

基于 AODV 的贪婪路由协议

朱 鸿, 单 洪, 黄 郡

(电子工程学院网络工程系, 合肥 230037)

摘要: 针对 AODV 路由协议在节点高速移动环境中存在的低性能问题, 提出一种利用贪婪算法并充分考虑节点移动状态的路由协议 GAODV。该协议涉及扩展 Hello 报文、限制控制分组数量、建立稳定路由以及禁用本地路由修复。仿真分析表明, GAODV 比 AODV 具有更高的吞吐量, 在节点移动速度达 80 m/s 时, 端到端的平均时延减少了 55%。

关键词: Ad Hoc 网络; 贪婪路由协议; 按需距离向量路由协议

Greedy Routing Protocol Based on AODV

ZHU Hong, SHAN Hong, HUANG Jun

(Department of Network Engineering, Electronic Engineering Institute, Hefei 230037)

【Abstract】 Aiming at the capability deficiency of AODV routing protocol in situation where nodes move at high speed, this paper proposes a routing protocol called GAODV which using the greedy algorithm and considering nodes' moving state fully. This protocol includes extending Hello packet, restricting the number of control packet, establishing steady route and forbidding local routing repair. Simulation analysis shows that compared with AODV, GAODV has better throughput, and the average end-to-end delay is decreased by 55% when the node's speed comes to 80 m/s.

【Key words】 Ad Hoc network; greedy routing protocol; on-demand distance vector routing protocol

1 概述

无线 Ad Hoc 网络是由一组不依赖基础网络设施、带有无线收发装置的移动节点组成的无线通信网络^[1], 由于其结构简单, 具有较强适应性和生存能力, 因此被大量用于军事、抢险救灾等紧急任务场合。

路由技术一直是 Ad Hoc 网络研究的重点和热点, 目前较成熟、典型的路由协议有 DSDV, DSR, AODV, ZRP 等^[2]。Ad Hoc 网络拓扑结构动态变化的特性给路由协议设计带来了一定困难, 现有协议不能适应所有网络状况。尤其是在节点移动性增强的情况下, 有效路由上链路断开的几率增大, 数据包丢失率上升, 导致网络性能急剧下降, 无法保证实时性要求较高业务的服务质量。

2 研究现状

按需距离向量路由协议 AODV 是一种经典的反应式路由协议, 已标准化并得到广泛应用。该协议由路由发现和路由维护 2 个阶段组成。在路由发现阶段, 通过有效控制洪泛 RREQ 路由请求包、目的节点回复 RREP 路径应答包, 源节点找到一条到目的节点的路由后, 采用单播发送数据分组。在路由维护阶段, 由断链处上游节点负责路由的本地修复或向上游节点发起 RERR 路由错误通告, 源节点收到 RERR 通告后重新启动路由发现过程^[3]。

在使用 AODV 协议的 Ad Hoc 网络中, 移动节点在需要时才维护路由条目, 网络开销少; 节点间周期性地向邻居发送 Hello 消息以进行本地连通性管理, 能够快速应对有效路由中链路断开的情况。该协议通过维护一系列序列号和标识 ID 号, 能有效防止路由循环。它具有一定可扩展性, 在节点移动性不强的情况下能适应大规模网络。图 1 描述了 AODV 协议的报文交换方式, 其中, S 和 D 分别代表源节点和目的

节点; 1 和 2 代表中间节点。

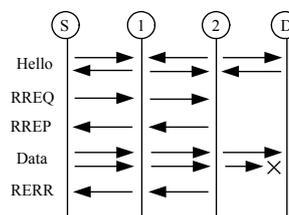


图 1 AODV 报文交换

目前对 AODV 协议进行改进的研究很多。文献[4]提出在 Ad Hoc 网络中依靠一些可靠(计算能力强、电池能量充足)节点来提高路由生存能力。将可靠节点放置在网络中有利于通信的优化位置, 并根据网络状态的变化, 动态改变可靠节点的位置和运动轨迹, 以增加在网络中找到从源节点到目的节点之间可靠路由的几率。但上述算法较复杂, 且需要一些可靠节点的支持。文献[5]提出一种基于链路状态的自愈路由协议, 基本思想是节点测量接收到的数据包或路由请求包的信噪比, 并根据事先设定的阈值判定链路状态, 以便在链路彻底断开前重新发起路由请求或修复路由, 从而加速路由恢复、避免不必要的数据丢失和重传。文献[6]的方法与文献[5]类似, 主要思想是通过观察接收到的数据分组的能量值, 判断链路状态, 提前告知所有正在使用即将失效链路的节点, 使这些节点及时删除缓存中即将失效的路由, 并进行新的路由发现, 进而减少网络中的分组丢失率。

