

中文摘要

随着无线通信技术和计算机网络技术的发展，无线移动网络由于其灵活性和实用性近年来受到越来越多的关注，同时，由于组播技术的发展，如何将无线局域网技术和组播应用相结合也成为新兴的研究课题，在此背景下，各种 Ad hoc 网络路由协议的开发及相关的性能分析评价成为学术界的研究热点。本文的工作即由此展开。大致安排如下：

首先对 Ad hoc 技术及网络路由协议进行综述，介绍了网络路由协议的分类以及 Ad hoc 网络的一些专用路由协议；按需路由和组播的定义及主要相关概念；重点分析了 Ad hoc 组播网络中具有相对较优性能的 ODMRP 路由协议的机制。

其次，在对多种协议分析研究的基础上，将 ODMRP 作为重点研究对象，并利用软件仿真的方式，实现了基于 Ad hoc 网络的 ODMRP 路由协议改进工作。提出了提高 ODMRP 性能的 EODMRP 协议，该协议通过使用链路状态预测，路由和转发组可以在拓扑改变前重建。对刷新时间的选择避免了不必要的控制包的传输而引起的带宽浪费。本文提出的新路由选择算法能够使路由在最长时间内保持有效。从而进一步减少了控制包引起的负载。被动应答和重传机制的应用则提高了应答传输的可靠性。

本文还提出了减少获得路由延时的方法。改进后的路由协议以延迟作为路径规格的度量，通过主动探测的方式感知路径的状态，从而达到充分利用 Ad hoc 网络资源、提高通讯质量的目的。

仿真结果表明：EODMRP 提高了组播环境下的路由效率和数据包的端端投递率，同时降低了端端延迟，对 QoS 业务性能也有很大的提高。总之，改进后的 EODMRP 对于移动网络有着更好的适应性。

论文最后简单讨论了无线 Ad hoc 网络环境下的 QoS 保证机制，并将此作为今后研究的方向。

关键词：MANet，组播，ODMRP，EODMRP，QoS

ABSTRACT

With recent development in computer and wireless communication technology, MANet (mobile ad hoc network) is increasingly widespread used and applied due to its flexibility and utility convenience. Multicast plays an important role in MANet. A critical challenge in the design of multicast in the MANet is the development of efficient routing protocols that can provide high-quality communication.

This paper describes the concept of MANet and introduces the classification of the specific routing protocols in the MANet, especially analyses the performance of on-demand multicast routing protocol (ODMRP).

Based on ODMRP, a new ad hoc multicast routing protocol-EODMRP (Enhanced ODMRP) is proposed. This novel on-demand protocol uses movement and link state prediction to rebuild routing and forwarding group before the changing of topology. The new size of flush timer eliminates the bandwidth overhead caused by unnecessary control packets transmission. The improved routing algorism could make the route entries valid as long as possible, which in return reduces the load of control packets. Passive acknowledgement and retransmission are also introduced to improve the route reliability. To reduce the route acquirement latency, the probing-based method to get the link state is also proposed, which will make full use of the resource of ad hoc network and improve the quality of communications.

Simulation result in the laboratorial experiments show that EODMRP outperforms the ODMRP in terms of network overhead, route acquisition latency and utilization of available bandwidth. The enhanced ODMRP is feasible to improve the network QoS.

Finally, the QoS mechanism in ad hoc network is discussed, which is expected to be implemented in the future work.

Keywords: MANet, multicast, ODMRP, EODMRP, QoS

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 天津大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名： 签字日期： 年 月 日

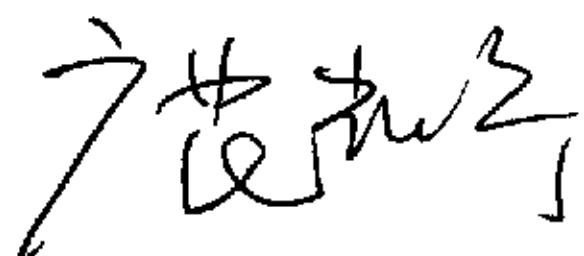
学位论文版权使用授权书

本学位论文作者完全了解 天津大学 有关保留、使用学位论文的规定。特授权 天津大学 可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：

导师签名：



签字日期： 年 月 日

签字日期： 02 年 12 月 25 日

第一章 前 言

1.1 无线局域网技术的发展历程

无线局域网是计算机网络与无线通信技术相结合的产物，利用无线多址信道这种有效方法来支持计算机之间的通信，并为通信的移动化、个性化和多媒体应用提供了可能。

在同一建筑物之内，无线局域网使得计算、协作无论在线或者移动状态下都能进行。只要在笔记本或手持式 PC 上安装 PC Card 适配器，就能够在办公室内自由移动而保持与网络的联结，具有传统局域网无法比拟的灵活性。因此，无线局域网非常适合于暂时性的工作小组或者快速发展的组织。

IEEE 802.11 是在 1997 年由大量的局域网以及计算机专家审定通过的标准，规定了无线局域网在 2.4GHz 波段进行操作。目前应用最广泛的是 1999 年修订的 IEEE802.11b 标准，IEEE 802.11b 采用 2.4GHz 频带和补偿编码键控 (CCK) 调制方式，从根本上改变 WLAN 设计和应用现状，扩大了 WLAN 的应用领域。802.11b 使用与以太网类似的连接协议和数据包确认，以便提供可靠的数据传送和网络带宽的有效使用。现在，大多数厂商生产的 WLAN 产品都基于 802.11b 标准。

2001 年 11 月 15 日，IEEE 试验性地批准一种新技术 802.11g。该技术使无线网络传输速率可达 54Mbps，并且和 802.11b 兼容。这意味着用户将可能以 54 Mbps 的高速通过无线方式接入互联网，并使用今后推出的多媒体和即时视频服务。无线局域网的新标准还考虑了 QoS。可以预见，随着 802.11g 标准的推出，无线局域网的时代即将来临。

移动 Ad hoc 网络是一种完全由无线连接的移动节点构建的网络，具有网络拓扑变化及有限频宽等特性，网络成员则可能由笔记本电脑，PDA 和移动电话等构成。Ad hoc 网络研究中一个非常重要的领域就是路由问题。因为每个节点都兼有路由功能，就需要适合这种特殊环境的路由协议和路由算法。传送不同类型业务也是 Ad hoc 网络需要解决的问题之一，在某些实时应用的场合（如传送语音、视频图像等信息），不但需要正确的路由信息，还需要提供有一定保障的 QoS (Quality of Service)。这些要求对带宽较低、通信信道质量都相对受限的 Ad hoc 网络而言，将是很大的挑战。所以，如何在 Ad hoc 网络条件下为业务

传输选择最佳路由，对于合理利用无线网络资源，实现有针对性的 QoS 保证具有重要意义。

1.2 无线局域网路由技术的研究和应用现状

无线局域网路由技术是无线网络研究的一项重要内容。国内外众多研究机构已经开始关注 Ad hoc 网络中路由技术的实验和研究。美国 UCLA 等多所大学开发了几类适用于无线网络的路由协议，随着组播应用^[1-4]的发展，无线网络和组播应用的结合将成为无线局域网技术的一个发展方向。

早期静态网络环境中的组播路由协议有 DVMRP(距离矢量组播路由协议)^[5], MOSPF(开放式最短路径优先组播路由协议)^[6], CBT(基于核心树的组播路由协议)^[7]以及 PIM(协议独立的组播路由协议)^[8]等等。但由于 Ad hoc 网络的特殊性，这些组播协议在该环境下的路由性能都不十分理想，因此又出现了一系列应用于 Ad hoc 网络中的路由协议，这些新的组播路由协议是从早期的单播协议研究^[8-11]中发展起来的，现在较为常见的包括 AMRoute^[12], AMRIS^[13], CAMP^[14]等。

目前在 Ad hoc 领域所进行的主要开发工作有 UCLA (University of California, Los Angeles) Mario Gerla 领导的 NRL (Network Research Laboratory) 工作组，该工作组在对 Ad hoc 网络的单播，组播路由协议性能等进行了仿真与测量^[15,16]方面作了大量工作，深入研究了 Ad hoc 网络的 QoS 支持，并提出了用于 Ad hoc 网络的按需组播路由协议 ODMRP^[21,29]。此外，还有众多知名大学如 University College London^[17]等也都在进行 Ad hoc 网络的开发研究工作。

尽管国内外在无线 Ad hoc 组播网络路由协议方面的研究非常活跃，这些研究活动都将使人们对无线网络及其组播业务的认识得以不断深入。然而，由于无线网络结构的特殊性，已有的路由协议都存在着一定的不足之处，不能完全提高数据传输效率，因此，优化无线路由协议、提高 Ad hoc 网络 QoS 保证仍然是现今国内外一项很有研究潜力的课题。

1.3 课题背景

本课题所在的研究小组是国内最早开展无线局域网研究的单位之

一，课题组获得国家自然科学基金的支持，并在 CISCO 与 IBM 共同资助下顺利展开。课题组由多个小组组成，分别研究如何在无线局域网上实现并提高单播路由协议、组播路由协议性能^[18]，如何实现无线局域网的负载均衡，以及设计实验床测量进行协议性能的测量和实际应用的模拟^[19]等。

本课题组承担了天津市信息港工程项目—“无线 IP 网络控制技术”。任务是：根据已有的无线网络组播业务的研究成果，开发无线组播路由协议算法，研究网络流量控制和拥塞控制技术。并结合无线通信的特点，研究无线局域网路由优化技术。同时将无线路由技术从单播环境扩展到组播环境下，并使用概率论方法进行仿真实验的分析。

在该项目中，通过分析无线网络业务流的特点，建立其业务模型，结合信道分配算法，并考虑入网节点的移动特性以及无线网的拓扑结构变化，研究了新的网络路由选择算法，并对路由算法进行性能评价。

1.4 论文结构

论文的第二章介绍与本论文相关的背景知识，对 Ad hoc 无线移动网络进行综述，包括 Ad hoc 网络及其路由协议以及组播路由的相关概念。第三章提出了改进 ODMRP 的思想，并给出了具体实现的算法。第四章介绍了仿真实验所用的工具软件，并说明实现 EODMRP 所使用的主要数据结构和方法，最后讨论文中所用的基于组播的网络延迟测量问题。第五章简单讨论了 Ad hoc 无线网络中 QoS 的保证策略。第六章给出研究结论：EODMRP 路由协议具有相当的优势，并提出 EODMRP 今后的研究方向。

