

基于 MAODV 的移动多播路由协议的改进 与仿真^①

Improvement and Simulations of Mobile Multicast Routing Based on MAODV

杨润萍 杜世民 (宁波大学 科技学院 浙江 宁波 315212)

汪小兰 (深圳华为技术有限公司 核心网 PDT 经理管理部 广东 深圳 518129)

摘要: 移动 Ad hoc 网络是指由一组带有无线收发装置的移动节点组成的一个多跳的、不需要固定中心接入点或者基站支持的自治系统。对移动 Ad hoc 网络多播路由中的 MAODV 协议进行理论分析,提出了一种使用转接机制改进 MAODV 协议的方法,该改进协议使用新的 NACK 抑制机制来防止 NACK 风暴,减少了移动 Ad hoc 网络中大量的路由重构和数据分组的重传。采用 OPNET 软件进行仿真,仿真结果证明了该改进协议在 Ad hoc 网络中的可行性和有效性。

关键词: Ad hoc 多播路由 MAODV OPNET 仿真

1 引言

移动 Ad hoc 网络是由一组带有无线收发装置的移动节点组成的一个多跳的临时性自治系统。随着无线通信技术的不断发展,移动 Ad hoc 网络以其组网灵活,快捷等特点,在军用和紧急救援等特殊环境中有着越来越广阔的应用前景。

由于移动网络需要面临拓扑结构动态变化、存在单向信道、有限的无线传输带宽以及移动终端能力的局限性等一系列挑战,因此对移动自组网多播路由协议的研究必然担负着更重的任务。本文在 MAODV 协议的基础上,提出了基于转接机制的移动 Ad hoc 网络的多播路由协议,它使用新的 NACK 抑制机制来防止 NACK 风暴,减少了分组丢失和大量的不必要的分组传送。

2 多播路由协议^[1]

多播路由协议设计的基本思想是以最少的冗余,将多播数据分组传送至多播组的每个成员节点,在协议设计的过程中,需要考虑节点的移动性的影响、可能产生的环路、路由的非理想、是否按需创建、路由

更新、控制分组的传送方式(全网洪泛还是受限于成员节点的传播)等问题。

迄今为止,人们已经提出了多种以移动 Ad hoc 网络为应用环境的多播路由协议,它们在寻路机制和路由结构上各有不同,可以根据通往组成员的路由产生方法不同可把它们分为三类:基于树的多播路由、基于网格的多播路由、混合的多播路由。

在基于树的协议中,多播路由使用基于源树或者共享树的方法,任意一对节点间只能有唯一的一条路径。其优点是初始化多播树所需要的带宽资源通常比其它结构要小。缺点是多播树容易受到链路/网络节点失效和节点移动性的影响。另外,基于树的多播路由协议由于需要维护多播树,所以控制开销比较大。

对于网格型的多播路由协议,创建网格所需要的资源数量(带宽,处理的负载,存储的负载)通常比创建基于树状结构的多播协议要多^[2]。基于网格的路由还有另外一个缺点,就是容易出现路由环路,所以必须采取特殊的措施来防止该现象的出现。它最大的优势是在多播会话中存在多条冗余的路由来保证链路失败和节点移动后的健壮性。所以基于网格的多播路由协

^① 收稿时间:2009-04-05

议更适用于网络拓扑变化很快的严酷环境。

混合型路由协议取树状和网格二者之长, 适合于中等移动程度的移动自组网环境。

IETF 的 MANET 工作组推荐两种应用环境的多播路由协议, 即 MAODV 和 ODMRP。由于树状结构多播路由协议在多播树的初始化所需要的带宽资源比其它结构要小, 树状多播路由协议具有很高的分组转发效率, 同时 MAODV 协议和单播 AODV 协议有很紧密的依赖关系, 本文以 MAODV 协议为研究对象。

3 MAODV协议分析

MAODV 是单播路由协议 AODV 的多播扩展, 它是通过广播路由发现机制的基于树的按需多播路由协议。它为每一路由表项使用了目的地序列号, 序列号确保无路由环路, 并解决了计数到无穷问题。

MAODV 是采用双向共享树的多播分发机制^[3], 多播分发树是按需创建的, 而且 MAODV 多播树中的组成员都可以受理加入组申请。“双向”是指多播树中的节点同时设置指向上游节点(父节点)和下游节点(子节点)的指针。MAODV 的多播树所有节点可分为三类: 组成员、路由器和组长, 组长负责为自己所在的组维护一个组序列号, 定期向全网广播组联络报文(GRPH), 收到 GRPH 报文的节点将获悉该组的存在。组长每次广播 GRPH 报文前, 先将组序列号加 1, 这样越大的组序列号就对应着越新的组状态。