本文假定利用 GPS 或其他手段来获得节点的位置、速度、

作者简介: 朱 鸿(1982 -), 男, 硕士研究生, 主研方向: 无线网络; 单 洪, 教授、博士生导师; 黄 郡, 硕士研究生

收稿日期: 2008-03-02 **E-mail:** hong_001_916_82@163.com

运动方向等信息,在 AODV 路由协议基础上提出贪婪路由协议 GAODV,以提高传统 AODV 协议在节点高速移动、网络拓扑变化频繁情况下的性能。

3 移动参数

节点的高速移动导致网络拓扑结构不断变化,给自组路由带来不良影响。如果网络中的节点可以相互通信,那么节点之间必定存在有效路径,如果能从这些路径中找出一条最稳定的,就能降低链路失效的几率,进而减少丢包率。

在 Ad Hoc 网络中,由于各个节点移动方向不同,导致某些节点间的通信链路在很短时间(由节点的高速运动决定)之后就会断开,因此选择这种链路组成的路由不是好路由,而传统 AODV 协议在路由发现时无法对链路状态作出预测,因此,无法对上述链路进行筛选。本文利用移动参数 mp 来反映相邻节点间的相对运动趋势,在路由发现阶段尽量选择不易断开的链路,建立相对稳定的路由,以减少分组丢失率。设 i, j 为网络中 2 个相邻节点, mp_{ij} 根据式(1)来计算。

$$mp_{ij} = (\Delta x_i - \Delta x_j)^2 + (\Delta y_i - \Delta y_j)^2 \quad (1)$$

其中, $\Delta x_i = v_i \times \cos \theta_i$; $\Delta x_j = v_j \times \cos \theta_j$; $\Delta y_i = v_i \times \sin \theta_i$; $\Delta y_j = v_j \times \sin \theta_j$ 。其中, v 表示节点移动速度; θ 表示节点移动方向; Δx 表示 $t=1$ s 内节点在 X 轴方向上的位移; Δy 表示 $t=1$ s 内节点在 Y 轴方向上的位移。由于节点在高速移动情况下运动方向一旦确定,短时间内不会有太大变动,因此规定移动时间为 1 s 以方便计算,此时 2 个节点在 X 轴和 Y 轴方向上位移之差的平方正好反映了节点间的相对运动趋势,由式(1)可以看出, mp_{ij} 值越大,节点 i 和 j 之间距离随着时间增加而增大的可能性越大。

4 GAODV 路由协议算法描述

本文提出的 GAODV 路由协议利用贪婪算法对 AODV 路由协议进行改进。通过 GPS 系统或其他手段辅助提供节点移动信息,网络中的节点可以获得相邻节点和目的节点的坐标。与 AODV 类似,整个协议分为路由发现和路由维护 2 个阶段。设 S 表示源节点, D 表示目的节点, mp 为移动参数。

4.1 算法可行性

在将算法应用于 AODV 协议前,先进行简要的方案论证,从链路稳定度的角度说明本文算法的可行性。

4.1.1 接收信号强度

在 Ad Hoc 网络中,接收节点能否正确接收来自发送方的信息取决于接收节点接收到的信号强度大小^[7],接收信号强度定义为

$$P_r = N \frac{P_t}{r^4} \quad (2)$$

其中, P_t 为发送功率; N 是天线常数; r 是节点之间的距离。

可见,在发送功率 P_t 一定的情况下,由于节点移动引起 r 的细微增加会使信号强度 P_r 的值快速减小,因此当 P_r 小于某个阈值时,节点间不能正常通信,即它们之间的通信链路断链。这说明在节点做高速移动的网络环境中,移动性对链路稳定性有较大影响。下文将对链路稳定性进行量化,并说明算法的可行性。