1.5 作者的工作

作者分析了多种 Ad hoc 网络协议的特点，并使用 linux 下的 gcc 环境实现了基于 Ad hoc 网络 ODMRP 协议的改进。改进后的 EODMRP 通过主动探测链路的方式感知路径的状态，根据不同数据包大小采用不同的路由查找方式，一方面不增加网络的总体负载，一方面不增大数据包的传输延迟，这样就可达到充分利用 Ad hoc 网络资源，提高节点通讯质量的目的。

鉴于实验条件，作者将算法移植到 NS 的仿真环境中进行性能测

试。利用软件仿真，对 ODMRP 和改进后的 EODMRP 的路由性能进行了比较。对仿真输出文件的分析表明，改进后的 EODMRP 路由协议比原有的 ODMRP 协议在投递率、丢包率和延时等方面都有了不同程度的改进，这表明 EODMRP 在 Ad hoc 的网络环境中能够有效提高网络吞吐率。

第二章 Ad hoc 网络及网络路由综述

2.1 Ad hoc 网络

2.1.1 Ad hoc 网络定义

Ad hoc 网络是若干无线移动节点的集合，这些节点可以动态地任意移动，之间的链接可以随时间连续变化，网络不依赖于建设好的基础设施（established infrastructure），不存在集中式的网络管理。

自二十世纪七十年代出现以来，无线网络越来越受到计算机产业的重视。尤其是最近十年间，无线网络与移动相结合，令其大受欢迎。

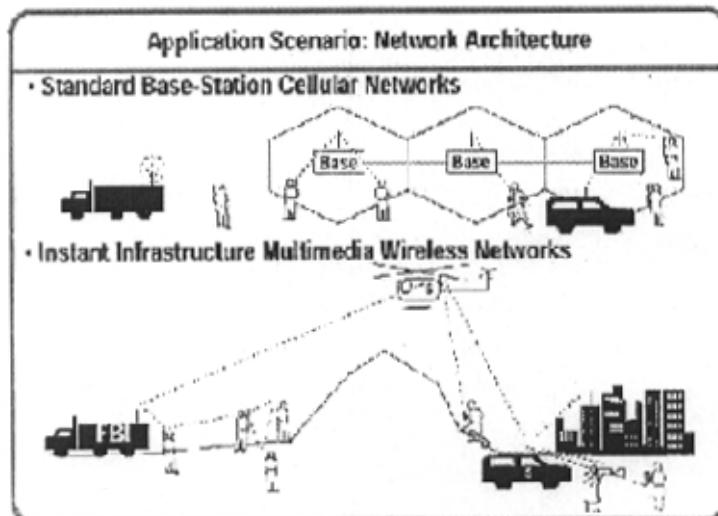


图 2-1：蜂窝无线网络和 Ad hoc 网络的比较

无线移动网络可以分为两类。第一类是有基础设施的（infrastructured）网络，即依赖于固定有线网关的网络。这种网络中有线与无线部分的桥接也称为“基站”。网络中的移动单元直接与其通讯半径之内的最近基站连接与通讯。当该移动单元离开了某一基站的通讯范围而进入另一基站的通讯范围时，原基站与新基站之间进行转接（handoff），从而移动单元可以不受妨碍地继续通讯。第二类无线移动网是无基础设施的网络，通常称为 Ad hoc 网络。这类网络中不存在

固定的路由器；所有的节点都可移动，并且可以动态地自由连接。每个节点都具备路由功能，负责发现并维护通往其它节点的路径。图 2-1 分别显示了带基站的传统无线蜂窝结构和不带基站的 ad hoc 网络结构。

2.1.2 Ad hoc 网络的适用环境

Ad hoc 网络的应用实例包括：紧急搜索救援行动；人们希望快速交流信息的会议（meetings or conventions in which persons wish to quickly share information）；不适合居住地区的数据获取（data acquisition operations in inhospitable terrains）。

2.2 网络路由协议

网络路由是寻找将指定报文送达其目的地网络的路径的过程，通过路由协议在网络的所有路由器中建立相互一致的路由表实现。有线网络下的路由协议有多种分类方法^[34]。主要有如下四种方式：

集中式与分布式路由协议：集中式的路由中，有专门的处理器负责收集和处理网络链路的状态信息，为每个节点计算和分发路由表。分布式路由中，各路由器按照分布式路由协议进行合作，建立起相互一致的路由表。当整个网络是集中控制且网络规模不太大时，采用集中式路由较为合适。集中式路由的缺点是单点故障可导致全网通信瘫痪以及路由通信量集中于一点。

源路由与逐跳路由：源路由（source route）允许源节点精确地指定信息包发送的路径，但同时也要求源节点必须了解整个网络的拓扑结构。以增加的包头大小换得路径的可指定性。逐跳路由则仅携带目的节点地址，由经过的每个路由器决定下一跳的地址。

确定性路由与随机路由：确定路由中，路由器使用同一条路径发送目的地为同一节点的所有信息包。随机路由中，路由器为同一目的节点维护多条的路径，发送数据包时，路由器从维护的若干条路由中任意选择一条使用。随机路由的优点是将负载分摊在多条路径上，消除了确定路由的震荡性。但另一方面，应用随机路由的方法，目的节点在同一个连接上收到的数据包可能会乱序，且各个数据包的传输延迟时

间长度不等。所以，现代的网络通常使用确定路由方法。

状态依赖(state-dependent)路由与**非状态依赖(state-independent)**路由：状态依赖的路由（也称为动态路由，dynamic routing）对于路径的选择依赖于当时（所测量到）的网络状态。非状态依赖的路由（也称为静态路由，static routing）中，选择路径时忽略网络的状态。状态依赖的路由方法所找发现路径通常比非状态依赖的路由方法所发现的路径更优，但是前者会受到网络状态波动所引发的问题的困扰（如路由震荡），而且为了监视网络状态其所要求的开销也更大。Internet 中既使前者也使用后者。

2.3 Ad hoc 网络中的路由协议

2.3.1 概述

假如 Ad hoc 网络中仅仅包含相距很近的两台移动主机，则不需要路由协议。

而实际情况是，在许多 Ad hoc 网络中，两台需要通讯的主机可能互相不在对方的无线传播半径之内。如果位于这两台主机之间的其它移动主机能够为它们转发信息，它们就可以进行通讯了。例如在图 2-2 所示的简单 Ad hoc 网络中，主机 C 位于主机 A 的无线传播半径之外（用环绕着 A 的圆圈表示），由于主机 B 位于 A 与 C 的传播半径的重叠区域，因而 A 和 C 可以利用 B 为它们转发数据包。实际的 Ad hoc 网络的路由协议不像这里一样简单，因为无线传播在各个方向上传播特性不均匀，而且任何主机在任何时候都可能移动。

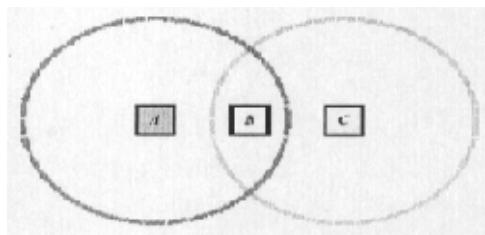


图 2-2：仅由三个移动节点构成的简单 Ad hoc 网络

Ad hoc 网络路由协议的基本目标是正确、高效地在节点对之间建立路径连接，及时传输数据。路径的建立应该尽可能降低开销、少占

用网络带宽。自从二十世纪七十年代早期 DARPA 包交换无线网络出现以来，人们已经开发了大量针对 Ad hoc 无线网络的协议。这些协议必须处理好 Ad hoc 网络的典型局限，包括能量耗损 (power consumption)、低带宽、高误码率 (high error rates)。如图 2-3 所示，总体上，这些路由协议可以分为两类：(a) 表驱动 (table-driven) 协议 (b) 源发起的随选驱动 (source-initiated on-demand driven) 协议。图中的实线表示直接后裔，虚线表示逻辑后裔。尽管都是为同种网络而设计，这些协议的特征却截然不同。

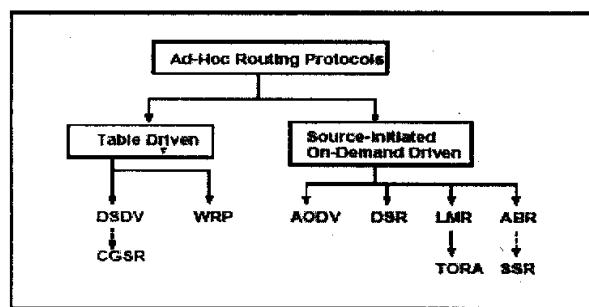


图 2-3: Ad hoc 路由协议的分类

2.3.1.1 表驱动型路由协议

Table-driven 的路由协议试图维护网络中从各个节点到所有其余节点的最新路由信息，所有路由信息保持一致。这类协议要求每个节点维护一或多个路由信息表，当网络拓扑变化时，更新信息被传遍整个网络，以维护一致的网络视图。这类协议之间的不同之处在于：必需的路由相关表的数量不同，网络结构变化信息的传播方法不同。主要代表有：DSDV(Destination-Sequenced Distance-Vector Routing)、CGSR(Clusterhead Gateway Switch Routing)和 WRP(The Wireless Routing Protocol)。

2.3.1.2 按需路由协议

与表驱动路由协议不同，源发起的随选驱动协议只有当源节点需要时才寻找路径。当源节点需要一条通往目的节点的路径时，它在网络中发起一次路径发现过程 (a route discovery process)。当找到一条路径后或者所有可能的路径都被检查过后，路径发现过程结束。路径

一经建立，它就被某个路由维护程序所维护，直至目的节点无法经由任何路径到达或者该路径不再被需要。源发起的随选驱动协议的代表协议有：AODV（Ad hoc On-Demand Distance Vector Routing）、DSR（Dynamic Source Routing）、TORA（Temporally-Ordered Routing Algorithm）、ARB（Associativity-Based Routing）和 SSR（Single Stability Routing）。

