定,并进行归一化。

2.3.2 分簇结构的设计

使用 AOW 算法需要综合考虑 3 种因素(节点的电池剩余量、节点的移动性、节点的度数)以进行簇头的选举。可以通过调节权重因子改变不同因素的重要性以适应不同的系统环境和应用需求。在分簇结构中,不同簇节点之间的通信依赖

于簇头。簇头首先要担当一个簇内所有节点的 DHCP 服务器。簇内节点的 IP 由簇头统一分配。每个簇头中运行 DSR 协议,充当路由转发器,通过它的按需驱动机制,尽量减少网络信息冗余。

簇头的选举采用按需自适应地选择,而非周期性地。簇结构的维护是基于消息/事件驱动的。簇头可以工作在双功率模式下,使用较大的功率进行簇间通信,使用较小的功率进行簇内通信。

3 设计方案的实现

本文使用 NS-2 对上述方案进行了模拟实现。在实现过程中,面临的主要技术问题就是对分簇算法中权值的计算。考虑了 3 个因素:移动性,节点度和剩余能量。

(1)移动性的计算

在算法的实现中,采用相对移动性指标来表示节点的移动性:节点通过比较收到的来自某一邻居节点连续 2 次的信号强度来估计它们之间的相对移动性。节点每收到一次邻居信息报文,除了记录邻居信息外,还记录收到时的接收功率。在邻居信息表中记录连续 2 次的接收功率就可以近似计算出节点的相对移动性。

(2)节点度的计算

邻居节点之间通过信息交互了解对方,节点定期向其邻居节点广播邻居信息报文,并根据收到的邻居信息报文实时更新自己的邻居信息表。可以通过邻居信息表的长度计算节点度。

(3)剩余能量的计算

NS 中集成了节点的能量模型,可以在模拟开始时设置节点的初始能量、发送功率、接收功率等,节点能量模型会根据发送数据包、接收数据包中消耗的能量自动计算节点的剩余能量,本算法在实现中直接获取节点能量模型中的剩余能量值。

(4)权值的计算

$$weight = a \times mobility + b \times fabs(degree - AOW_DIDEAL) + c \times (INIT_ENERGY - energy) / INIT_ENERGY$$

其中, a, b, c 为权重因子,且 $a + b + c = 1$; AOW_DIDEAL 为理想节点度 D_{ideal} ; $INIT_ENERGY$ 为节点的初始能量。

4 方案的测试和性能评价

簇头数和统治集更新的次数是分簇 Ad hoc 网络中 2 项重要的指标,本文通过实验检验了算法的性能。图 1、图 2 为实验结果。

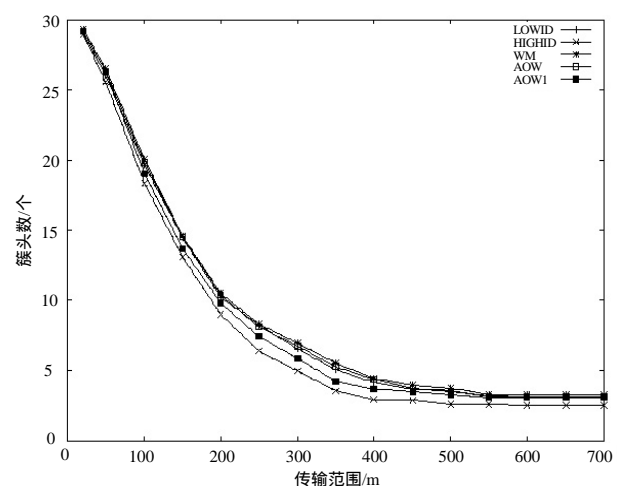


图 1 簇头数随传输范围的变化

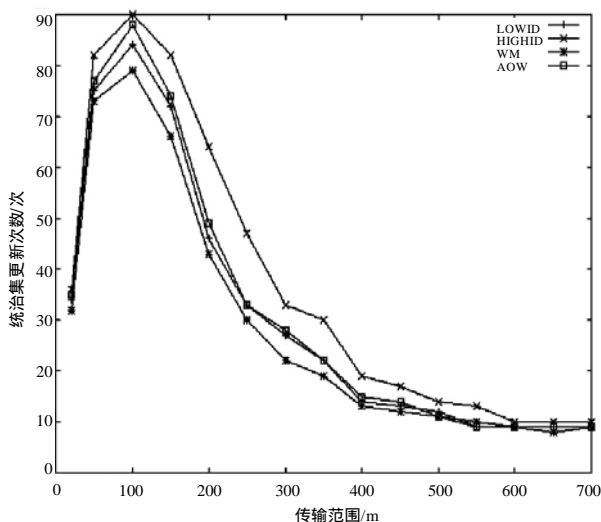


图2 统治集更新次数随传输范围的变化

实验场景是由 NS-2 中的 setdest 随机生成的,大小为 1 000 m×1 000 m,节点数为 30,节点的最大速度为 15 m/s。为了对比实验效果,设置了 2 组 AOW 参数。其中,LOWID 表示最小节点度算法;HIGHD 表示最高节点度算法;WM 表示最低节点移动性算法。AOW 中的参数为 $a=0.6$, $b=0.2$, $c=0.2$, $D_{ideal}=5$;AOW1 中的参数为 $a=0.2$, $b=0.6$, $c=0.2$, $D_{ideal}=10$,其中, a , b , c 分别为移动性、节点度、剩余能量的权重因子; D_{ideal} 为理想节点度。

