

基于能量和链路状态的 AODV 路由请求转发机制研究

郝聚涛¹ 赵晶晶² 李明禄³

(上海理工大学计算机与电气工程学院 上海 200093)¹

(重庆大学输配电装备及系统安全与新技术国家重点实验室 重庆 400044)²

(上海交通大学计算机科学与工程系 上海 200240)³

摘要 按需距离矢量路由算法(AODV)为移动无线自组织网络(Manet)提供了高效的、扩展性能良好的路由解决方案。然而 AODV 在选择路径时仅以最短路径和最快响应作为度量准则,并未考虑节点能量、负载以及链路状态等因素,导致得到的路径并非为最优且节点能量消耗不均衡,降低了网络生存时间。在传统 AODV 算法基础上,提出了一个改进的路由算法,该算法在路由选择时充分考虑了节点的能量、负载以及节点间的链路状态,可以有效提高网络性能并延长网络生存时间。在 NS2 平台上的仿真实验结果也显示,所提算法在路由负荷、时延、分组传递率以及节能方面均优于传统 AODV 算法。

关键词 按需距离矢量路由,移动自组织网络,链路状态,能量高效路由

中图分类号 TP393 文献标识码 A

Energy Level and Link State-based AODV Route Request Forwarding Scheme Study

HAO Ju-tao¹ ZHAO Jing-jing² LI Ming-lu³

(School of Computer and Electrical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)¹

(Department of Electrical Engineering, Chongqing University, Chongqing 400044, China)²

(Department of Computer Science & Engineering, Shanghai Jiaotong University, Shanghai 200240, China)³

Abstract Ad-hoc On-demand Distance Vector(AODV) provides scalable and effective solution for packet routing in mobile wireless ad hoc networks, however, the path generated by this protocol may deviate far from optimal because of inconsideration of node power, load unbalance and link state, resulting that energy consumption of each nodes is imbalanced and reducing the lifetime of whole networks. Based on the basic AODV protocols, an improved protocol was presented. When the protocol is selecting a route, node power, load balance and link state between nodes are all considered. The proposed protocol can improve the network performance. Through the simulation on NS2, it is confirmed that the improved AODV protocol is more energy-efficient than AODV and has higher data package delivery rate, lower end to end delay and lower route load.

Keywords On demand distance vector routing, Mobile ad hoc networks, Link state, Energy efficient routing

1 引言

移动自组网(Mobile Ad Hoc Networks, MANET)是由一组移动节点构成的、不依赖于任何固定基础设施的无线网络^[1]。处于无线传输范围内的节点可以直接通信,而相距较远的节点则需要依靠中间节点的转发进行通信。中间节点就充当了路由器的角色。由于不需要固定基础设施的支持以及迅速部署的特点,Ad hoc 网络非常适合通信基础设施不存在或者无法正常使用的环境,其潜在的应用领域非常广泛,例如军事战术网、灾难救助、移动会议、传感器网络和个人域网等。

Ad hoc 网络中,由于通信半径的限制,网络节点之间是通过多跳数据转发机制进行数据交互的,需要路由协议完成

分组转发决策。与传统路由协议相比,Ad hoc 路由协议的设计面临着网络拓扑动态变化、带宽受限、信道容量变化、移动终端有限的可用资源等新的问题和挑战。

目前提出的 Ad hoc 网络路由协议一般基于两种方式:先验式(Proactive)路由协议和反应式(Reactive)路由协议。先验式路由协议又称为表驱动(Table Driven)路由协议,在这种路由协议中,每个节点维护一张包含到达其它节点的路由信息的路由表。当检测到网络拓扑结构发生变化时,节点在网络中发送更新消息。收到更新消息的节点将更新自己的路由表,以维护一致的、及时的、准确的路由信息,所以路由表可以准确地反映网络的拓扑结构。源节点一旦要发送报文,可以立即获得到达目的节点的路由。因此这种路由协议的时延较