2.3.2 Ad hoc 网络路由协议比较

S.J.Lee,W.Su, 和 M.Gerla 等人通过大量的仿真实验对四种 Ad hoc 网络路由协议的性能进行了比较和分析^[20]。这四种协议是：DSR, ABR, DBF, ODMRP，其中既包括 table-driven 的需要周期地进行路由广播的协议，也有 on-demand 的依靠路由发现过程的协议。其仿真的 Ad hoc 环境是 30 个移动主机，在不同的移动频率和移动速度下对上述四种协议进行比较。结论是：每一种协议都可以在某些情况下工作得很好，而在另一些情况下有各自的缺陷。“ABR 的性能很规律：当节点的移动频率低且速度慢时，ABR 几乎成功传送了所有的数据包；而当节点的移动频率和速度增大时，ABR 协议汇聚很慢。ABR 从路由开销方面来讲是实验中各种协议中最差的，但是当数据源为 10 到 20 个（即网络处于轻载状态）时，ABR 成功传送了 90% 以上的数据包。当数据源增加到 30 个（即网络处于重载状态）后，网络无法应付路由协议所带来的通讯量开销，有很大比例的数据包被丢弃（drop）。尽管 ODMRP 因使用按需路由而增加了路由协议开销的字节数，但它仍然在实验中所尝试的各种移动频率和移动速度场景中具有非常好的性能。DSR 在各种移动频率和移动速度场景中的性能与 DBF 几乎一样好，同时又实现了降低路由字节开销的目标，但是它要求发送大量的路由信息包，所以当节点移动频繁快速时 DSR 实际上比 ABR 招致了更多的路由开销。”^[20]

总体来说，ODMRP 具有相对优秀的性能。因而本文对 Ad hoc 路由协议的研究选择 ODMRP 协议作为切入点。

2.4 Ad hoc 网络中的组播路由协议

2.4.1 组播

随着 Internet 的迅速发展与普及，许多应用例如电话会议都需要

组播路由协议的支持，对组播路由协议的需求正变得越来越广泛。

组播是一种允许一个或多个发送者（组播源）发送单一的数据包到多个接收者（一次的，同时的）的网络技术。组播源把数据包发送到特定组播组，而只有属于该组的地址才能接收到数据包。组的成员动态形成，可以在任何时间加入一个组或离开一个组。组的大小和位置没有限制。一个主机可以是多个组的成员。组可以是永久的，也可以是临时的，永久组有一个公开的分配好的 IP 地址，永久组内的成员数也可以为 0。有一些组播地址保留用于临时组，只有存在组内成员时组才存在。网络上传输组播数据报时是通过组播路由器进行的，组播路由器可以和网关在一起，也可以和网关分离。在单播情况下，对每个目标地址都要发送一个数据包拷贝，数据包对网络带宽的占用与目的主机增加成正比。组播可以大大的节省网络带宽，图 2-4 (a) (b) 是有线方式下单播与组播通信的示意图。

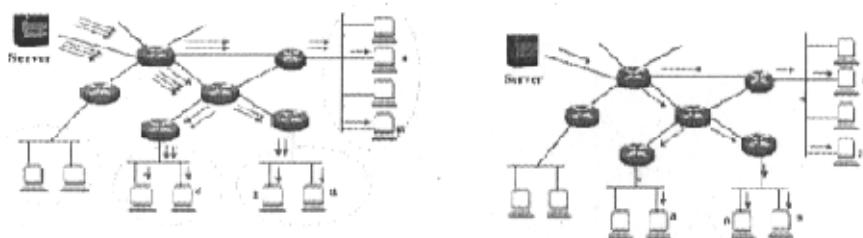


图 2-4(a): 单播通信示意图

图 2-4(b): 组播通信示意图

典型的无线 Ad hoc 网络中，无线主机通常以组的方式共同执行某项任务。组播在 Ad hoc 网络中起着很重要的作用。用于静态网络中的组播路由协议有 DVMRP(Distance Vector Multicast Routing Protocol), 最短路径优先组播路由协议 MOSPF (Multicast Open Shortest Path First) , 基于核心树的组播路由协议 CBT (Core Based Trees) , 协议独立的组播路由协议 PIM (Protocol Independent Multicast)。但是在无线移动 Ad hoc 网络中，由于无线节点的移动，组播的树形结构经常发生变化，需要不断根据连接的改变进行调整。此外，跟距离向量和链路状态路由协议相同，组播树的建立需要知道全局的路由结构。在 Ad hoc 网络中，频繁的拓扑变化使网络的距离向量或链路状态路由表不断发生变化，会给网络信道和处理带来过大的负载。因此应用于有线静态网络中的组播路由协议并不一定适合无线 Ad hoc 网络。要在无线移动

Ad hoc 网络中实现组播，就需要对有线组播路由协议的路由发现方式，组的保持方式进行改进，使其适用于无线移动 Ad hoc 网络。

根据 Ad hoc 网络的具体特性，即拓扑不断发生改变，而网络带宽又极其有限的特点，提出了对 Ad hoc 网络使用按需路由协议。所谓的按需路由协议就是指该路由协议并不是一直保持到每一目的的路由，而是根据源的需要建立相应的路由。只有当源需要建立到达某目的的路由时，才发送路由请求包建立源到目的的路由。收到路由请求包后，目的根据相应的路由算法选择最佳路由。目的对源的应答包从这条最佳路由返回给源。不使用定期的路由更新，这种按需建立路由的路由算法，大大减少了网络中控制包的负载。

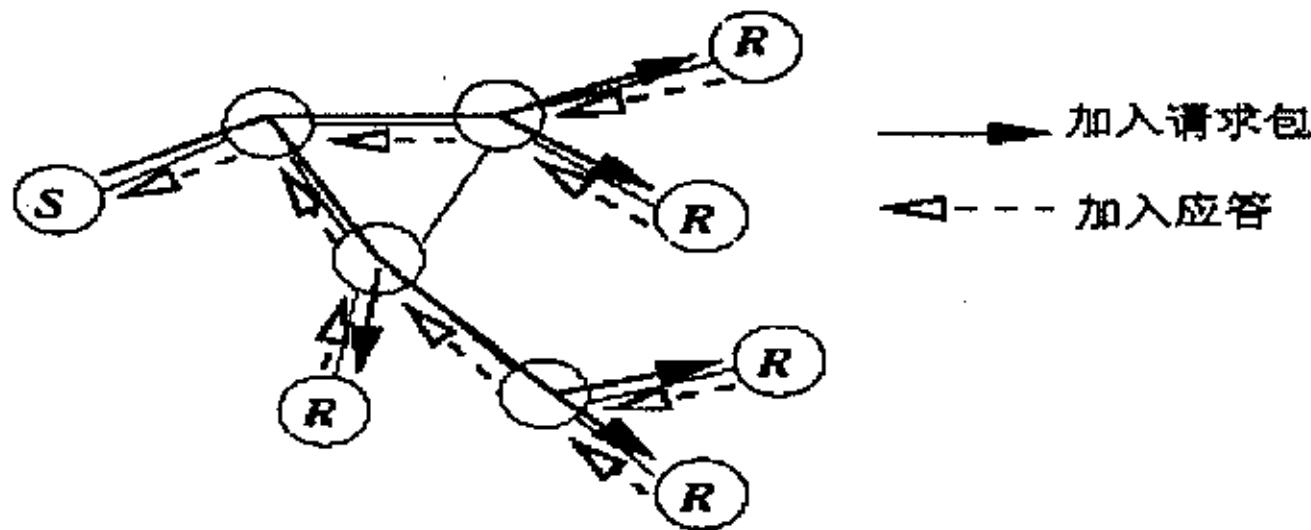
目前已经提出了许多 Ad hoc 网络中的按需路由协议，包括轻量移动路由协议 LMR (Lightweight Mobile Routing), 动态源路由协议 DSR (Dynamic Source Routing), 临时路由算法 TORA (Temporarily Ordered Routing Algorithm), Ad hoc 按需距离向量路由 (Ad hoc On Demand Distance Vector), 多路径动态源路由 MDSR (Multipath Dynamic Source Routing) 等。

2.4.2 ODMRP 的路由机制

按需组播路由协议 ODMRP(On-Demand Multicast Routing Protocol)正是应用按需路由技术的一种组播路由协议，这种技术有效的减轻了网络信道中协议控制包的负载，提高了信道利用率。按需组播路由协议非常适合应用于无线 Ad hoc 多跳网络，ODMRP 的开发与改进对于 Ad hoc 网络的大规模应用具有重要的意义。

ODMRP 使用了一个转发组的概念，即由一系列节点负责在组成员之间转发组播数据，这样每个组播组都形成了一个网状 (mesh) 的转发结构，从而用网状结构替代了传统上组播的树形结构。组播通过使用和保持网状转发结构，克服了在无线移动网络中组播树形结构中节点间间断性的连接，通信过于集中，频繁的树形结构的重新配置，共享树中非最短路径等缺点。ODMRP 在成员离开组时，不发请求信息。ODMRP 的这些机制明显的减轻了信道负载，使其更适用于大型网络，使移动网络更加稳定。

下面是 ODMRP 建立转发组，对网状转发结构的维护过程。ODMRP 按照信源的要求建立并更新组成员和多跳路由，其中包括了一个请求阶段和一个应答阶段。图 2-5 为 ODMRP 按需建立的请求与应答过程。



2-5: ODMRP 组成员建立与保持示意图

当组播信源要发送数据包时，会广播一个带有要发送数据的广播包，这个包称为加入请求，该包周期性的在全网广播以刷新成员信息和更新路由，当某节点收到一个非重复的加入请求时，会将其上游节点的信息（主要是地址和 ID）存入自己的路由表中并再将其广播出去，当加入请求包到达组播信宿时，信宿产生并广播一个加入应答包给其邻居。节点收到加入应答后，先检查某一条路由中下一节点信息是否和自己路由表相匹配。如果匹配，意味着节点位于通往信源的路径上，是转发组的一部分，该节点就会被打上标记以标志成为转发组的成员，同时，节点广播该加入应答。这样的加入应答包全部以最短路径返回信源，包所经过的节点组成一个“转发组”。

图 2-6 为转发组的示意图。如图所示，转发组实际上就是转发组播数据包的一系列节点，通过这些节点的支持组播组成员以最短路径发送数据包。不论是组播组成员还是转发组成员都会转发组播数据包。如果组播的一个信宿恰好在一个组播的信源和另一个信宿之间，那么这个节点也可以作为转发组节点。很明显，这种网状结构比树形结构