移动多播路由协议为保证传输的准确性需要进行差错检测, 方法通常是接收者对数据分组进行确认来表示是否所有分组都正确传递到所有接收者。

但是如果采用 ACK 反馈机制, 可能给发送者带来大量的 ACK 消息, 也极大地消耗了网络资源。NACK 机制可以只反馈未接收到的分组信息, 这样在丢失率不高的情况下就相对节约了网络带宽, 然而在丢失率较高、多播组很大时, 仍存在大量反馈的问题, 并且还需要有相关机制来防止 NACK 风暴。

4 MAODV协议的改进

4.1 协议改进原理

本文提出使用转接机制改进 MAODV 协议, 主要方法是在发送者和接收者的路径上设定一个协议转接节点, 每个转接节点在它的内存中设定有限的缓冲区

存放转发分组, 并且每个分组都有一个序列号。当一个转接节点检测到一个分组丢失时, 它向上级节点发送 NACK 消息, 上级节点仅仅在它的下级节点中对该丢失的分组进行多播。转接节点仅仅管理它的上级节点和数量有限的下级节点, 并且允许任意一个下级节点当检测到分组丢失时可以直接向上级节点发送 NACK 消息, 且只能向上级节点发送 NACK 消息, 上级节点只接收它的下级节点发送来的 NACK 需求, 这样就减少了网络中 NACK 需求消息的传播, 从而有效地抑制了 NACK 风暴。

(1) 下级节点的处理过程

当一个转接节点检测到一个分组丢失时, 以单播方式向它的上级节点发送 NACK 需求重传消息, 要求上级节点重传相应的分组, 并同时设置相应的定时器, 此时, 它对当前接收到的分组进行重新编号后继续将分组向下转发, 下一级转接节点就检测不到分组丢失情况, 从而使得分组重传仅仅在一跳节点之间进行。

(2) 上级节点的处理过程

上级节点收到下级节点传送来的 NACK 消息后, 依据其中的信息决定是否重传相应的分组, 上级转接节点仅仅应答第一个到达的 NACK。当应答第一个到达的 NACK 时, 上级转接节点就开始重传分组并设置一个定时器, 该定时器的大小可以根据不同的网络进行设置。

(3) 节点移动时 NACK 的处理过程

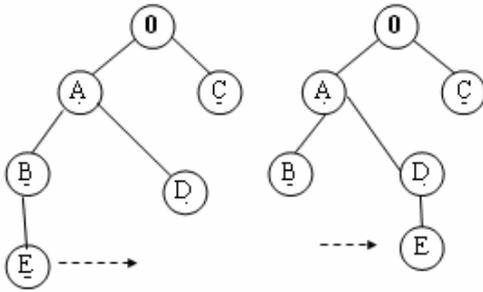
在移动 Ad hoc 网络环境中由于节点的移动, 链路的失败率是非常高的。图 1 是移动节点 E 移出节点 B 的传输范围, 导致链路失败时 NACK 分组丢失的处理情况, 由于节点 E 的迁移, 导致节点 B 在相应的定时器内无法收到所要重传的数据分组, 这时节点 E 就向节点 D 重传 NACK 分组信息, 节点 D 将 NACK 信息转发至节点 B, 要求重传数据分组, 使得节点 B 就将重传数据传到节点 4, 节点 D 就可以将重传数据传到节点 E。

4.2 改进 MAODV 协议的性能分析

如果每个节点不使用协议转接机制, 节点不存在有缓冲区及 NACK 机制, 传输一个数据分组后将暂停, 在这种情况下, 一个数据分组第一次传输成功的延迟^[4]为:

$$T_1 = T_{data} \times P_{suc} \quad (1)$$

第 N 次数据分组传输失败, 第 N+1 次传输成功的延迟为:



0 多播源节点 A-E 多播组成员

图 1 节点移出范围时的处理过程

$$T_{n+1} = \sum_{k=1}^n \{ (K \times T_{timeout} + T_{data} + T_{nack}) \times P_{fail}^k \} \times P_{suc} = \quad (2)$$

$$\left[\left(\frac{P_{fail} \times (1 - P_{fail}^n)}{(1 - P_{fail})^2} - \frac{n \times P_{fail}^{n+1}}{1 - P_{fail}} \right) \times T_{timeout} + \frac{(T_{data} + T_{nack}) \times P_{fail} \times (1 - P_{fail}^n)}{1 - P_{fail}} \right] \times P_{suc}$$