4.1.2 链路稳定度

观察网络中 2 个高速移动节点 i 和 j ,它们的移动状态和位置关系如图 2 所示。设 R_{\max} 为节点间的最大有效通信距离。在图 2 描述的场景中,虚线圆圈表示节点 i 的最大通信范围。2 个节点在 t_0 时刻的坐标分别为 (x_{i0}, y_{i0}) , (x_{j0}, y_{j0}) 。经过时间 t 后,2 个节点的坐标分别为 (x_{it}, y_{it}) , (x_{jt}, y_{jt}) 。在 (t_0+t) 时刻,节点间

距离为 R_{\max} ,且下一时刻节点 i 和 j 之间的距离将超过 R_{\max} (假定 i 和 j 做相互远离的运动),即节点间链路断开,则 2 个节点在最大有效通信距离范围内的运动时间 t 可用式(3)估算。

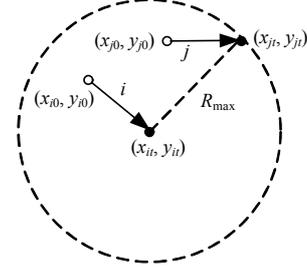


图 2 节点移动模型

$$(x_{it} - x_{jt})^2 + (y_{it} - y_{jt})^2 = R_{\max}^2 \quad (3)$$

其中, $x_{it} = x_{i0} + v_i \cos \theta_i t$; $x_{jt} = x_{j0} + v_j \cos \theta_j t$; $y_{it} = y_{i0} + v_i \sin \theta_i t$; $y_{jt} = y_{j0} + v_j \sin \theta_j t$ 。

代入式(3)得

$$\begin{aligned} & [(x_{i0} - x_{j0}) + (v_i \cos \theta_i - v_j \cos \theta_j)t]^2 + \\ & [(y_{i0} - y_{j0}) + (v_i \sin \theta_i - v_j \sin \theta_j)t]^2 = R_{\max}^2 \end{aligned} \quad (4)$$

设 $a = x_{i0} - x_{j0}$, $b = v_i \cos \theta_i - v_j \cos \theta_j$, $c = y_{i0} - y_{j0}$, $d = v_i \sin \theta_i - v_j \sin \theta_j$ 。

代入式(4)得

$$(a + bt)^2 + (c + dt)^2 = R_{\max}^2$$

解得

$$t = \frac{(cd - ab) \pm \sqrt{(b^2 + d^2)R_{\max}^2 - (ad + bc)^2}}{b^2 + d^2} \quad (5)$$

其中, t 称为链路稳定度,由于假定节点作相互远离的运动,因此只考虑 t 为正实根的情况,对于正实根, t 值越大,表示 2 个节点在有效通信范围内活动的时间越长,即节点间链路越稳定,反之相反; $b^2 + d^2 = (v_i \cos \theta_i - v_j \cos \theta_j)^2 + (v_i \sin \theta_i - v_j \sin \theta_j)^2$ 。上文提到,当节点高速移动时,其移动方向在短时间内不会发生太大变化,且式(2)说明 t 应是一个较小的值(由于节点高速移动,因此节点间距离变化较快,在很短时间之后接收强度就可能低于阈值),由式(1)可得 $b^2 + d^2 \approx mp_{ij}$ 。

因为 $\lim_{b^2+d^2 \rightarrow \infty} t = 0$,所以当 mp_{ij} 值很大(节点相对运动趋势很大)时,节点间链路稳定度趋近于 0。本文算法始终选择 mp_{ij} 值最小的节点作为下一跳转发节点,从而保证了有效路径上节点间的链路具有最优链路稳定度,即保证了断链几率最小。