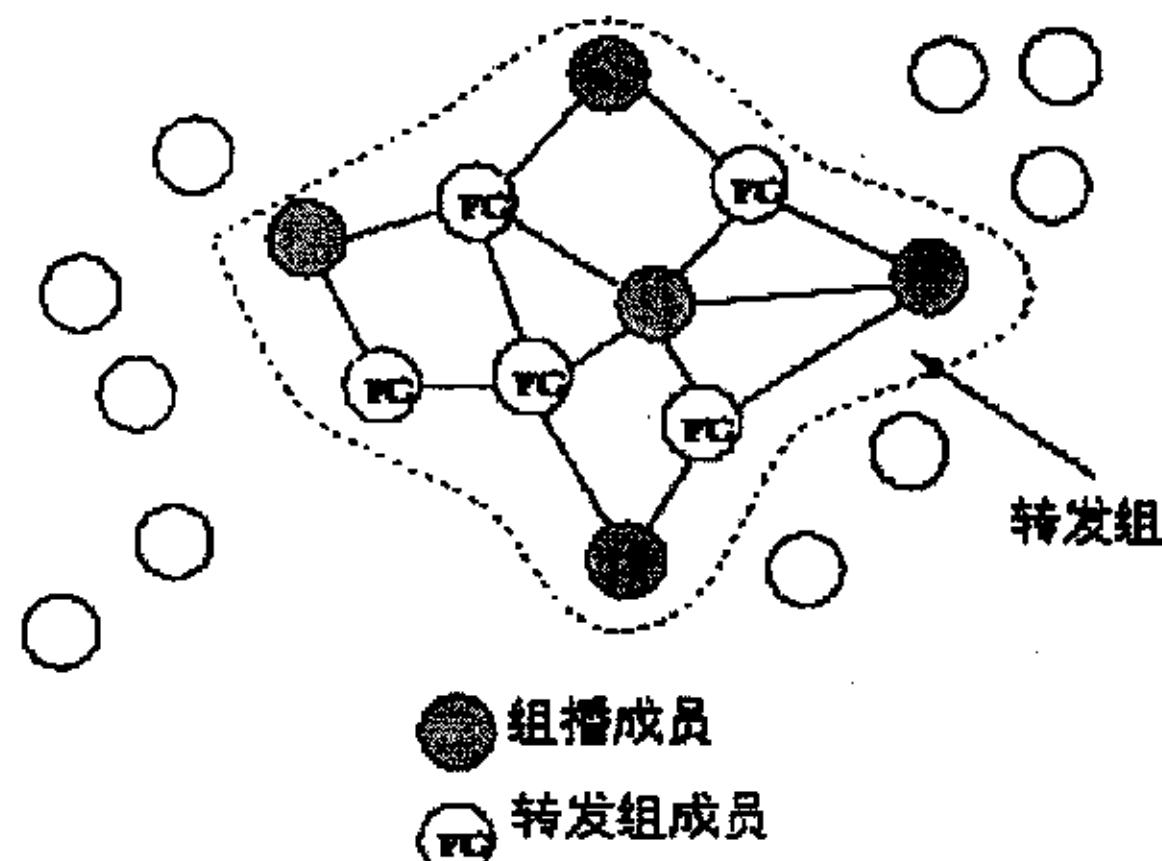


图 2-6：转发组示意图

保持了较多的冗余连接，可以帮助克服节点的变换，信道的衰减等，也避免了组播树中频繁的重新配置。

经过转发组确认和路由建立的阶段，信源以通过所选的路径和转发组向组播信宿发送数据包。当节点收到组播数据包时，只有在该包为非重复出现且节点是组播会话的转发组成员的情况下才会转发。这个过程在保证为组播成员提供备份路由和多条连接的情况下，减少了业务负载并防止数据包通过失效路由发送。

在 ODMRP 中，没有专门的控制包用来加入或离开组。当组播的信源离开某个组时，仅停止发送加入请求数据包即可。同样，当某个信宿不再从某个组播组接收数据时，仅不再发送加入应答。在定时器过时后，该节点自动不再成为组成员。

2.4.3 ODMRP 在 Ad hoc 网络中的优点

相对于传统路由协议，如有线网中大量使用的距离矢量协议和链路状态协议，ODMRP 在 Ad hoc 网络环境中具有许多优势。

首先，ODMRP 不使用周期性的路由广播。这降低了对网络带宽的额外开销，尤其是当网络中的主机没有显著移动时。同时，移动主机上电池电量的损耗减少，因为主机既不需要发送也不需要接收路由广播。另外，两个有线网络通常固定地配置为只通过一个或少量几个路由器连接，而在 Ad hoc 网络环境中不可能存在这样的固定配置，所有的通讯都只能通过广播方式进行，所以周期性的路由广播中的冗余路径信息在 Ad hoc 网络中带来更大不必要的负担，包括带宽消耗方面和 CPU 占用方面。

其次，基于距离矢量协议和链路状态协议的传统路由协议可能会计算出一些不能在 Ad hoc 网络中工作的路径。在无线环境中，一对主机可能在相对的两个方向上传输效果不一样，因为在两个主机周围可能存在不同的传播模式(propagation pattern)或干扰模式(interference pattern)。例如距离矢量协议，尽管主机 A 能够收到主机 B 发出的路由通告 (routing advertisement)，但是从 A 向 B 传送的信息包却有可能无法到达 B。ODMRP 不要求主机之间的传输必须是双向的，尽管当链路层协议(如 MACA、MACAW)做出如此的保证时 ODMRP 也会使用双向传输。

最后，传统路由协议不是针对 Ad hoc 网络中存在的动态拓扑变化

设计的。在传统有线网络中，路由器之间的链路会偶尓断开（go down）或恢复（come up），链路代价（cost）也会因拥塞而变化，但是路由器毕竟不会到处移动。在移动主机作为路由器的环境中，传统路由协议，尤其是距离矢量协议，在网络拓扑结构动态变化之后向新的稳定路径汇聚的过程可能会很缓慢。ODMRP 协议能够迅速适应主机移动后带来的拓扑变化，而当没有主机移动时不增加额外的路由开销。

总之，ODMRP 通过构造全网状（mesh）结构提供冗余路径，从而增强了协议对节点移动的可靠性。当网络高度移动时，协议不产生额外开销，因为没有因链路断开而引发任何控制包。然而，当存在大量组播发送节点时，协议可能会承受过多的控制开销。对大组播群的可扩展性必须增强。

第三章 增强型 ODMRP 协议

3.1 问题的提出

按需 Ad hoc 组播路由方式下，协议的工作不是周期性进行，而是由数据包驱动。Ad hoc 网络中，对于单播传输，这种按需的路由方式在路由协议负载和对网络路径改变的快速反应方面显示了很大的优点。研究表明这种优点也将体现在 Ad hoc 网络的组播路由中。

多数组播路由协议利用了周期机制，这种周期可以看作协议中非按需的部分，而协议按需特性来自这样的一个事实：即协议的多数重要操作的对象是当前活动的组播组。但是，协议中的周期机制要负责核心路由功能，这样就会限制协议其他的按需功能带来的优点。ODMRP 协议通过周期性发送全网组播控制包来建立组播 mesh。协议依赖于这种广播来修复广播期间组播 mesh 中的链路故障，MAODV 协议则要求通过进行周期性的邻居探测来发现链路故障和用周期性的发送 hello 信息来建立组播转发状态，不管组播组中是否有发送者，只要有至少一个接收者，组播组中都会周期性的产生 hello 信息。和 MAODV 一样，ABAM 协议也要求进行周期性的邻居探测以发现链路的故障。另外，这些协议依赖于显式的“修剪”信息来删除无用的路由状态。但是由于接口或发送者移出通信范围而引起的修剪信息的丢失会使得节点继续转发无接收者的数据包，这样就会加大网络不必要的负载。

另外，Ad hoc 无线移动网络中的带宽总是相当有限；并且，同 Internet 一样，在链路上以及在路由器上的拥塞是造成 Ad hoc 网络中的大延迟的主要原因。所以，在 Ad hoc 网络中，充分利用网络带宽以及避开拥塞链路和拥塞路由器显得更为重要。

3.2 改进 ODMRP 的可行性

在 Ad hoc 环境中，由于采用无线通信方式，所以当主机数量足够多（如 50 个）时，潜在的可用路由相对于有线网而言更多，如果能够充分利用网络带宽，那么就可以提高 ODMRP 的效率。其次，在移动环境中，路径的变化性强，如果能够在链路空闲期间及时获得信源与信宿之间可达性的信息，以及关于链路状态的反馈信息，则数据包尤

其是无连接的数据包到达目标主机的概率就可以更稳定，从而提高通讯质量。

按需的组播路由协议根据信源的需要建立、更新组成员和组播路由。在无线网络上，链路并不总是处于繁忙状态，同时链路上的节点由于随时变化，会使信源到信宿的路由路径发生变化，鉴于这样的情况，要提高路由协议的效率，必须充分利用有限的网络带宽。

在 ODMRP 中，信源携带数据进行路由探测，这样源主机确定地了解它有哪些可以到达目标主机的路径，从而自由地从中选择合适的路径来发送数据包。这就使在 ODMRP 上的组播路由相对于在逐跳的 (hop-by-hop) 路由协议上的组播路由更为简单，且免除了路径上各节点为保持对多径的一致认知而需要进行的额外通信和计算。已完成的 Ad hoc 网络的基础测量工作^[19]显示，ODMRP 本身具有较好的路由性能，结合组播应用的特点，本文选择 ODMRP 作为协议改进对象，提高其在 Ad hoc 网络中的路由性能。扩展后的路由协议称 EODMRP (Enhanced On-Demand multicast Routing Protocol，即增强型按需组播路由议)。

3.3 EODMRP

3.3.1 设计思想

EODMRP 支持传统的 IP 组播服务模型，既允许接收者接收任意发送者发送的组播数据包，还支持新的特定源模式的服务模型，即接收者只加入含有某一特定源的组播组。在这两类组播服务模型中，节点不必为了成为组播组的接收者而先向组播组发送数据，发送者在发送数据前也不必提前向组播组成员进行宣告。发送者在退出发送状态时，也不需要发送显式通告。

在 EODMRP 中，将 ODMRP 中的 mesh 结构改变为组播树结构。只要网络中有至少一个发送者和接收者，就建立对应于某一特定源的组播转发树。EODMRP 监控组播源的业务状态，并据此来发现组播树中的链路故障；如果组播源没有处于激活状态，则不会发送任何数据。前一种情况下，协议初始化本地修补进程，如果本地修补失败则转为全局修补。后一种情况下，组播转发状态不需要显式发送任何关闭信息而自行终止转发状态。当组播源暂停数据发送，为了启动对组播树的链路故障监控，EODMRP 以递增的时间间隔发送有限个 keep alive