到稿日期:2008-08-15 返修日期:2008-10-21 本文受国家 973 重点基础研究发展规划项目(2006CB303000)资助。

郝聚涛(1976-),男,博士,讲师,主要研究方向为移动无线网络、生物信息处理等,E-mail:jt_hao@usst.edu.cn;赵晶晶(1980-),女,博士研究生,主要研究方向为无线传感器网络、电力系统;李明禄(1965-),男,教授,博士生导师,主要研究方向为网格计算、服务计算、对等计算、无线网络等。

小,但是路由协议的开销较大。反应式路由协议,又称为按需(On Demand)路由协议,是一种当需要发送数据时才查找路由的路由算法。在这种路由协议中,节点不需要维护及时、准确的路由信息,当向目的节点发送报文时,源节点才在网络中发起路由查找过程,找到相应的路由。与先验式路由协议相比,反应式路由协议的开销较小,但是数据报传送的时延较大。典型的表驱动路由协议包括最佳链路状态协议 OLSR (Optimal Link State Routing)^[2]、目的序列距离向量 DS-DV^[3]、簇首网管交换路由 CGSR (Clusterhead Gateway Switch Routing)^[4]等;按需路由协议包括按需距离向量路由 AODV(Ad hoc On-demand Distance Vector)^[5]、动态源路由协议 DSR(Dynamic Source Routing)^[6]、临时排序路由算法 TORA(Temporally-Ordered Routing Algorithm)^[7]、基于关联的路由协议 ABR(Associativity-Based Routing)^[8]等。

AODV 路由协议致力于在源端和目的端之间发现时延最小的最短路径。在有线网络中,该路径确实是最优的。但对于移动 Ad hoc 网络来说,最短路径可能恰恰是最坏的路径。这是由于在最短路径中,相互通信的两个端点可能处于对方通信半径的边缘,节点移动可能引起链路的中断,这就是所谓的边缘效应问题^[9]。同时,AODV 路由协议在进行路由选择时并未考虑节点的实际负载情况,当节点的负载很重时,会增加通信延迟。同时,处于多条通信链路中的节点由于长期为其它节点转发数据包,会很快因电能耗尽而失效。当系统中存在过多实效节点时,便会引起网络的分割,而不能正常通信。

2 按需距离矢量路由协议 AODV

按需路由协议通常分为路由发现和路由维护两个阶段。

2.1 路由发现

AODV 路由协议是一种按需路由协议,只有在两个节点需要进行通信,且源节点没有到达目的节点的路由时才进行路由发现过程。AODV 使用广播式路由发现机制。当源节点想与另一节点进行通信,而它的路由表中又没有到达目的节点的路由条目时,就广播一个路由请求(RREQ)报文。中间节点在收到 RREQ 报文后,首先比较本节点和目的节点的 IP 地址。如果自己是目的节点,就向源节点回复路由响应(RREP)报文。如果自己不是目的节点,则根据 RREQ 报文中的源 IP 地址和广播 ID 判断是否收到过该 RREQ 报文。如果收到过,则丢弃该 RREQ 报文。若没有收到过该 RREQ 报文,就记录相关信息,用以形成反向路由。记录的信息主要包括目的地址、源地址、广播 ID、源序列号、反向路由超时时长,同时将 RREQ 报文中的跳数字段(hop count)值加 1,并向邻居节点转发 RREQ 报文。当中继节点或目的节点沿着反向路径回复 RREP 报文时,这条路径上的节点建立前向路由。当 RREP 报文到达源节点后,源节点就可以使用已经建立的路由发送数据报文。

2.2 路由维护

AODV 通过周期性地广播 Hello 报文来监视链路状态。若节点在使用某个链路时发现该链路断开,节点将在 DELETE_PERIOD 之后从路由表中删除包含该断开链路的路由,并发送“路由出错”报文(RRER)通知那些因链路断开而不可达的节点将对对应路由从路由表中删除,沿途转发 RRER

的节点也删除自己路由表中的对应路由。如果断链处的上游节点与目的节点之间的距离小于 MAX_REPAIR_TTL 跳,则该节点启用生存时间比较小的 RREQ 广播来修复路由,即采用局部修复机制来降低延迟,否则由源端节点发起 RREQ 来进行修复。