4.2 路由发现阶段

文献[8]给出了单个路由寻找代价的计算公式:

$$1 + FwReq + OgRep + FwRep \quad (6)$$

其中, 1 表示起始路由请求分组的传输; $FwReq$ 表示所转发的路由请求分组数量; $OgRep$ 表示路由应答分组产生节点的个数; $FwRep$ 表示所转发的路由应答分组数量。传统 AODV 采用洪泛 RREQ 的方式寻找路由,根据式(6)可知,大范围的洪泛分组会增加协议开销,导致网络性能降低。

节点移动性造成的协议负载与断链数成正比^[9],每条链路断链产生的协议负载为 $O(L \lg(n)/r^2)$,其中, n 为网络中的节点数; r 为节点的传输半径; L 是与路径跳数有关的值(n 的函数)。可见,随着网络规模扩大,断链产生的协议开销也会增加,而传统 AODV 协议选出的最终用于转发数据分组的路由没有考虑链路状态,在节点高速移动的情况下断链几率高,随着网络规模的扩大,其协议开销也会增大,因此,网络可

扩展性不强。

在 GAODV 路由协议中,节点之间通过周期性地互通携带运动参数信息的 Hello 报文,利用移动参数 mp 来限制 RREQ 的洪泛,极大减小了寻由代价,并能建立起稳定的路由,可以较好地适应网络规模的扩大。具体扩展部分的伪代码如下:

(1)在 Hello 报文中加入节点移动参数信息:

```
typedef struct
{
    ...
    double x,y;//节点当前坐标
    double nextx,nexty;//节点下一点坐标
    double avgSpeed;//节点移动速度
} AODV_RREP_Packet;
```

(2)在邻居列表表项中加入移动参数信息:

```
typedef struct NTE
{
    ...
    double nbx,nby;//节点当前坐标
    double nbnextx,nbnexty;//节点下一点坐标
    double nbavgSpeed;//节点移动速度
    double nbdistance;//邻居节点到目的节点距离
    ...
} AODV_NT_Node;
```

在上述扩展部分中,根据三角公式(式(7)),节点坐标信息可用于估算 $\sin\theta$ 和 $\cos\theta$ 。

$$\begin{cases} \sin\theta=(y_2-y_1)/\sqrt{(y_2-y_1)^2+(x_2-x_1)^2} \\ \cos\theta=(x_2-x_1)/\sqrt{(y_2-y_1)^2+(x_2-x_1)^2} \end{cases} \quad (7)$$

当节点 S 有数据包向节点 D 发送时,本文协议的具体处理过程如下:

(1)如果 S 自身有到 D 的有效路由,则直接发出数据包,否则执行贪婪算法,转(2);

(2)S 根据自身维护的邻居表,先选出邻居节点(一跳节点)中距离目的节点 D 比自己近的节点,设为 (n_1, n_2, n_3, \dots) ,然后求出 $\min(mp_{sn_1}, mp_{sn_2}, mp_{sn_3}, \dots)$,选择具有该最小 mp 值的节点作为下一跳并转发 RREQ,其中, mp_{sn_i} 表示源节点 S 和邻居节点 n_i 之间的 mp 值;

(3)中间节点收到 RREQ 后,若本地有到 D 的有效路由,则直接沿反向路径向 S 发送 RREP,否则利用相同方法,计算与邻居节点(比自己离 D 近的节点)的 mp 值,选择 mp 值最小的节点作为下一跳转发 RREQ;

(4)D 收到 RREQ 后,沿反向路径向 S 发出 RREP。

上述过程较 AODV 协议的主要不同如下:节点不采用洪泛 RREQ 的方式,而是贪婪地向与自己具有最小相对运动趋势的邻居节点转发 RREQ。

本文算法存在一种特殊情况,即局部优化问题^[10],如图 3 所示。

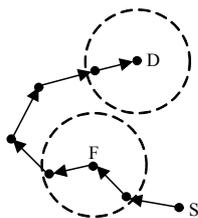


图 3 局部优化问题

在图 3 中,虚线表示节点的最大通信范围,F在一跳范围

内找不到比自己距离 D 更近的节点,根据贪婪算法,F会选择自己作为分组转发的下一跳,而使分组无法到达目的节点 D(此时存在到 D 的路径,如图 3 箭头所示),此时,F称为最佳主机^[10]。为防止局部优化的产生,本文协议采用与 f-GEDIR^[11]协议相同的处理方式:最佳主机洪泛 RREQ,邻居节点收到 RREQ 分组后先检查是否收到过该 RREQ,若是则丢弃该 RREQ,否则继续使用贪婪算法寻找下一跳。