信息，若组播源在一定时间内不再发送数据，则 keep alive 信息也被终止，整个组播树也随之终止。组播源发送模式的严重改变意味着组播源在一段时间内不工作，此时如果继续维持网络路由将成为一种浪费。EODMRP 还能够自动对无用的组播树的枝叶部分进行修剪，修剪与否不依靠显式的修剪信息，而根据是否收到来自下游节点的被动应答。

EODMRP 有时会以全网广播而不是现有的组播的方式发送一个现有的组播数据包，这个广播包只按非周期性间隔发送（如几个十分之一秒的时间发送一次），且只有当新数据发送给组播组时才产生。另外，这种广播并不是协议机制必需的，因为它并不具有核心路由功能。EODMRP 可以不借助 GPS 或其他定位信息及额外的业务控制来发现网络中节点是否因为移动速度过高而导致无法及时建立和维护组播状态。如果发现这样的情况，EODMRP 暂停当前工作转而以全网广播的方式发送所有数据包，一段时间后，当网络中的节点移动速度降低到一定程度，EODMRP 会再次尝试用组播路由进行更高效的工作。

网络中使用 EODMRP 协议的发送方和接收方共同建立和维护转发状态以进行组播通信。这里假设网络节点在任意时间可以任意移动，任何数据包都可能因为包碰撞或无线传输距离限制等原因丢失。EODMRP 监控组播转发状态，当一个或多个接收节点或转发节点与发送节点失去连接时及时进行修补。

只有组播转发树中的成员转发组播数据包，转发节点对每个包最多只转发一次。另外，数据包并不局限于沿着某一特定的父链路或子链路进行传输。例如，图 3-1 中接收者 R1 通过节点 B 收到包 X，通过节点 D 收到包 Y。当节点 D 先于 B 获得对介质的访问并先发送包 Y，或当节点 B 因为无线接口的因素没有正确接收到组播包，从而引起无法转发组播包时，就会发生图 3-1 的情况。

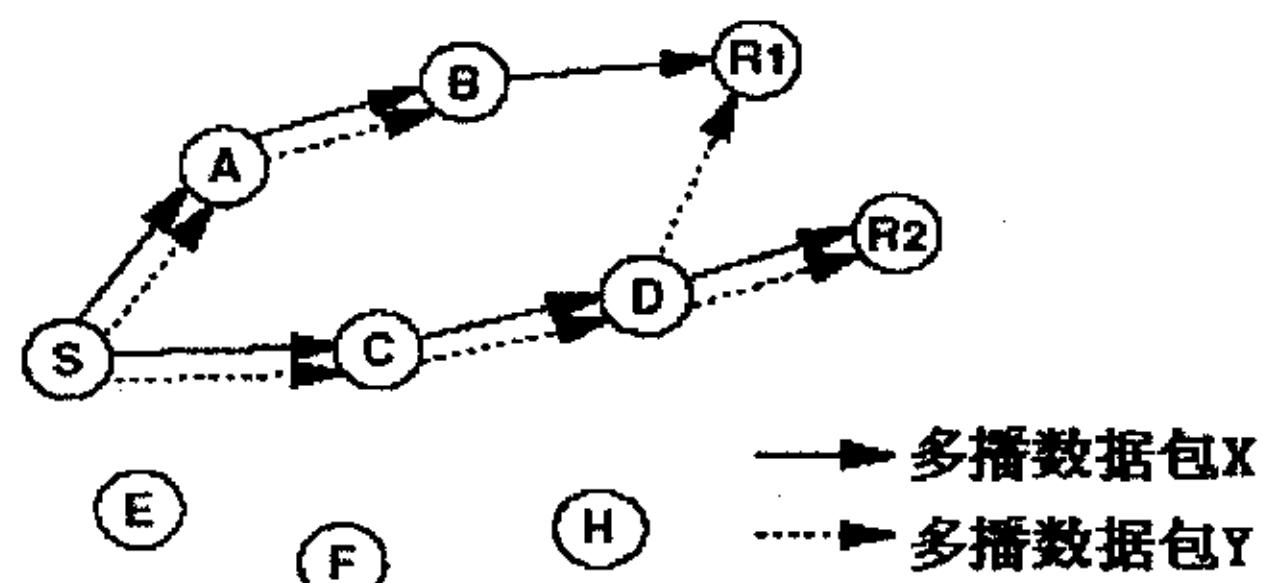


图 3-1：组播数据包转发

这里把限制在组播树内的广播称为树内广播，而将其他普通类型

的广播称为全网广播（见图 3-2），在组播树内的广播类似于 ODMRP 的“转发组”的概念，只是这里转发状态只适用于特定发送者，而不是被整个组共享。当发送者使用 EODMRP 发送组播数据包时，只需向组播树内的接收者发送广播，而在 ODMRP 中，数据包还要再反向广播到发送者。

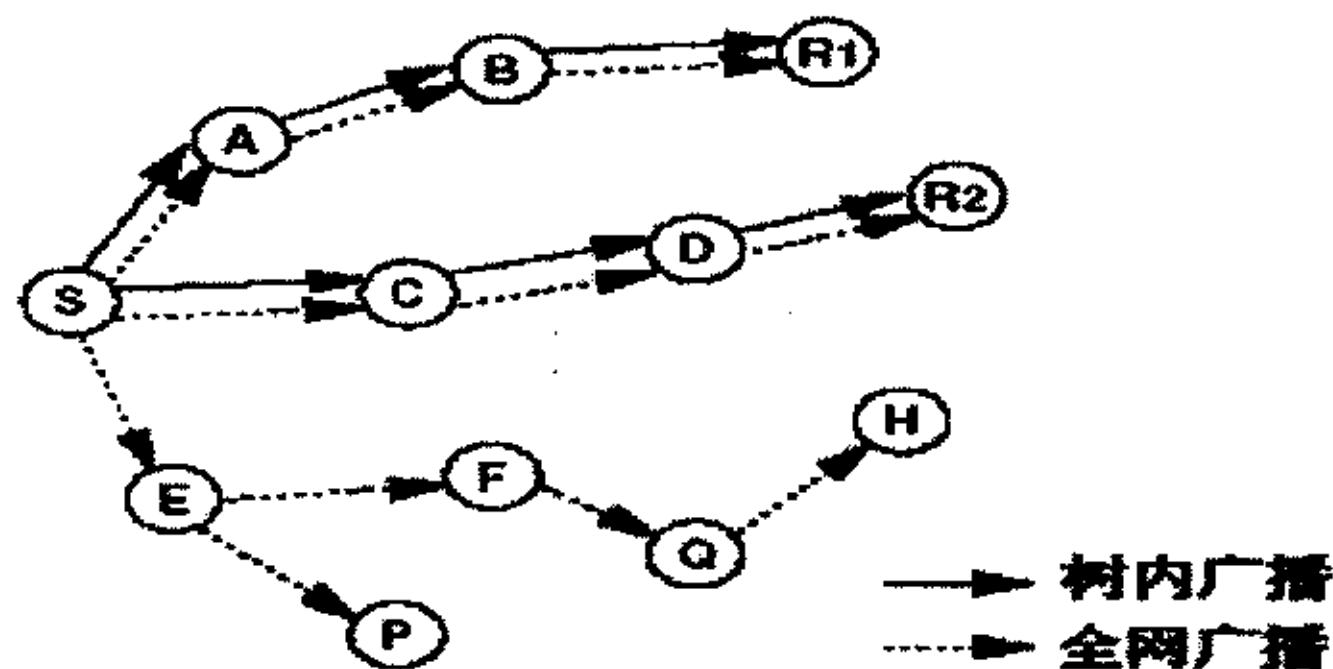


图 3-2：树内广播和全网广播

由组播组 G 的源 S 产生的每个组播数据包都包含一个较小的 EODMRP 包头，该包头包括了一组协议用来转发数据包和维护组播分布树的字段。EODMRP 包头中的序列号唯一定义了数据包，并可以作为源 S 产生数据包的计数器。S 将跳计数初始化为 0，并在每个节点转发数据包时增加计数器的值。对于被转发的数据包，EODMRP 包头中的前一跳地址是收到包的 MAC 层的传输源地址，该包在传输前将 MAC 层地址拷贝入该字段，包一产生就将该字段初始化为 0。

EODMRP 包头中还包括一个发包间隔时间。发送方 S 每经过一个间隔时间发送一个新的数据包到组 G。该时间字段由 S 根据动态跟踪获得的向组播组 G 发送数据包的平均间隔来决定。如果节点 S 的应用层在几个（如 1.5）当前发包时间间隔内没有产生新的组播数据，那么节点 S 的路由层就会向组播组 G 发送 keep alive 包。keep alive 包只广播到整个组播组而不是全网络，用于维护组播组现有的转发状态。发送 keep alive 包的间隔时间是由成功发送 keep alive 包的时间与某个因子（如 2）的乘积，且因子的值不断增加，直到达到最大间隔时间。经过一段时间的发送后，可以认为 S 已经不存在，同时 keep alive 包停止发送。和发送节点 S 相关的组播组中的节点都终止转发状态。

EODMRP 包头包括了用于增加 keep alive 包发送间隔的乘数因子，以及发送成功的 keep alive 包的数量，keep alive 包信息可以使节点获

得当前组播树的状态。

如果在一定间隔时间内，没有收到组播数据或 keep alive 包，这就意味着节点与转发树失去了连接。当处于组播组 G 且源节点为的 S 的转发节点在一定时间内没有收到数据包或 keep alive 包时，该组播节点就会进行本地修补，重新与组播树建立连接，如果本地修补失败，接收者将通过向全网发送广播进行全局修补，以重新建立连接。

EODMRP 不依赖 Ad hoc 网络中的单播路由协议，因此可以和任意单播路由协议一起工作，甚至可以在无单播协议的情况下独立工作。尽管在单播和组播协议之间进行信息共享会显得比较有用，但不这么做无疑会提高模块化和便携化。

3.3.2 根据移动预测改进路由刷新周期

ODMRP 通过周期性广播加入请求来建立和刷新路由。但过量的广播会占用 Ad hoc 网络中宝贵的带宽资源，引起网络拥塞和冲突。所以找到最佳的广播周期在 ODMRP 设计中非常重要。本文提出了一种能够使广播间隔与移动方式和速度相适应的方法，即使用 GPS（全球定位系统）提供的信息来预测路由有效的时间。根据预测的路由通断时间。加入请求只在数据发送过程中即将发生路由出错的情况下才发送。