3 改进的 AODV 路由请求转发机制

在传统的 AODV 协议中,为了避免产生广播风暴,任何中间节点收到路由请求 RREQ 后,都要随机延迟一段时间,再转发该请求。该延迟时间为服从 $[0,1]$ 均匀分布的任意随机取值与 0.01s 的乘积。

很显然,在传统的 AODV 协议的路由请求转发机制中,并没有考虑到节点本身的特性差异,例如电量、负载等。因此建立的路由更容易受节点电量的限制,同时使得某些节点异常繁忙、其它节点非常空闲,导致负载分布不均衡,降低系统的吞吐量和生存时间。

我们对路由请求转发机制的改进主要包括:当节点收到路由请求后,节点能够根据自身的剩余电量、负荷情况以及与发送节点之间的链路状态来决定延迟的时间。

第一,在能量有限的移动自组织网络中,节点的剩余电量是选择路由时应该考虑的重要因素,并已经成为研究的热点之一。当节点收到路由请求时,节点能够根据自己的剩余能量来计算请求转发延迟时间。引入该约束条件,可以节省节点电池能源,延长网络寿命。

本文所采用的能量模型计算公式为

$$\text{Energy} = \text{Power} \times \text{Time} \quad (1)$$

即当一个节点发送或者接收一个包时所消耗的能量是由该节点发送或接收的功率和处理该包所需要的时间决定的。其中,处理一个包所需要的时间为

$$\text{Time} = 8 \times \text{Packet size} / \text{Bandwidth} \quad (2)$$

因此,

$$E_{tx} = P_{tx} \times 8 \times \text{Packet size} / \text{Bandwidth} \quad (3)$$

$$E_{rx} = P_{rx} \times 8 \times \text{Packet size} / \text{Bandwidth} \quad (4)$$

其中, P_{tx} 、 P_{rx} 分别代表发送和接收时的功率。

对于转发一个包所消耗的全部能量为

$$E_f = E_{tx} + E_{rx} \quad (5)$$

令 E_{i0} 和 E_{i1} 分别表示节点 i 的初始能量和剩余能量, T_E 表示因节点 i 的能量影响而引起延迟时间。 r_E 表示节点 i 当前剩余能量比,计算公式为

$$r_E = E_{i0} / E_{i1} \quad (6)$$

为了尽可能保持网络的连通性,降低延迟,同时延长网络的生存时间,在本文中,只有当 $r_E < 0.2$ 时,即节点的剩余能量为初始能量的 20% 时,能量约束才起作用。因能量约束而引起的延迟为

$$T_E = 1 / (1 + r_E) \quad (7)$$

第二,在传统 AODV 协议中,路由请求转发机制并没有考虑节点缓存队列的情况。节点缓存队列反映了节点本身对包的处理能力以及与其它节点间的通信能力。因此将节点的缓存队列中所缓存的包的情况引入到路由建立过程中,对于节点负载不均衡和能量消耗不平衡具有有效的抑制作用。

令 Q_{Lm} 表示节点 i 缓存队列的最大长度, Q_{Li} 表示当前节点 i 当前缓存队列长度。那么,由于节点缓存队列所引起的

延迟时间 T_{iQ} 为

$$T_{iQ} = Q_{iL} / Q_{iLm} \quad (8)$$

第三,在移动自组织网络中的节点,其运动具有连续性。也就是说,在短时间内,新的拓扑结构是在原来的拓扑结构基础上进行微小变动得到的。传统的 AODV 协议总是选择最短路径。由于节点的传输距离有限,因此最短路径往往也是最容易断开的。路径失效通常只是因为最薄弱的链路断开引起的。为此,在路由建立时,尽可能寻找稳定的链路,可以有效地避免链路断开,减少路由请求次数,降低时延,提高系统吞吐量。

目前常用的无线传输模型主要有自由空间(Free space)模型、双径地面反射(Tow-ray ground reflection)和阴影(Shadowing)模型。在本文中,采用双径地面反射模型。该模型除了考虑直线传播,还考虑地面反射路径。此模型在长距离传输环境中具有更准确的预测能力。接受信号功率的公式如下:

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \quad (9)$$

d 为发送方和接收方的距离; P_t 为发射的信号功率, G_t 和 G_r 分别为发送方和接收方的天线增益; L 为系统损耗; h_t 和 h_r 分别为发送方和接收方的天线高度。在本文中,假设各节点发射能量相同并且传输信号能量损失完全由节点距离增加引起。因此,根据式(9)可以用节点间的距离来表示该对节点间的链路状态。在考虑链路状态的请求转发机制中,当节点间链路信号状态较好,即距离相对较小时,节点将以相对较小的时延将路由请求转发出去。

令 T_{sig} 表示由于链路状态而引起的转发时延,其计算公式为

$$T_{sig} = D_{ij} / D_{Max} \quad (10)$$

其中, D_{ij} 表示发送节点 j 与接收节点之间的距离,而 D_{Max} 表示节点的最大发射距离。

综合上面 3 个方面而得到最终延迟为:

$$T = T_C \times (\alpha_1 T_E + \alpha_2 T_{iQ} + \alpha_3 T_{sig}) \quad (11)$$

其中 α_1, α_2 与 α_3 表示权重系数,且满足 $\alpha_1 + \alpha_2 + \alpha_3 = 1$; T_C 为一常量,用于控制延迟大小。

4 改进协议仿真与性能分析

为了验证改进的路由请求转发机制对协议性能的影响,本文在网络模拟平台 NS2 上进行了模拟试验,并与 AODV 协议进行了比较。

4.1 模拟环境设置

在仿真模型中,设置了 50 个移动节点,随机分布在 $1200m \times 1200m$ 的矩形区域内。每个节点的无线带宽设为 $1Mb/s$,无线传输模型采用双径地面反射模型,MAC 层协议采用 IEEE802.11。传输层使用 UDP,数据流采用 CBR 业务,包长度为 512B,发送间隔为 1s,最大连接数为 30。节点的无线发射半径均为 250m。节点位置与运动速度随机产生,节点在运动到一个目的地后不做停留继续运动,节点的最大运动速度分别为 $0m/s, 5m/s, 10m/s, 15m/s, 20m/s$ 。所有节点的初始能量设置为 60J,发射功率和接收功率分别为 $200mW$ 和 $600mW$ 。仿真时间为 600s。

在本文仿真试验中,式(11)参数设置为: $\alpha_1 = \alpha_3 = 0.4$,

$\alpha_2 = 0.2, T_C = 0.08$ 。

4.2 模拟仿真结果

图 1 给出了 AODV 与改进协议性能的比较结果。图 1(a)给出了两种协议平均跳数的比较,结果显示改进协议发送数据包所经历的平均跳数低于传统 AODV 协议。图 1(b)给出了路由开销之间的对比,结果表明,改进协议的路由开销比 AODV 中的要小,这主要是因为路由建立过程中考虑了链路稳定因素,使得链路对于节点移动具有一定的抵抗能力,减少了链路断开次数,降低了路由开销。由图 1(c)可以看出,改进协议在包到达率方面比 AODV 有显著提高。图 1(d)给出了时延的比较,由图可以看出,在不同最大速度下,时延对比各有优劣。

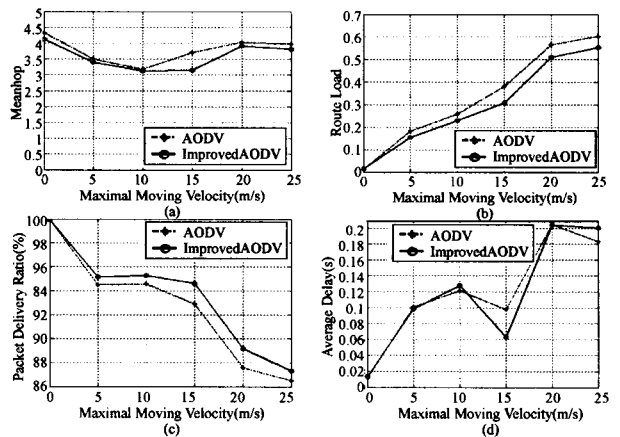


图 1 AODV 与改进 AODV 协议性能比较

图 2(a)显示了在使用 AODV 协议时,各个移动节点的电池能量消耗情况。图 2(b)是在采用改进协议情况下各节点剩余电量的变化情况。对比两图,虽然随着时间推移,各个移动节点都会消耗能量,但由于协议在路由选择时会考虑节点的剩余能量,避免某些节点由于能量过早耗尽而退出系统,因此在使用改进协议情况下,各节点的剩余能量更为均衡。

