

一种自适应均衡的移动 Ad Hoc 路由协议

陈卓¹, 李银国², 王田³

(1. 重庆工学院网络信息中心, 重庆 400050; 2. 重庆邮电大学自动化学院, 重庆 400065; 3. 重庆工商大学计算机学院, 重庆 400067)

摘要: 针对 Ad Hoc 路由协议普遍缺乏对网络负荷进行考察的缺点, 提出一种根据节点的负载高低进行自适应路由选择的协议 ADMAR, 其中包括一个反映当前网络负荷高低的负荷因子及模型。实验仿真结果表明, 该协议具有较理想的路由稳定性, 能在一定程度上实现网络负荷的分担, 减少网络热点的产生。

关键词: 自组织网络; 路由协议; 自适应

Adaptive Dispersive Mobile Ad Hoc Routing Protocol

CHEN Zhuo¹, LI Yin-guo², WANG Tian³

(1. Network Information Center, Chongqing Institute of Technology, Chongqing 400050;

2. College of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065;

3. College of Computer, Chongqing Technology and Business University, Chongqing 400067)

【Abstract】 Most Ad Hoc routing protocols select route based on shortest path mechanism, which does not take network load into consideration. This paper puts forward an adaptive dispersive mobile Ad Hoc routing protocol, namely ADMAR, which is based on the traffic load of the network node. The simulation shows that the routing protocol has good stability, can distribute the traffic load of Ad Hoc network to some extent, and reduces the network hotspots.

【Key words】 Ad Hoc network; routing protocol; adaptive

移动 Ad Hoc 网络在军事和商业领域有着重要的应用, 特别是在一些特殊的环境中, 由于某种因素而无法提供基站等通信基础设施, 用户又需要快速配置通信网络。同时, 移动 Ad Hoc 网络是一种典型的分布式系统。由于移动 Ad Hoc 网络是一种多跳的无线网络, 移动节点之间的数据通信使用共享的带宽受限的数据信道, 因此为了实现数据的多跳转发, 需要合理的分布式算法来实现路由、资源管理等功能。而移动 Ad Hoc 网络拓扑结构经常发生变化。因此, 设计的路由算法应能够自适应且较快地响应这种变化, 并能够维持网络的稳定性, 防止热点区域的产生。

1 按需激发的 Ad Hoc 路由协议

目前国内外关于移动 Ad Hoc 网络的研究主要集中于如何以低信息交换量来实现快速路由查找, 并以合理的度量方式选择可能存在的多条路径中最优的一条进行数据通信, 最常用的是以路径最短的标准来选择路径, 例如: AODV 协议规定在活动路径上的源节点和中间节点都以 RREP 分组中目的序列号 seq_num 较大和跳数 hop_cnt 最少的路径作为选取标准^[1]。这类路由协议在网络规模较小、相互通信的节点对较少的时候表现良好, 但随着网络规模的扩大、通信量增加后, 会暴露出一些问题, 如: 不能有效进行 QoS 控制, 不能合理分配网络负荷, 可能会使 Ad Hoc 网络在较短时间内出现局部热点, 导致网络寿命降低等。

在目前大量 Ad Hoc 路由协议设计中, 用于建立路由的信息包主要维护代表最短路径的信息及生命期等信息, 缺乏对通信路径负荷的考虑, 因此, 本文针对这一不足, 提出一种自适应均衡的移动 Ad Hoc 路由协议(Adaptive Dispersivity Mobile Ad Hoc Routing, ADMAR), 以均衡网络负荷。

2 ADMAR 协议

2.1 协议原理

由于 AODV 协议提供了比较完善的路由发现和路由维护机制, 并且支持 IP^[2], 因此选择将 AODV 作为基础来设计 ADMAR 协议。移动 Ad Hoc 网络的 MAC 层提供对无线信道的接入控制, 能为上层协议提供快速可靠的分组传送支持, 基于 IP 的网络层提供路由功能。将 MAC 层和网络层相结合进行跨层设计^[3], 即通过 MAC 层提供给网络层表示当前节点通信负载的负荷因子, 在 AODV 的交换信息数据帧头加入表示负荷因子的域, 并把负荷因子作为选择合理路由的另一个标准, 由此实现设计。