在路由方法预测中使用了两个假设。第一，假设有足够的空间进行信号传输，接收到的信号强度只取决于和传输者之间的距离；第二，网络中所有的节点都通过网络时钟协议 NTP（Network Time Protocol）或 GPS 时钟进行了时钟同步。通过这两个假设，只要已知两个邻居节点之间的移动参数，如移动的速率，方向，信号的覆盖范围，就可以预测这两个节点之间可以继续保持连接的时间。

假设两个移动节点 i 和 j 在各自的传输范围 r 内， (x_i, y_i) 和 (x_j, y_j) 分别为 i 、 j 的坐标。 v_i 和 v_j 为 i 、 j 的移动速度，而 θ_i 和 θ_j ($0 \leq \theta_i, \theta_j < 2\pi$) 分别为 i 和 j 的移动角度。那么这两个节点之间可以继续保持连接的时间 D_i 可以通过公式(3-1)来预测。

$$D_i = \frac{-(ab + cd) + \sqrt{(a^2 + b^2)r^2 - (ad - bc)^2}}{a^2 + b^2} \quad (3-1)$$

其中

$$a = v_i \cos \theta_i - v_j \cos \theta_j$$

$$b = x_i - x_j$$

$$c = v_i \sin \theta_i - v_j \sin \theta_j$$

$$d = y_i - y_j$$

注意，当 $v_i = v_j$ 同时 $\theta_i = \theta_j$ 的时候， D_i 为 ∞ ，而不使用上面的公式。

为了利用以上信息，必须在请求包和应答包中再加入新的字段，当信源发送请求时，必须携带其位置，速率，移动方向信息。因为信源节点没有前一跳节点，因此在此节点处把 MIN-LET（最小链路存活时间）设为 MAX LET VALUE。信源的下一跳节点收到加入请求后，用上述公式估算本身和下一节点间链路的存活时间。然后选择所得的存活时间与 MIN-LET 间的最小值存入数据结构中。这样做的原理是，一旦某一路径上的一链路断开，那么整个路径都会失效。当组播成员收到加入请求时，计算本路由的上一段链路的 LET。

计算所得的上一链路的存活时间和请求包中的 MIN-LET 两者中的最小值就作为 RET（路由存活时间）。该 RET 值被写入应答包并广播给下一节点。如果转发组中的节点收到重复的应答包，但其 RET 值不同，就选择 RET 最小值写入自己的应答包中。信源收到应答后选择所有应答中最小的 RET，然后信源在最小 RET 内通过发送加入请求，建立新的路由。因为信源只在需要时才发送加入请求，而接收节点也只是在收到加入请求后才发送应答，因此应答的发送不需要组播接收节点周期性的发送。

在选择加入请求的最佳广播时间时，除了要估算 RET 值，还应考虑其他的因素。如果节点移动速率较高，网络拓扑改变较频繁，那么路由会经常很快失效。那么信源就会产生大量的加入请求，大量的广播会引起网络冲突和拥塞。所以此时 MIN_REFRESH_INTERVAL 的值要能够强制避免控制信息的过量传输。另一方面，如果节点静止或移动缓慢，或者链路长时间保持不变，那么路由极少失效，信源也极少发送加入请求。在这样的情况下产生的问题有：第一，如果路由中的节点突然改变其移动方向或速率，那么预测的 RET 值失效，路由不能及时重建；第二，当一远端非成员节点加入该组时，只有收到加入请求后才能发送新成员信息或者进行数据的接收。此时，应设置 MAX_REFRESH_INTERVAL 值。MIN_REFRESH_INTERVAL 值和 MAX_REFRESH_INTERVAL 值的选择应和网络环境相适应，如业务类型，业务负载，移动模式，移动速率，信道承载能力等。改进后的包格式如下（黑体部分为用于移动预测的字段）：

0	1	2	3	4
Type	Reserved	Time to Live	Hop Count	
Multicast Group IP Address				
Sequence Number				
Source IP Address				
Previous Hop IP Address				
Previous Hop X Coordinate				
Previous Hop Y Coordinate				
Previous Hop Moving Speed	Previous Hop Moving Direction			
Minimum Link Expiration Time				

但由于此种方式实现的复杂性，这里仅仅作为建议提出，在仿真实验中并没有利用到这个算法。

3.3.3 提高路由可靠性

加入应答的可靠传输在建立刷新组播路由表和转发组方面起了重要作用。如果应答不能正确传输，就无法实现 EODMRP 在组播路由中的高效性。用于无线网络的 IEEE802.11 MAC 协议在没有收到应答的情况下通过对包的重传保证了包的可靠传输。但是如果包是以广播形式发送的，那么就不存在应答和重传。EODMRP 中，因为要处理的是组播源，所以加入请求通常是以广播的形式发送给一个以上的节点。这种情况下，MAC 层无法保证请求包的逐跳可靠传输，也无法进行重传。

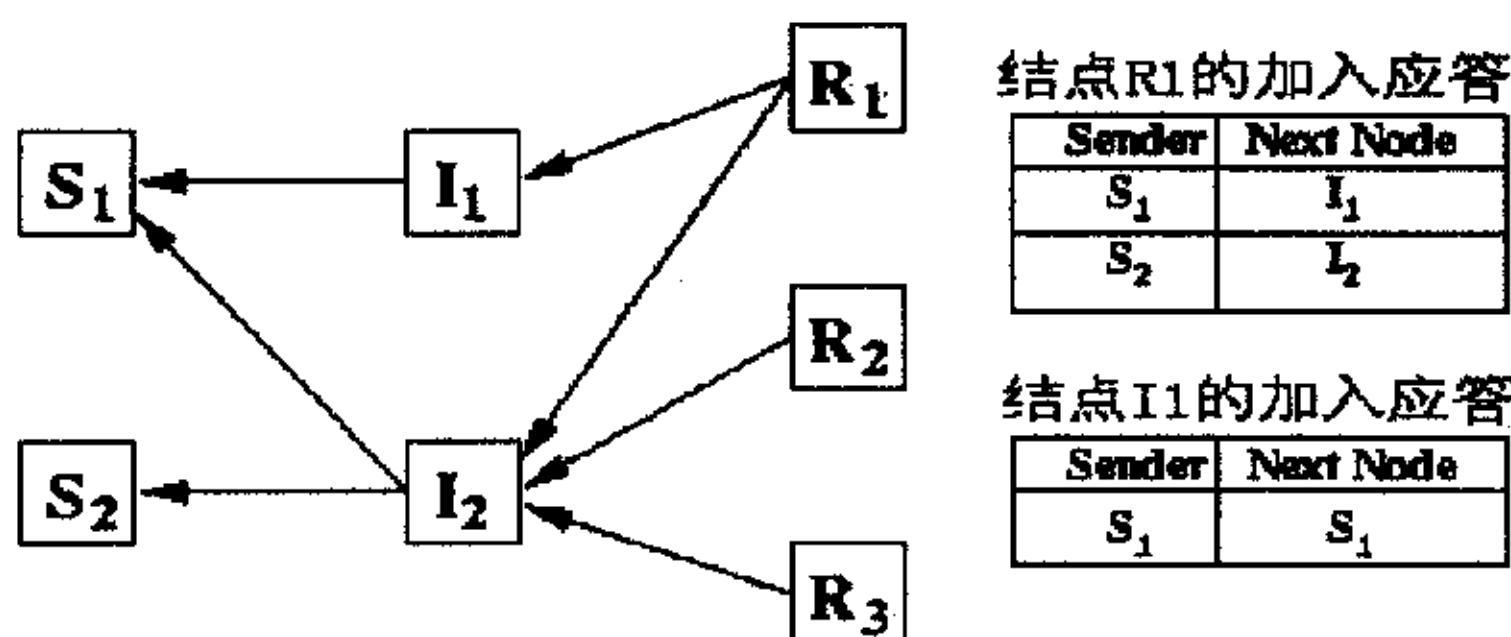


图 3-3：加入表转发图

为了保证可靠传输，可以将应答再划分成几个子表，每个子表对应一个下一跳节点。图 3-3 显示了节点 R1 处的加入应答被分割为两个子应答，一个对应邻居 I1，另一个对应邻居 I2。分割后的应答使用可靠的 MAC 协议-IEEE802.11，并以单播形式单独发送。因为邻居的个

数一般是有限的（一般在多跳的网络中，6个最佳），所以对于多信源的场景，该算法有比较好的性能。这种算法还可以用于下文将要叙述的被动应答的备份算法。

图 3-4 所示为一新的算法，当节点 B 收到节点 A 的包后转发给节点 C。如果节点 A 在 B 的传输范围内，那么节点 A 就能够侦听到 B 的传输，这样节点 B 到节点 C 的包传输就可以用于到节点 A 的被动应答。这个被动应答可以用来对加入请求进行确认。因为信源没有发送应答的下一跳，因此必须对前一跳发送主动应答。