每个节点使用协议转接机制的情况下，每个节点可使用缓冲区和 NACK 机制，一个数据分组第一次传输成功的延迟为：

$$T_1 = T_{data} \times P_{suc} \quad (3)$$

一个数据分组传输失败和 NACK 传输失败的延迟为：

$$T_1' = (T_{data} + 2 \times T_{nack}) \times P_{fail} \quad (4)$$

第 N 次数据分组传输失败，第 N+1 次传输成功的延迟为：

$$T_{n+1} = \sum_{k=1}^n \{ (T_{data} + 2 \times T_{nack}) \times P_{fail}^k \} \times P_{suc} \quad (5)$$

其中，各个参数的意义如下：

P_{suc} ：没有链路失败时传输数据和 NACK 分组时的传输率；

P_{fail} ：链路失败时传输数据和 NACK 分组的传输率；

$T_{timeout}$ ：暂停时间；

T_{data} ：数据分组的传输延迟；

T_{nack} ：NACK 分组的传输延迟。

由于 $K \times T_{timeout}$ 是一个线性值，随着加入多播组的节点数 K 的增大而增大，则关系 $K \times T_{timeout} > T_{nack}$ 就会一直成立，因此不使用协议转接机制的 MAODV 协议的传输延迟会大于使用改进协议的传输延迟。

4.3 协议实现的核心程序代码

(1) 节点加入多播组的过程

如果有节点通过发送 RREQ 控制消息来请求加入多播组，通过调用 maodv_join 函数来进行处理：

```
static void maodv_join(int grp_addr)
{
    mrt_table *group_rec;
    group_rec = mrt_table_find(grp_addr);
    if(group_rec == NULL)
    {
        group_rec = mrt_table_insert(grp_addr);
    }
    if(!group_rec.is_router)
    {
        /* 没有到多播组的路由，发起 join RREQ 请求 */
        rreq_route_discovery(grp_addr, RREQ_JOIN,
        NULL, 0);
    }
    else if(!group_rec.is_member)
    {
        /* 本节点已经是多播数据转发节点，所以将其变为多播组成员 */
        group_rec.is_member = 1;
    }
}

(2) 节点退出多播组的过程
节点退出多播组的过程通过调用 maodv_leave 函数来实现：
static void maodv_leave(int grp_addr)
{
    mrt_table_t *group_rec;
    group_rec = mrt_table_find(grp_addr);
    /* 本节点不是多播树成员，不做任何动作 */
    if(group_rec == NULL
    || !group_rec.is_router)
    {
        return;
    }
    /* 退出多播组 */
    group_rec.is_member = 0;
    if(group_rec.am_leader)
    {
        if(group_rec.activated_downstream_cnt > 0)
        {
            /* 有 DOWNSTREAM 方向下一跳，选择一个作为多播组组长 */
            mact_make_downstream_leader(group_rec);
        }
    }
    group_rec.am_leader = 0;
    mrt_table_stop_routing(group_rec);
}
else
if(group_rec.activated_downstream_cnt == 0)
```

```

/* 没有 DOWNSTREAM 节点, 剪除自己, 路由表无效 */
mrtnexthop_t *upstream;
upstream= mrt_table_get_activated_upstream(group_rec);
if(upstream != NULL)
{mact_send(MACT_PRUNE, 1, grp_addr, upstream.addr,1);
mrtnexthop_deactivate(group_rec, upstream.addr);}
mrt_table_stop_routing(group_rec);
/* 删除该多播组缓存的所有 RREQ 数据包 */
seek_list_remove(grp_addr);
/* 删除缓存的多播数据包 */
packet_queue_drop(grp_addr);
/* 删除所有下跳节点表 */
mrtnexthop_remove_all(group_rec);
}

```

5 仿真结果及分析

对移动 Ad hoc 网络路由协议仿真的目的是对采用 MAODV 以及改进后的路由协议的移动 Ad hoc 网络的通信性能进行全面、深入的分析评估, 对网络中涉及的技术参数进行讨论。

按照移动 Ad hoc 网络背景来设计仿真环境^[5], 节点模型包含应用层模块, 路由层模块, MAC 层模块, 物理层模块构成, 其中路由层模块按照 MAODV 路由协议的要求来设计, MAC 层模块采用 802.11 中 CSMA/CA 的机制, 物理层为双向无线链路, 采用跳频扩频技术。50 个节点随机分布在 $1000 \times 500\text{m}^2$ 的矩形区域中, 每个节点采用 Random Way-point 移动模型。

在构建的移动 Ad hoc 网络仿真系统中, 网络的平均节点度为 4, 链路传输丢失率在 $[0, 0.5]$ 之间随机分布, 数据分组和 NACK 分组传输延迟在 $[5, 30\text{ms}]$ 内服从指数分布。

图 2 为假定缓冲区的大小为 5, 传输 100 个数据分组, 不使用协议转接与使用协议转接机制时不同重传数据分组总的传输延迟的比较, 从图中可以看出每个节点使用协议转接机制的性能明显好于每个节点不使用协