2.2 协议实现

本文定义一个表示 Ad Hoc 节点通信繁忙程度的量 T_i , 即节点 i 的负荷因子:

$$T_i = \frac{\sum_{m=1}^{k1} TX_m}{\sum_n^{k2} RX_n} \quad (1)$$

在式(1)中, 设节点 i 有 $k1$ 条数据发送路径, 通过这 $k1$ 条路径一共完成的数据发送速率为 $\sum_{m=1}^{k1} TX_m$, $k1$ 可视作节点 i 的出度。同时, 数据从 $k2$ 条路径发送至该节点, 总的接收速率为 $\sum_{n=1}^{k2} RX_n$, $k2$ 可视作节点 i 的入度。如果节点转发分组的

基金项目: 国家“863”计划基金资助项目(2004AA1Z2380); 重庆市科技攻关项目(5990)

作者简介: 陈卓(1980-), 男, 助教、硕士, 主研方向: 网络协议, Ad Hoc 网络, 嵌入式系统; 李银国, 教授、博士生导师; 王田, 副教授、博士后

收稿日期: 2007-06-20 **E-mail:** rakechen2000@yahoo.com.cn

效率较高,则 T_i 较大,如果节点接收分组的效率较高,则 T_i 较小。 T_i 值越大,说明节点 i 当前接收数据负荷的能力越强,即越“空闲”;对于 TX_m 和 RX_n ,采用对前一次计算出的值和当前计算出的值进行加权计算得出:利用 TX_{old} 和 TX_{new} 得到 TX_m ,即 $TX_m = \alpha TX_{old} + (1-\alpha)TX_{new}$ 。利用 RX_{old} 和 RX_{new} 计算出 RX_n ,即 $RX_n = \alpha RX_{old} + (1-\alpha)RX_{new}$ 。为了较好地反映节点当前的负荷状态, α 设置为较小的值 0.4,使计算结果受 RX_{new} 和 TX_{new} 的影响更大。从 T_i 的计算方式说明:一个节点的输出通道越多,数据发送速度越快,节点的 T_i 越大;相反,节点的输入通道越多,数据输入速度越快,节点的 T_i 越小。式(1)可以动态反映节点 i 的负载程度。另外,将采样时间间隔设定为 2 s,即每过 2 s,MAC 层协议会计算一次 T_i 。这里设置 2 s 为采样周期是因为:如果采样周期设置过小会导致一次路由还没有完成就再次计算节点的负荷,这样的计算是没有意义的;如果采样周期设置过大会使网络完成了多次路由后才计算一次负荷,从而不能正确反映某节点实际的负荷状况。本文取 $Sample_Interval$ 为 2 是实验所得较合理的经验值。

通过对节点负荷因子的计算,进一步可以得出某条路径 R_i 的负荷值的计算方法。定义某条 Ad Hoc 网络的路径为 $R_i(N_1, N_2, N_3, \dots, N_n)$,即数据从源节点 N_1 开始经多个中间节点到达目的节点 N_n 。该条路径负荷值 T_{R_i} 的计算如下:

$$T_{R_i} = \begin{cases} DefVal & hop_cnt = 1 \\ \min(T_i) & hop_cnt > 1 \end{cases} \quad (2)$$

当源和目的节点只经过一条路由时,采用默认值 $DefVal$,而当跳数大于 1 时,以路径中多个中间节点 T 最小的值作为该节点所在路径的 T_{R_i} 值。

AODV 协议的数据源节点在收到 RREP 分组后,会按照分组中的序列号 seq_num 及跳数 hop_cnt 来选择路由。即序列号较大表示一条较新的路由,最优先作为考虑,在序列号相同的情况下,选择跳数最小的路径作为实际的路由。在引入负荷因子后,选择路由多了一个衡量标准,序列号较大者仍作为最优先考虑,在序列号相同的情况下按照路径负荷 T 最大进行选择,如果 T 相同,按照跳数最小的路由进行选择。穿过“热点区”路径的负荷较小,在引入负荷因子后,数据源会选择负荷较小的路径,这样就可以绕开“热点区”,保证网络负荷的自适应分配。图 1 为数据源节点 N_1 选择合理的路由发送数据至目的节点 N_6 的示意图。