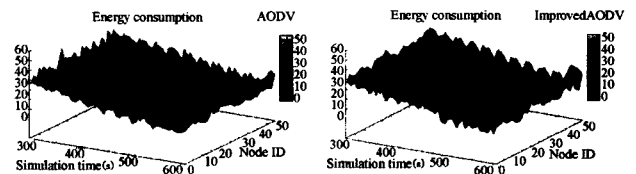


图 2 AODV 与改进 AODV 节点电量变化比较

结束语 按需距离矢量路由算法(AODV)为移动无线自组织网络提供了高效的、扩展性能良好的路由解决方案。然而 AODV 在选择路径时仅以最短路径和最快响应作为度量准则,并未考虑节点能量、负载以及链路状态等因素,导致网络性能下降以及能量消耗不均衡。本文在传统 AODV 算法基础上,提出了一个改进的路由算法,该算法在路由选择时充分考虑了节点的能量、负载以及节点间的链路状态,可以有效提高网络性能并延长网络生存时间。仿真实验结果显示,该算法在路由负荷、时延、分组传递率以及节能方面均优于传统 AODV 算法。

参考文献

[1] Chlamtac I, Conti M, Liu J. Mobile ad hoc networking: Imperatives and challenges[J]. Ad Hoc Networks, 2003, 1(1): 13-64

mantic(Data_Type Descriptor, ...)每个事件都属于某个特定的构件对象,称之为主体对象,Objclass 指定事件对象所属的类。Descriptor 是对象产生的一类事件的抽象描述,为分类对象上操作语义相似的事件。Operator_Semantic 指定事件的操作语义。Dey 介绍了情境信息的 4 个本质特征^[11]: 身份、位置、状态和时间。被应用使用的身份标识在命名空间内必须是唯一的,Objclass, Descriptor, Operator_Semantic 表明情境信息的身份。自描述的数据类型表达情境的位置、状态和时间等信息。

物理环境中的实体建模为面向对象的软件中的类。当感知到情境信息发生变化时,对象触发相应的事件。事件在服务构件内定义,由客户构件订购。服务构件获取来自传感器或外设的原始数据。以事件建模情境,分类应用客户感兴趣的事件,维护情境数据,当监视到某一事件发生时,立即通知事件注册者。

事件驱动的体系结构在设计时面临高并发性和向软件开发者提供简单透明编程接口的挑战。计算实体或物理设备间交换数据有同步和异步两种模式。就文件系统来说^[12],同步保证了提交到磁盘的文件在操作系统崩溃或电源失效时不会丢失数据。但是,在数据提交过程中其它应用不能有效地利用磁盘设备。相反,异步方式能有效地利用磁盘设备,但存在数据丢失隐患。因此,文件系统的设计需要权衡性能和响应时间的问题。构件间交互模式的设计面临更为复杂的系统问题,系统的消息/事件处理机制的线程模型是复杂系统软件系统设计中的一个非常核心的部分,对于系统的性能、稳定性和开发成本有决定性的作用。在构件模型方面,引入面向应用领域的 Applet 构件,Applet 构件的回调上下文对象维护事件队列,接收服务构件对象的事件通知,同步调度回调处理函数。在运行的初始阶段,Applet 构件部署满足用户需求的 UI 以及创建服务构件,在服务对象上注册感兴趣的事件,循环处理事件队列中由服务构件投递的事件。图 4 为 Applet 构件和服务构件的交互模式。

