

基于 AODV 的长路由探测研究

陶洋, 李勇, 陈虹璇

(重庆邮电大学软件技术中心, 重庆 400065)

摘要: 针对移动 Ad Hoc 网络中现有 AODV 路由协议中存在的长路由现象, 提出一种路径优化策略, 以缩短路由长度, 避免优化过程中不必要的通信中断, 实现无缝切换。通过 NS2 仿真证明, 在现有带宽和反应时间内可以提高网络的性能, 没有增加太多额外的资源耗费, 端到端延迟和分组传输率等性能都有提高。

关键词: 移动 Ad Hoc 网络; 长路由; 路径优化策略

Research on Long Routing Detection Based on AODV

TAO Yang, LI Yong, CHEN Hong-xuan

(Center of Software Technology, Chongqing University of Posts and Telecommunications, Chongqing, 400065)

【Abstract】 This paper presents a method of Path Optimizing Strategy(POS) to solve the phenomenon of long routing, which is existing in Mobile Ad Hoc Network(MANET) of AODV routing protocol. The results of NS2 simulation prove that POS improves the network performance without increasing the additional resources cost too much in the existing bandwidth and response time, such as end-to-end delay and packet transmission rate.

【Key words】 Mobile Ad Hoc Network(MANET); long routing; Path Optimizing Strategy(POS)

1 概述

移动 Ad Hoc 网络(Mobile Ad Hoc Network, MANET)不需要固定的基础设施, 不需要预先配置主机, 能在任何时间、任何地点快速组建一个移动通信网络^[1]。但节点的任意移动及网络拓扑结构的动态变化等特点也给无线自组网的路由设计带来了许多新的问题和挑战。

AODV 协议是结合 DSDV 协议和 DSR 协议的产物, 是一种比较经典的按需路由协议。但 AODV 算法目前还存在一些不足, 如在现有的 AODV 协议中, 节点的路由表中存储着下一跳的节点信息, 并且只要该路由保持活性就一直使用它。但是, 路由越长, 耗费的带宽越多, 无法连通的可能性就越大, 其后果是严重影响网络性能并耗费大量的资源, 产生长路由现象^[2]。

本文提出了一种基于 AODV 的路径优化策略(Path Optimizing Strategy, POS)解决这个问题: 如果有一条较短的路由存在, 就尽可能使其变短, 并避免优化过程中不必要的通信中断, 实现无缝切换。

2 长路由现象

随着移动 Ad Hoc 网络中节点的增多, 网络变得越来越大, 因此, 降低路由的平均跳数和提高路由的可靠性变得越来越重要。在现有的 AODV 路由协议中, 节点的路由表中存储着下一跳的节点信息, 并且只要该路由保持活性就一直使用它。同样, MANET 中节点的移动会使网络形状有很大的改变, 而这时只要链路不断, 路由就会一直保持它的连通性, 导致包头变得很长, 严重影响网络性能并耗费大量的资源。

现假设在一个使用 AODV 路由协议的无线网络中, 经过 RREQ 阶段和 RREP 阶段之后建立了一条最短的路由(S, 1, 2, 3, 4, 5, D), 一个包从源节点 S 到目的节点 D 要经过 6 跳。图 1 是节点移动后的路由, 表明存在的路由已不是最优(即最短), 但路由仍然保持完好。此时, 节点 2 在 S 的传输范围内,

节点 4 在节点 2 的传输范围内, 同样, D 在节点 4 的传输范围内。因为原始路由并没有中断, 所以路由表的条目并没有更新, 即一个包从 S 到 D 仍然需要通过 6 跳的路由。但是从理论上来说, 从 S 到 D 的最短路由应该只有 3 跳, 见图 1。

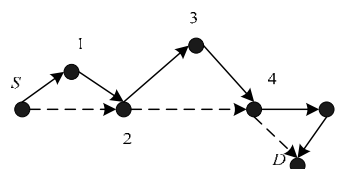


图 1 节点移动后的路由

从可靠性上来讲, 路由越长, 中断的可能性越大, 其边缘效应就越明显; 路由越长, 节点数越多, 路径的平均链路距离就越向传输半径逼近。在这种情况下, 节点的少许移动便可导致链路断开, 路径相当脆弱。