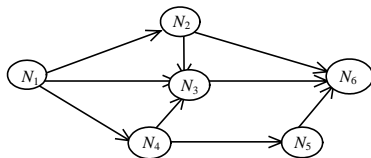


图 1 网络模型拓扑图

在某个时刻,假设 MAC 协议得到各节点当前的负荷因子为:节点 T_2 为 2, T_3 为 1, T_4 为 3, T_5 为 2;可供节点 N_1 选择的 路由 包括: $R_1(N_1, N_2, N_6)$; $R_2(N_1, N_3, N_6)$; $R_3(N_1, N_2, N_3, N_6)$; $R_4(N_1, N_4, N_3, N_6)$; $R_5(N_1, N_4, N_5, N_6)$ 。以及 $T_{R_1}=2$, $T_{R_2}=1$, $T_{R_3}=1$, $T_{R_4}=1$, $T_{R_5}=2$ 。根据式(2)的计算结果来选择合适的路由,由于经过路径 R_1 和路径 R_5 的负荷相同,因此这 2 条活动路径上的网络繁忙程度相当,相对于

R_2 , R_3 , R_4 空闲。根据跳数进一步选择路由,由于 R_1 的 hop_cnt 为 2,而 R_5 的 hop_cnt 为 3,因此选择跳数较小的 R_1 作为实际路由。在一次分组发送完成后,第 2 次发送分组时会向相邻节点发送 RREQ 广播分组,然后再次选择最合适的路由,这样保证了网络在出现热点区域以前新的路由可以绕开这一区域。中间节点的路由更新,本文采用了 AODV 的更新原理,即只有当确实发现更优的路由时才对路由信息进行更新,这样可以降低网络中的分组量。中间节点选择最优路由的标准按照目的序列号、网络负荷、跳数的次序进行。

ADMAR 协议引入了负荷因子,将其值保存于数据帧头部,并基于 AODV 的帧头实现了 ADMAR 帧头。另外,ADOV 协议使用了分布式的基于路由表的路由方式,路径中的每个节点都要执行维持和管理路由信息,由于引入负荷因子,需要改进路由表中保存的信息。在 MAC 层协议进行采样的同时会通过 Hello 分组向相邻节点广播,告知自己当前的负荷因子,如果某节点收到来自相邻节点 Hello 分组中的负荷因子大于自己路由表中的 $Load_Factor$ 值,说明相邻节点相对以前的负荷减轻了,应修改 $Load_Factor$ 为 Hello 分组中的负荷因子值。图 2 和图 3 分别表示 ADMAR 协议 RREP 分组的帧头和节点所维护的路由表。

类型	预留	前缀	跳数(hop_cnt)
目的节点 IP (Dest_IP)			
目的节点序列号(Dest_Seq)			
源节点 IP(Source_IP)			
目的节点负荷因子(Dest_Factor)			
生命周期(Life_Time)			

图 2 ADMAR 的 RREP 帧头信息

目的 IP 地址(Dest_IP)
目的节点序列号(Dest_Seq)
接口(Interface)
跳数(hop_cnt)
下一跳节点(Next_hop)
前驱节点(List of Precursors)
生命周期(Life_Time)
下一跳节点负荷因子(Load_Factor)
路由标记(Route_Flag)

图 3 ADMAR 所维护的路由信息

对于路由发现及维护,ADMAR 采用了和 AODV 一样的机制。而对于 MAC 层协议,本文采用 802.11 无线局域网协议进行仿真。由于 ADMAR 增加了负荷因子这个衡量标准,为了赋予网络层该值,对标准的 802.11 协议进行了部分修改。

3 实验仿真与结论

本节通过仿真实验验证 ADMAR 协议的性能,使用的仿真环境为 OPNET,实现的网络模型包括 16 个编号为 $mobile_node_i(1 \leq i \leq 16)$ 的移动节点,这些节点均匀分布于 $1\ 000\ m \times 500\ m$ 的范围中,采用 OPNET 提供的物理层模型并对 802.11 MAC 模型进行部分改动以支持负荷因子。这 16 个节点的移动模型采用随机 waypoint 模型^[4],该模型的基本原理为:节点首先在网络空间中计算一个随机选择的目的地位置,并以随机选择的恒定速度向节点移动。到达该节点后,节点暂停一个等待时间再计算下一个目的节点。设置 $Max_Speed = 20\ m/s$,节点移动的速度在 $[0, Max_Speed]$ 区间服从均匀分布。在利用 OPNET 进行仿真的时候,设置每个节点的数据分组到达时间间隔服从 $\lambda = 0.1$ 的指数分布,节点

(下转第 118 页)