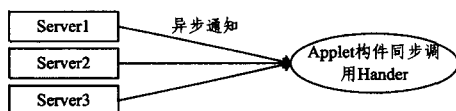


图 4 构件间的交互模式

事件在 CAR 文件里以回调接口定义,由自动代码框架工具产生数据存储、事件合并、事件通知等相应的软件设施,并为客户端产生订阅和取消订阅事件接口。

结束语 由于普适计算环境的复杂性和挑战性,迫切需要一种具有感知和自适应能力的中间件。与其它中间件技术

比较,提出的面向普适计算的中间件特点在于以下几个方面:接口元模型以同步和异步接口的方式获取构件信息并提供省内省设施和对外的服务;情境信息在构件对象内被建模,构件间以基于事件异步通知的方式交互,获得了时间、空间和同步上的解耦;框架元模型允许应用开始运行时装载一些核心功能,随着运行环境和需求的变化,动态聚合或拆卸一个或多个方面构件,从而改变自身的结构和行为。异质普适设备的计算能力往往是有限的,在运行阶段,软件启动运行时加载一些核心功能,随着运行环境的变化,系统不间断地添加、删除和替换构件成分。最终,应用能够感知和适应动态变化的环境,满足普适计算环境下的动态自适应需求。

参 考 文 献

- [1] Weiser M. The computer for the twenty-first century [J]. Scientific American, 1991, 265(3): 94-104
- [2] 徐光祐, 史元春, 谢伟凯普适计算[J]. 计算机学报, 2003, 26(9): 1042-1052
- [3] Blair G, Coulson G, Andersen A, et al. The design and implementation of Open ORB 2[J]. IEEE Distrib. Syst., 2001(6)
- [4] Monson-Haefel R. Enterprise JavaBeans 3. 0, fifth ed [M]. O'Reilly & Associates, 2006
- [5] Barnett M, Schulte W. Runtime verification of . Net contracts [J]. Systems and Software, 2003, 65: 199-208
- [6] Object Management Group. CORBA Component Model Specification [EB/OL]. OMG Available Specification Version 4. 0, 2006. OMG Document formal/06-04-01
- [7] Blair G, Coulson G, Grace P. Research Directions in Reflective Middleware: the Lancaster Experience[C]//Proc. 3rd Workshop on Reflective and Adaptive Middleware (RM2004). Canada, 2004: 262-267
- [8] 刘家红, 吴泉源. 一个基于事件驱动的面向服务计算平台[J]. 计算机学报, 2008, 31(4): 587-599
- [9] Szyperski C. Component Software: Beyond Object-Oriented Programming [M]. Addison-Wesley, 2002
- [10] Ranganathan A, Campbell R H. A Middleware for Context - Aware Agents in Ubiquitous Computing Environments [C]//ACM/IFIP/ USENIX Int '1 Middleware Conf. LNCS 2672. Springer-Verlag, 2003
- [11] Dey A, Abowd G, Salber D. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications[J]. Journal on Human Computer Interaction, Lawrence Erlbaum publ., 2001, 16(2-4): 97-166
- [12] Edmund B N, Kaushik V, Peter M C, et al. Rethink the Sync [C]//OSDI'06, 7th USENIX Symposium on Operating Systems Design and Implementation. Seattle, WA, October 2006: 1-14

(上接第 70 页)

- [2] Clausen T, Jacquet P. Optimized Link State Routing Protocol [J]. Internet Draft draft-ietf-manet-olsr-11, July 2003
- [3] Perkins C E, Bhagwat P. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for mobile computers [C]//SIGCOMM. 1994: 234-244
- [4] Chiang C C. Routing in Clustered Multi-hop, Mobile Wireless Networks with Fading Channel [C]//IEEE SICON'97. 1997: 197-211
- [5] Perkins C, Royer E, Das S. Ad Hoc On-demand Distance Vector Routings[S]. RFC 3561, 2003-07

- [6] Johnson D, Maltz D. Dynamic source routing in Ad hoc wireless networks [C]//Imielinski T, Korth H, eds. Mobile Computing, Kluwer, 1996: 153-181
- [7] Park V D, Corson M S. A highly adaptive routing algorithm for mobile wireless networks [A]//Proceedings—IEEE INFOCOM [C]. 1997: 1405-1413
- [8] Chai - Keong T. Associativity - based routing for Ad - Hoc mobile networks [J]. Wireless Personal Communications, 1997, 4: 103-139
- [9] 冯美玉, 程胜, 张勤, 等. Ad hoc 网络中自愈路由协议研究[J]. 北京邮电大学学报, 2005, 28(2): 46-49