4.3 路由维护阶段

本文协议扩展了 AODV 路由维护过程中使用的 Hello 报文,邻居节点间交换 Hello 消息时捎带了坐标、速度、运动方向等信息,节点通过邻居列表维护这些信息以便路径发现时使用。此协议禁用了本地路由修复功能,若某条路由在使用过程中出现了断链,则断链处上游节点不再负责本地修复,而是采用直接向源节点 S 发送 RERR 的方式,使其重启路由发现过程,这样做原因在于:节点移动性强,本地修复失败几率大,使用本地路由修复反而会导致分组传输时延增加、传输效率降低。

5 仿真分析

5.1 仿真环境

为了验证 GAODV 协议的有效性,本文使用在无线网络仿真中应用较广泛的 GloMoSim 作为仿真平台,它提供了传统 AODV 协议的源代码,仿真参数设置如表 1 所示。

表 1 仿真参数

参数	值
节点数/个	100
最大运动速度/(m·s ⁻¹)	20, 40, 60, 80, 100
信号衰减模型	TWO-RAY
发包率/(packet·s ⁻¹)	1
区域大小/m ²	5 000×5 000
MAC 协议	IEEE802.11 DCF
业务类型	CBR
节点传输半径 ra/m	250
移动模型	Random Waypoint
带宽/(Mb·s ⁻¹)	2
数据包大小/Byte	512
仿真时间/min	10

5.2 仿真结果及性能评估

本文提取 CBR 业务吞吐量及数据分组端到端平均时延作为协议性能比较的依据。对 AODV 协议和 GAODV 协议,分别在节点最大移动速率为 20 m/s, 40 m/s, 60 m/s, 80 m/s 和 100 m/s 时进行仿真实验。

图 4 给出了吞吐量与节点最大移动速度的关系。实验结果表明,当节点移动速度增大时,由于节点间链路失效率的增加,导致引起协议耗费较多信道资源在路由发现和错误报告上,因此 2 种协议数据业务吞吐量都有下降趋势,但 GAODV 的吞吐量比 AODV 高,原因是虽然链路失效率增加,但 GAODV 采用的贪婪算法仍然有效限制了路由控制报文的数量,使网络能更多地传递数据分组,尤其当速度大到一定程度时,该优势会更明显。

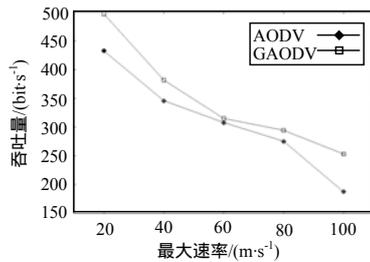


图4 CBR业务吞吐量

图5比较了GAODV和AODV的端到端平均时延,可见,随着节点移动速度的不断增大,GAODV的平均时延低于AODV,原因如下:(1)由于GAODV利用了贪婪算法,节点在路由发现阶段尽量选择运动方向大致相同的节点作为下一跳,因此容易找到较稳定的路由,减少了重启路由发现的次数,进而减少数据分组的丢失,因此,对整个网络而言,降低了数据分组传递的端到端时延。(2)GAODV禁用了本地路由修复。当节点移动速度足够大时,链路失效率的增加将引起本地修复失败的几率增加,因此,AODV采用的本地修复将消耗大量路由修复和重建时间,导致数据分组传输的端到端时延增大。

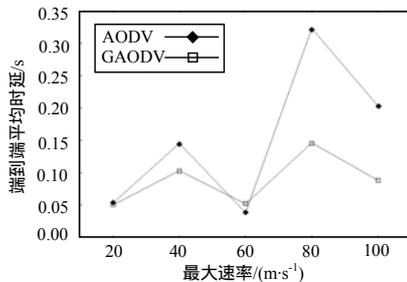


图5 数据分组端到端平均时延

6 结束语

由于移动 Ad Hoc 网络中节点的能量、存储容量有限,因此必须在完成任务的同时提高协议效率和性能。例如在军事应用方面,将无线传感器节点置于己方车辆、坦克等装备上,节点在随部队高速移动的同时,需要完成自动组网、动