网络中的簇头数不宜过多或过少,否则都将使分簇的意义减弱或消失,在极端情况下退化成平面的网络结构。由图 1 可以看出,HIGHD 中的簇头数最少,WM 中的簇头数在多数情况下是最多的,而其他算法介于 HIGHD 与 WM 之间。在 AOW 和 AOW1 中,权重因子与理想节点度起到了调节簇头数的作用,由于 AOW1 中节点度的权重因子较大,而且理想节点度较高,因此簇头数较 AOW 少。在具体应用中,可以根据实际情况调节 AOW 的参数。

由图 2 可以看出,统治集更新的次数随着节点传输范围的增大先急剧增加,然后急剧减少,到 300 m 左右开始缓慢减少,最后趋于一个固定值。WM 的统治集更新次数最少,HIGHD 最多,而 AOW 与 LOWID 介于两者之间。HIGHD 中的统治集更新次数较多是因为 HIGHD 中簇头的选举只考虑

节点度,而节点的移动使得簇头的节点度处在变化中,2 个簇头也会成为相邻节点,从而导致了簇结构的变化。WM 中统治集更新次数较少是因为 WM 中簇头相对簇成员的移动性较小,簇结构变化较缓慢。AOW 算法介于两者之间,说明 AOW 能在节点的移动性和节点度之间作很好的折中。

另外,当把传输范围设置在 300 m 附近时,根据实际需要改变 AOW 算法的权值,得到的簇头数在 5~7 之间。此时 30 个节点组成的网络中,统治集的更新次数也较少。而使用 DSR 协议维护 5~7 个簇头的路由,网络开销比较经济。

在确定本方案有较好的性能后,将其在 Windows XP 系统中实现,并成功移植到 WinCE 模拟器中,程序的移植性良好。

5 结束语

本文提出了一种将源路由协议与自适应加权分簇算法相结合的设计方案。实验结果表明,DSR 路由算法能够满足簇头节点路由转发功能的需要;AOW 分簇算法在仿真中比其他典型分簇算法更具优越性,并且通过调节解权重的分配,实现了灵活性。

参考文献

- [1] 王海涛. Ad hoc 网络的体系结构和分簇算法研究[D]. 南京: 解放军理工大学通信工程学院, 2003.
- [2] Basagni S. Distributed Clustering for Ad hoc Networks[C]//Proc. of International Symposium on Parallel Architectures, Algorithms and Networks. Washington, USA: IEEE Computer Society, 1999.
- [3] 龚月荣, 林晓明. 移动自组网反应式路由协议在 Linux 中的实现[J]. 计算机工程, 2004, 30(7): 98-100.
- [4] Vahid G. Analysis of Network Traffic in Ad hoc Networks Based on DSDV Protocol with Emphasis on Mobility and Communication Patterns[C]//Proc. of the 1st IEEE and IFIP International Conference in Central Asia on Internet. [S. l.]: IEEE Press, 2005.
- [5] Perkins C E, Royer E M. Ad hoc On-demand Distance Vector Routing[C]//Proc. of the 2nd IEEE Workshops on Mobile Computing Systems and Applications. [S. l.]: IEEE Press, 1999.
- [6] Li Jie, Pan Yi, Yang Xiao. Performance Study of Multiple Route Dynamic Source Routing Protocols for Mobile Ad hoc Networks[J]. Journal of Parallel and Distributed Computing, 2005, 65(2): 169-177.

(上接第 93 页)

参考文献

- [1] 王 林, 戴冠中. 基于复杂网络社区结构的论坛热点主题发现[J]. 计算机工程, 2008, 34(11): 214-216.
- [2] Wu Fang, Huberman B A. Finding Communities in Linear Time: A Physics Approach[J]. The European Physics Journal B, 2004, 38(2): 331-338.
- [3] Newman M E J, Girvan M. Finding and Evaluating Community Structure in Networks[EB/OL]. (2006-10-23). <http://link.aps.org/abstract/PRE/v69/e026113>.
- [4] 岳 训, 迟忠先, 莫宏伟, 等. 基于网络社区模块结构的特征选择性能评价[J]. 计算机工程, 2007, 33(12): 16-18.
- [5] Cohen W W. Enron Email Dataset[EB/OL]. (2005-04-04). <http://www.cs.cmu.edu/~7Eenron/>.
- [6] Sago Networks Data Center. The Prefuse Visual Toolkit[EB/OL]. (2007-08-23). <http://prefuse.org/>.
- [7] Girvan M, Newman M E J. Community Structure in Social and Biological Networks[J]. PNAS, 2002, 99(12): 7821-7826.
- [8] Bagrow J P, Bollt E M. A Local Method for Detecting Communities[EB/OL]. (2007-06-25). <http://link.aps.org/abstract/PRE/v72/e046108>.
- [9] Duch J, Arenas A. Community Detection in Complex Networks Using Extremal Optimization[EB/OL]. (2007-11-25). <http://link.aps.org/abstract/PRE/v72/e027104>.