因此, 本文提出 POS 策略解决此问题, 以缩短活性路由的长度, 提高整个网络可靠性, 节约网络资源。

3 POS 基本思想

POS 的基本目标就是在可行时发现较短的路由并通过修改路由表实现长短路由的切换, 其最基本的步骤如图 1 所示, 例如节点 2 在 S 的传输范围内, 路径 S12 可以被简化为 S2。

POS 的基本思想就是基于 AODV, 在路由表中新增一个数组类型作为一个路由条目, 用于存储探测包信息; 在路由表中扩充 tempnexthop, 用于存储临时路由; 在路由表中新增一个布尔变量 temproute, 用于判断数据包传输所走路由; 在消息体系中扩充图 2 所示的探测包 DRREQ, 用于定期检测有无链路捷径出现。POS 的基本流程是在维护路由时由源节

作者简介: 陶洋(1964-), 男, 博士后, 主研方向: 通信协议及软件; 李勇, 硕士研究生; 陈虹璇, 助教

收稿日期: 2009-04-18 **E-mail:** liyong1234@gmail.com

点发出探测包 DRREQ ,通过对 DRREQ 中记录跳数和活性路由上相关节点跳数的比较发现是否有路由捷径。当一条捷径被发现后,修改相关节点的路由表实现这条捷径。

源节点地址 <i>SrcA</i>
目的节点地址 <i>DesA</i>
前驱节点地址 <i>PreA</i>
下一跳节点地址(<i>NextA</i>)
跳计数器 <i>HopC</i>
DRREQ 序列号(<i>Times</i>)

图 2 DRREQ 结构

4 POS 的设计实现

在 POS 策略中,每个 DRREQ 除了包含必需的目的地址 *DesA*、源地址 *SrcA* 和前驱节点地址 *PreA* 外,还必须在其包头中增加一个跳计数器 *HopC*(hop-count)、下一跳节点地址 (*NextA*)和 DRREQ 序列号(*Times*)。 *HopC* 在源节点被初始化为 0,以后每经过 1 跳,就增加 1;当转发给下一跳节点时,记录当前节点路由表中的下一跳作为包中 *NextA* 值; *Times* 值在第 1 次发送时初始化为 1,以后每次重新发送时,增量 1 即可,即每一个 DRREQ 包中存储的信息条目为 $(SrcA, DesA, HopC, PreA, NextA, Times)$ 。

设一个源节点 *SrcA* 和目的节点 *DesA*,在 *SrcA* 发送一个探测包到第 1 跳节点之前,DRREQ 条目表示为 $(SrcA, DesA, 0, SrcA, NextA, Times)$ 。设当前节点为 *i*,前驱节点为 *j*,那么,对于当前节点(节点 *i*)接收到的包,给出相关算法:

```

Procedure Packet process
If (接收到的分组是数据分组类型){
If (路由选择标识为真)
Then (通过临时路由转发数据分组)
Else (转发数据分组)
}
If (接收到的分组类型是 DRREQ 探测包){
If (当前节点是目的节点)
Then (DRREQ 再传一跳)
Else {
If (探测包中 NextA=当前节点地址)
Then (当前节点记录信息(SrcA, DesA, HopC, PreA, NextA, Times), 继续转发)
Else {
If (当前节点中并不存在一组与(SrcA, DesA)相对应的条目)
Then (记录信息条目(SrcA, DesA, HopC, PreA, NextA, Times))
Else
If(探测包和当前节点中 Times 值相等){
If(探测包中 HopC-当前节点 HopC > 2)
Then {在当前节点 i 的 Pre 节点中建立临时路由,设置路由选择标识值,修改路由表,使得其下一跳地址改为当前节点的地址;修改当前节点 i 的下一跳地址为转发探测包到 i 的节点地址;重置路由选择标识值,删除临时路由。}
Else (记录信息条目(SrcA, DesA, HopC, PreA, NextA, Times))
}}}
}
}