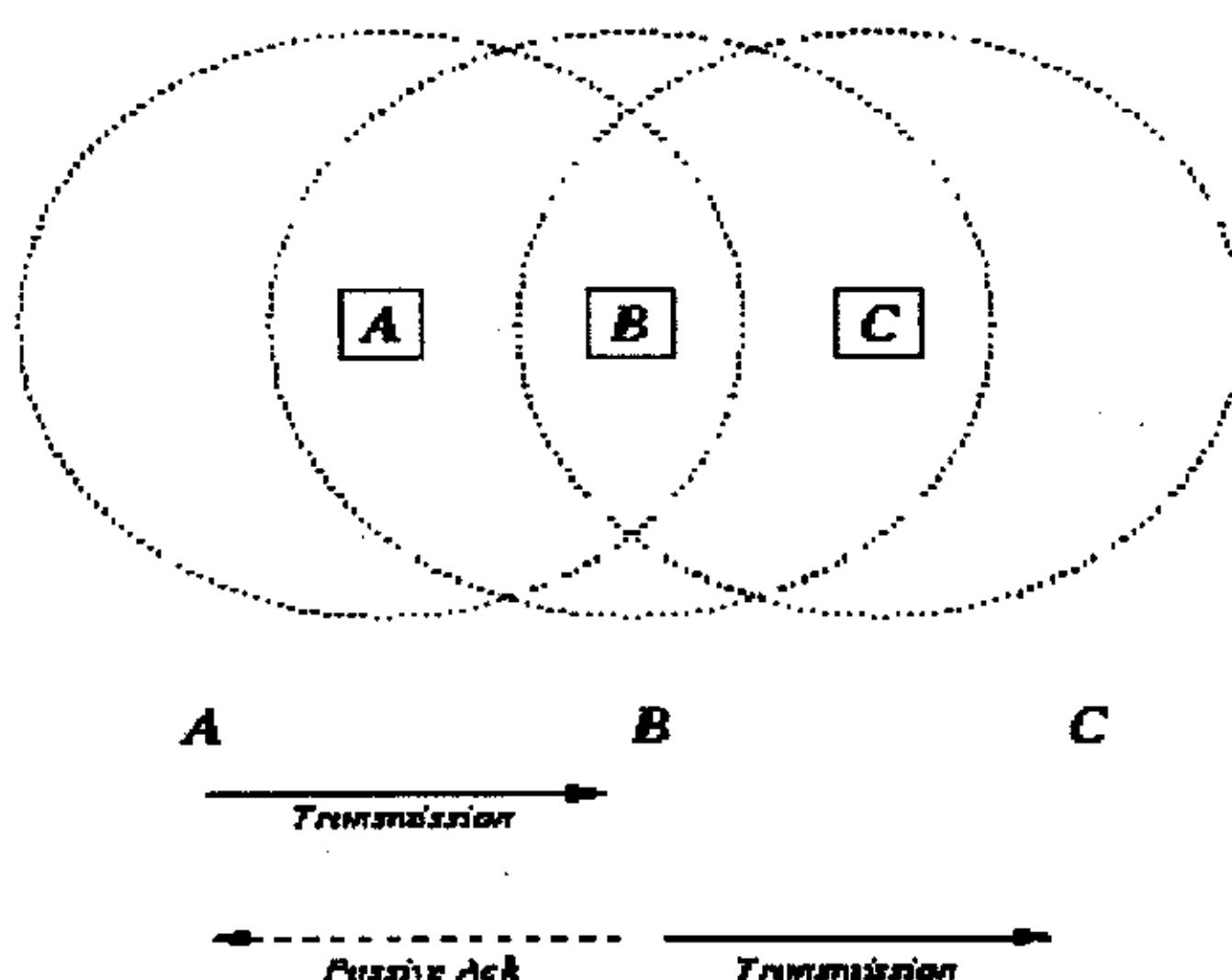


图 3-4 被动应答

由图 3-3，可以看到：一旦节点 I1 和 I2 收到来自 R1 的 JOIN TABLE，就会重建并发送自己的 join table 给下一跳节点（图中为 S1, S2）。在发送 join table 时，节点 I1, I2 可能会彼此重叠。如果 I1, I2 都在对方接收距离内，由于 CSMA 的载波特点会导致信号恢复。但如果不在信号范围内，他们将不会注意到节点 R1 的隐终端的情况，此时 R1 不能侦听到发生重叠的被动应答。这样，节点可能因为冲突引起的隐终端的问题而无法侦听其上游邻居节点的被动应答。同时，如果上游邻居被移走，节点也无法收到被动应答。在这种情况下，在一定的时间内，如果没有收到应答，那么节点就会重传消息。重传可以以单播方式选择要发送的邻居。如果包投递在一定数量的重传之后仍不能得到确认，节点就认为该路由已失效。从这个思想上看，最有可能引起路由失效的因素是链路上的节点出错或者离开了信号范围。这时必须找到另一条路由使节点向其邻居广播到某些信源的下一跳不可达的

消息。收到这个消息后，如果节点有到达组播源的路由，就会建立并以单播方式将加入应答告之其下一跳节点。如果不存在这样的路由，就只需告诉下一跳路由不可达即可。这两种情况下，节点都设置了 FG_FLAG 值。在实际应用中，这样的冗余在下次路由刷新形成的更有效的路由之前已经足够用于建立另一条路径。邻居节点设置 FG_FLAG 可能会导致过量的冗余，因为只有必要的转发组节点会在加入应答产生阶段被刷新，所以大部分的 FG_FLAG 标志会因超时而被清除。

3.3.4 路由选择标准

ODMRP 协议下，组播接收者根据最小延时（如收到的第一个路由请求中携带的路由信息）来选择最佳路径。在应用了移动预测之后，采用另一种选择方法。这种思想是在 ABR (associativity-based routing) 协议的启发下形成的。该协议选择相对稳定的路由。在新的算法中，不是选择最小延时的路径，而是选择最稳定的路径（例如，有最大的 RET 值的路径）。组播接收者收到第一个加入请求后必须等待一段时间，收到所有可能路径的 RET 值后才能作出路由选择。然后接收者选择最稳定的路由并广播加入应答。使用最稳定路由后，路由出错大大减少，加入请求包的数量也相应减少。图 3-5 显示了两种不同的算法

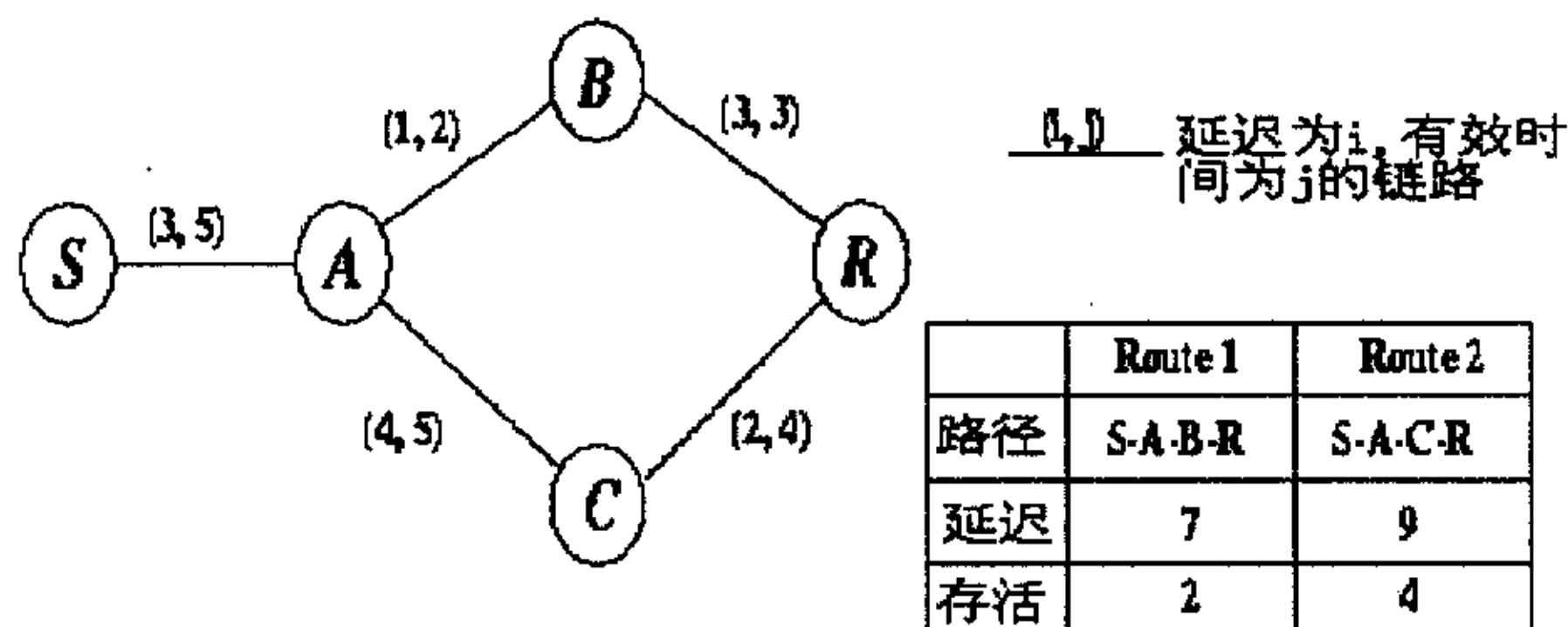


图 3-5：路由选择

下所得的结果。从信源 S 到接收端 R 有两条路径，路径 1 为 <S-A-B-R>，路径 2 为 <S-A-C-R>。如果采用最小延时作为最佳路径的度量，那么 R 选择路径 1，路径 1 的延时为 7 ($3+1+3=7$)，而路径 2 的延时为 9 ($3+4+2=9$)。因为加入请求选择路径 1 先到达接收端，因此节点 R 选择路径 1。如果以路径的稳定性作为选择标准，那么接收者就会选择

路径 2。路径 1 的路由存活时间为 2 ($\min(5,2,3)=2$) ,而路由 2 的存活时间为($\min(5,5,4)=4$).接收者选择 RET 最大的作为最佳路径，路径 2 入选。

3.3.5 减少路由形成延时

3.3.5.1 通过估测业务量大小降低网络拥塞

当信源有数据发送但无又无组播路由时，就直接广播数据而不是加入请求。周期性的加入请求被数据代替，一般来说，加入数据成为带数据的加入请求。这样，发送的加入数据在建立和刷新路由的同时，还完成了传送数据的任务，与单纯的加入请求相比，广播包大小增加，但是减少了路由延时。

ODMRP 的主要缺陷在于路由延时和网络拥塞。因为路由只在有需要时才建立，所以信源在数据发送前必须要等待路由的建立。为了减少这一延时，ODMRP 的加入请求包是携带用户数据进行寻路的。但这样的处理会使得广播包变大。从而会引起不必要的网络拥塞。通过对网络状态测量结果的分析，可以采用发送加入请求包之前估测用户数据包的大小方法。如果数据量太大，那么加入请求就不携带该包寻路，用户数据包只在路由完全建立后再发送。这样做将会带来更大的路由发现延时，但可以大大减轻 Ad hoc 网络的网络拥塞现象。

该处理方法如图 3-6(a) 中伪代码所示。

```

Procedure Packetprocess
    If(the receive packet is not a duplicate)
        And(flag for the multicast group os set)
        Then {
            If datasize>MAXSIZE
                Then{
                    Find route with join request data only:
                    Send date through the establish route;
                }
            Else find route carrying data;
        }
    }

```

图 3-6(a): 包处理伪代码

3.3.5.2 采用 probing 机制提高网络带宽利用率

按需路由最大的缺点在于获得路由前的延时。这一缺陷使得按需路由协议在有实时业务时性能下降。ODMRP 中，如果信源无组播路由信息，数据的传输就会有一定的延时，与单播相反，等待时间的选择不是直接的，单播环境下，信源一收到应答就可以发送数据，而在 ODMRP 中则不然，收到第一个应答后，远离接收者的路由可能还没有建立，所以数据的发送必须等待一定时间后才开始。

另外一个问题是，当数据发送开始时，由于网络中节点的移动，可能会造成路由失效，这时，如果继续以先前获得的路由信息发送数据，经过一段时间会返回目的地不可达的出错报告，这会大大降低数据传输效率。