议转接机制, 随着网络传输失败率的增大, 其每个节点使用协议转接机制的重传方式的性能更好。

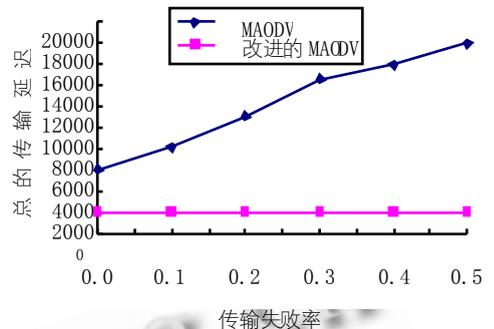


图 2 总的传输延迟与重传方式

图 3 为不同缓冲区大小时传输失败率与总的传输延迟的关系, 当缓冲区较大时, 其总的传输延迟较低, 而且传输失败率的增大对总的传输延迟影响不大。缓冲区大小为 7 与缓冲区大小为 10 其总的传输延迟比较接近, 当缓冲区的大小接近 10 时, 传输延迟将不会再减少了, 从而可以建议缓冲区可以设置在 7 至 10 之间, 如果更多, 就浪费了内存资源。

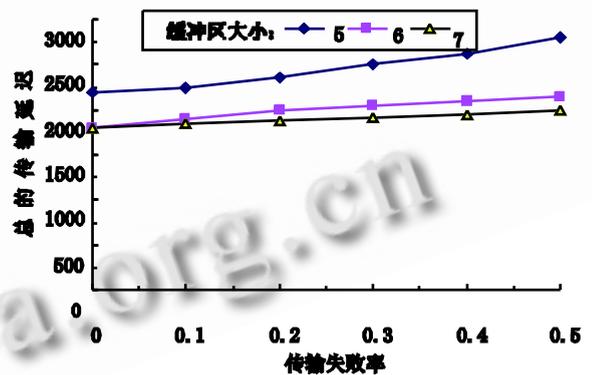


图 3 传输延迟与缓冲区大小

图 4 为不同缓冲区时传输失败率与平均分组延迟的关系, 当缓冲区较大时, 其单个分组在缓冲区内存留的时间就变得长, 从而其平均分组延迟就高, 缓冲区大小为 7 与缓冲区大小为 10 其平均分组延迟亦比较接近。

综合图 3 与图 4, 可以看出其缓冲区的大小可以设置为 7。模拟实验结果显示, 在一定移动性和通信流量下, 带有转接机制的 MAODV 路由由协议表现出更好的可靠性。

(下转第 116 页)

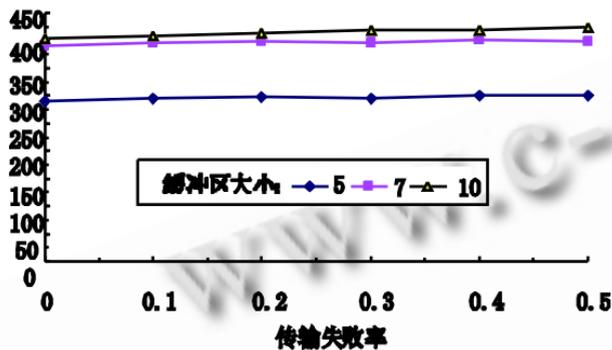


图 4 平均分组延迟与缓冲区大小

5 结语

本文介绍了移动 Ad hoc 网络的特点；重点阐述了 MAODV 多播路由协议的特点，在此基础上提出了对 MAODV 进行改进的方法；同时采用了目前业界一直推崇的网络仿真软件 OPNET modeler 对改进的协议进行仿真，对移动 Ad hoc 网络的多播路由协议进行了实验性研究。

移动 Ad hoc 网络多播路由是当前网络领域的前沿性研究课题。随着新一代互联网络技术的不断发展，该方向的课题的研究将会出现一些新的研究热点。

参考文献

- 1 Luo JH, Xue L, Ye DX. Research on multicast routing protocols for mobile ad-hoc networks. Computer Networks, 2008,52:988 – 997.
- 2 De Morais Cordeiro C, Gossain H, Agrawal DP. Multicast over wireless mobile ad hoc networks: present and future directions. IEEE Network, 2006,17(1):52 – 59.
- 3 李昌兵. 结合分布式与集中式特点的动态多播路由算法. 计算机系统应用, 2008,6:62 – 66.
- 4 王金龙, 王呈贵, 吴启辉, 龚玉萍. Ad Hoc 移动无线网络. 北京: 国防工业出版社, 2004. 1 – 8.
- 5 陈敏. OPNET 网络仿真. 北京: 清华大学出版社, 2004. 1 – 280.