(上接第95页)

5 结束语

本文提出的 MCF-LDLP 混合调度算法丰富了优先级确立的依据,在一定程度上提高了控制系统的整体性能,尤其是在网络负载较重时,仍能使系统保持较好的控制性能。下一步工作将研究如何更好地建立优先级与控制系统性能之间的映射关系,以期进一步提升 NCS 的整体性能。

参考文献

[1] Octavian V. Networked Control Systems[D]. Durham, NC, USA: Duke University, 2001.
 [2] Zhang Wei, Branicky M S, Philips S M. Stability of Networked Control Systems[J]. IEEE Control Magazine, 2001, 21(1): 84-99.
 [3] Zuberi K M, Shin K G. Design and Implementation of Efficient Message Scheduling for Controller Area Network[J]. IEEE Trans.

态调整网络拓扑、传输战场实时信息等任务。本文提出基于AODV的贪婪路由协议,利用节点运动参数,在节点高速运动的环境下建立较稳定的路由,提高了网络吞吐量,减小了分组传输的端到端平均时延,且具有较小路由控制开销,在延长节点工作时间、提高节点生存能力方面具有一定参考价值。后继工作将进一步简化路由算法,以提高协议的数据业务吞吐量。

参考文献

[1] 于宏毅. 无线移动组织网[M]. 北京: 人民邮电出版社, 2005.
 [2] 洪锡军. 无线自组网路由协议研究[J]. 计算机工程, 2005, 31(8): 105-107.
 [3] Charles E, Elizabeth M P. Ad-hoc On-demand Distance Vector Routing[S]. IETF RFC 3561, 2003.
 [4] Ye Zhenqiang, Srikanth V, Krishnamurthy S K, et al. Framework for Reliable Routing in Mobile Ad Hoc Networks[C]//Proc. of INFOCOM'03. California, USA: [s. n.], 2003: 270-280.
 [5] 冯玉美. 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.
 [6] 年梅. 通过链路失效预测机制提高 AODV 协议的性能[J]. 计算机应用, 2005, 25(6): 1251-1256.
 [7] Rapport T. Wireless Communication: Principles And Practice[M]. New Jersey, USA: Addison Wesley/Pearson, 1996.
 [8] Maltz D A. The Effects of On-demand Behavior in Routing Protocols for Multihop Wireless Ad Hoc Networks[J]. IEEE Journal on Selected Areas in Communications, 1999, 17(8): 1439-1453.
 [9] Jain R. Geographical Routing Using Partial Information for Wireless Ad Hoc Networks[J]. IEEE Personal Communications, 2001, 8(1): 48-57.
 [10] 于宏毅. 无线移动组织网[M]. 北京: 人民邮电出版社, 2005: 227-228.
 [11] Stojmenovic I, Xu Lin. Loop-free Hybrid Single-path/Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks[J]. IEEE Transactions on Parallel and Distributed Systems, 2001, 12(10): 1023-1032.
 on Computers, 2000, 49(2): 182-188.
 [4] Gregory C. Scheduling of Networked Control Systems[J]. IEEE Control System Magazine, 2001, 21(1): 57-65.
 [5] Gregory C. Stability Analysis of Networked Control Systems[J]. IEEE Trans. on Control Systems Technology, 2002, 10(3): 438-446.
 [6] Jose Y, Marti P, Josep M. Control Loop Scheduling Paradigm in Distributed Control Systems[C]//Proc. of the 29th Annual Conf. of the IEEE. Roanoke, USA: [s. n.], 2003.
 [7] Seung H. Scheduling Algorithm of Data Sampling Times in the Integrated Communication and Control Systems[J]. Control Systems Technology, 1995, 3(2): 225-230.
 [8] Seung H. Bandwidth Allocation Scheme for Cyclic-service Fieldbus Networks[J]. Mechatronics, 2001, 6(2): 197-204.