```

5 POS 的特殊情况

在上述过程中可能存在 2 种特殊的情况:

(1)包含源节点的捷径。如图 3 所示, *B* 是源节点,初始化为 $(B, I, 0, B, C, 1)$, DRREQ 值为 $(B, I, 1, B, C, 1)$ 。同样,节点 *D* 的存储值为 $(B, I, 2, C, 1)$, DRREQ 值为 $(B, I, 3, D, E, 1)$ 。当节点 *B* 收到 *D* 发出的 DRREQ 时,比较 *HC* 的差值为 3,大于 2,所以,存在一条捷径,那么节点 *B* 的前驱节点修

改路由表使得其下一跳节点指向 *B*,即节点 *B* 的下一跳节点变为 *B*,但是节点 *B* 的实际下一跳节点为 *C*,修改后应该指向 *D* 才能发现捷径,但是这并不会产生错误,因为紧接着节点 *B* 就会修改 *B* 的路由表,使 *B* 的下一跳节点指向 *D*,从而实现捷径。在这种情况下,前一步的操作是冗余的,但是不会带来任何错误。

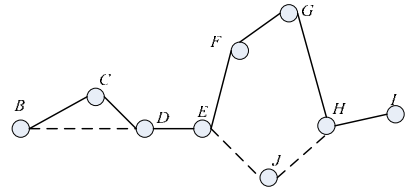


图 3 包含源节点的捷径

(2)包含目的节点的捷径。如图 4 所示,节点 *H* 是目的节点。如果 DRREQ 传送到节点 *H* 时就停止了,那么捷径 *EJH* 是不可能实现的,这时需要目的节点 *H* 将 DRREQ 再发送一跳,DRREQ 才能到达节点 *J* 并与其比较,从而实现捷径 *EJH*。

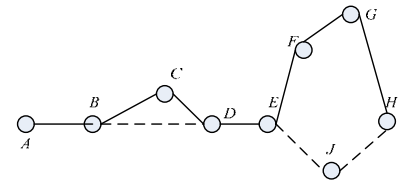


图 4 包含目的节点的捷径

6 仿真

6.1 仿真场景

本文选取 NS2 作为仿真平台^[3]。仿真参数为:节点随机分布在 $1200\text{ m} \times 1200\text{ m}$ 的矩形区域中,共 50 个节点进行仿真。开始仿真后,节点在暂停时间内保持静止,然后随机选择一个目的地,以 0 到最大速度之间的某个速度向目的地移动,到达目的地后在暂停时间内暂停,再随机选定另一个目的地,重复前面的过程。在整个仿真过程中,节点一直在重复上述过程,仿真时间为 800 s。定义最高移动速度为 20 m/s。

仿真采用 CBR (Constant Bit Rate)流量源,以每秒 5 个包的速率发送,每个包的大小均为 512 Byte,网络中含有 10 个 CBR 源的通信模式。为测试不同的移动性能,使用 0 s, 100 s, 200 s, 300 s, 400 s, 500 s, 600 s, 700 s 和 800 s 等不同的暂停时间(pause time)。

6.2 仿真结果分析

图 5 是端到端传输延迟时间的比较,POS 的延迟时间比原 AODV 协议小,这是因为 POS 在数据传输的过程中优化了路由,缩短了数据传输的平均距离,减少了跳数,使它在延迟时间上远远优于原协议。

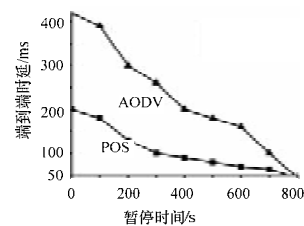


图 5 端到端传输延迟比较

数据传输率(图 6)是目的节点收到的数据包数量与 CBR 源节点发出的数据包数量的比值。

(下转第 116 页)