通过网络测量，可以得到某一时刻的链路状态和本地业务情况。针对上述问题，为了更好的利用链路带宽，本文提出了采用探测机制来寻找路由的方法。

当链路不忙时，探测包被用来建立路由，探测包的发送周期小于 ODMRP 的加入请求包，可以为探测包的定时器设定了一个阈值。如果链路忙，即此时网络负载较大，那么该定时器的值就会被修改，以减小发送周期。此时路由算法和 ODMRP 比较接近。反之，如果网络不忙，即实时业务量小于标称业务量，那么探测定时器的值就会减小，从而使探测包在网络上出现的频率增加，但又不会引起网络拥塞。

探测包采用 ICMP 机制，根据 PING 原理来探测当前网络状态。Ad hoc 网络 ping 包的作用过程如图 3-7 所示。

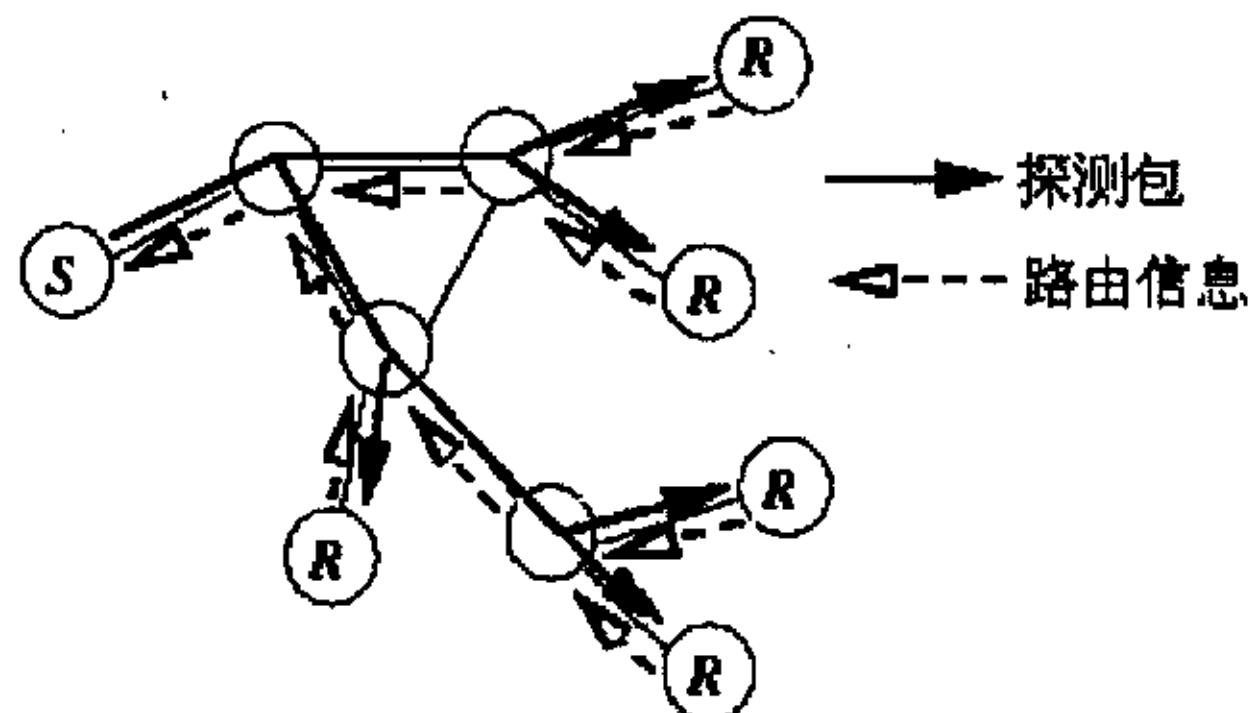


图 3-7：探测包的使用

可见，使用探测包来决定网络状态可以提高网络带宽的使用率。这对于网络传输，尤其在移动的 Ad hoc 网络中是尤为重要的，以下为

用探测包处理路由的算法伪代码(图 3-6(b))。

```

Procedure ProcessRoute
    While sendingdata
        Checknetwork status
        If(status=not busy)
            Decrease the value of probing timer;
        Else
            Increase the value of probing timer;
            Send probing packet;
            Store routing packet;
            Store routing information in the route table;

```

图 3-6(b): 路由处理

3.4 实现方法

3.4.1 概述

EODMPR 基于组播树而不是基于 mesh 结构，这样的组播协议可以在其组播成员间提供更快更有效的连通性。通过建立组播树拓扑和提供多条路由，组播包能够在节点移动和拓扑变化的情况下继续被发送到信宿，同时也避免了 mesh 结构的缺点（如业务过于集中，频繁重配置等等）。EODMRP 同时应用了按需路由的思想来避免链路负载过重，离开该组时，不需要显式的发送控制信息。链路负载的减轻和链路连接的多样性能够使得 EODMRP 在无线移动网络中有更好的表现。

3.4.2 数据结构

节点使用活动节点表、成员表和其他表维护 EODMRP 的组播转发状态，并使用路由表和消息缓存表共同完成数据包转发任务。

3.4.2.1 活动节点表 (sender table)

该表为所有组播组的当前活动节点建立一个条目。每个节点对应的条目内容包括当前发包间隔时间，已发送的 keep alive 包的数量。结构如下：

```

typedef struct node_state_ {
    int mcast_src_id;

```

```

        int prev_hop_address;
        int ka_ipt;
        int num_rev_joins;
        IDTable id_table;
        struct node_state_* next;
    } NodeState;
}

```

3.4.2.2 成员表(membership table)

该表记录了组播组中的发送节点和转发节点状态，每个成员表的条目包括接收节点标志，转发节点标志，当前节点向组播组发送数据包的时间间隔，以及从组播组中收到的 keep alive 包的数量。

```

typedef struct {
    short flags;
    int state;
    int mobility_counter;
    double last_pkt_time;
    double rdy_to_snd_out;
    int ipt;
    int ka_ipt;
    int ka_count;
    int max_ka_count;
    double mfact;
    int unacked_pkts;
} SourceState; /* 组播源的状态表 */

typedef struct {
    short flags;
    int state;
    int mobility_counter;
    int mobility_flag;
    int ipt;
    int hop_count;
    int parent_address;
    int ka_ipt;
    int ka_count;
}

```

```

    int mfact;
    int loss_metric;
} ReceiverState; /*接收节点的状态*/
typedef struct {
    short flags;
    int state;
    int parent_address;
    int hop_count;
    int unacked_pkts;
    int ipt;
    int ka_ipt;
    int ka_count;
    int mfact;
} ForwarderState; /*转发节点的状态*/

```

3.4.2.3 其他节点表(node table)

该表记录了网络中其他节点的状态。这些节点通过该表接收组播树的广播或全网广播。节点表的内容包括来自当前 EODMRP 包的包头序列号，再加上前一个包的序列号，用于发现或丢弃广播过程中的重复数据包。如果这两个序列号相等，说明接收到重复的数据包，如果后者小于前者，也为一重复的数据包，应予以丢弃。节点表中还包括了前一跳地址，该地址来自所收到的数据包的 MAC 层的传输信息。

```

typedef struct mcast_table_entry {
    int src_id;
    ReceiverState* recv_state;
    SourceState* src_state;
    ForwarderState* fwd_state;
    NodeState* node_state;
    struct mcast_table_entry* next;
    struct mcast_table_entry* prev;
} MulticastTableEntry;
typedef struct mcast_table {
    int group_id;

```

```

    MulticastTableEntry *table_entry;
    MulticastTableEntry *table_entry_tail;
    mcast_table* next;
    mcast_table* prev;
} MulticastTable;

```

3.4.2.4 路由表(route table)

路由表是由各节点按需建立和维护的。当节点收到非重复的加入请求时，相应的条目被创建或更新。节点存储信宿（加入请求的发送者）和倒数第二跳节点（最后产生加入请求的节点）的信息，路由表为发送应答提供了下一跳的信息。路由表的结构定义如下：

```

struct rtentry {
    struct rtentry *rt_next;
    u_int32          rt_origin;
    u_int32          rt_originmask;
    hort            rt_originwidth;
    u_char           rt_metric;
    u_char           rt_flags;
    _int32          rt_gateway;
    vifi_t          rt_parent;
    vifbitmap_t      rt_children;
    vifbitmap_t      rt_leaves;
    u_int32         *rt_dominants;
    u_int32         *rt_subordinates;
    u_int          *rt_leaf_timers;
    u_int          rt_timer;
    struct rtentry *rt_prev;
    struct gtable  *rt_groups;
};

```

3.4.2.5 消息缓存(message cache)

节点使用消息缓存来发现重复信息。当节点收到新的加入请求或数据包时，记录源 ID 和包的序列号。消息缓存中的内容不需要永久性维护。可以用 LRU (least recently used) 或 FIFO 来删除过时的内容，避免消息缓存溢出。

数据结构定义如下：

```
typedef struct message_cache {
    u_int32 Group_IP[Cache_Size];
    u_int32 Sequence_Number[Cache_Size];
    u_int32 Source_IP[Cache_Size];
    u_int32 Previous_Hop_IP[Cache_Size];
    struct timeval time[Cache_Size];
    int used[Cache_Size];
    int Count;
} Message_Cache;
```

3.4.3 组播数据包的转发

EODMRP 包头中目标地址为组播地址或广播地址的数据包以广播的形式发送，广播的类型由 EODMRP 包头中的类型字段决定。大多数包的广播类型都设置为树内广播，这样数据包就只沿着由组播源和组播组决定的组播树内的节点转发（见图 3-2）。节点收到数据包时，先检查自己的成员表，然后决定是否转发该数据包。数据包沿着组播树从组播源发送到组播组，但并不局限于在树中只沿着特定的树枝进行转发，这样在链路故障时，数据包可以自动的从故障点周围的其他节点上转发（见图 3-1）。如果 EODMRP 包头中的广播字段为全网广播，那么数据包被发送到所有节点。无论那种广播类型，节点处的节点表和 EODMRP 包头的序列号都能够保证转发节点不重复转发同一数据包。

节点收到数据包后，无论转发与否，都将节点表中的跳记数与包头中的跳记数做比较。如果新的跳记数值小于节点表中已有的值，那么节点将以新的跳记数值更新节点表相关内容，并将表中的前一跳地址设置为所收到的组播包的 MAC 层地址。另外，如果节点要转发该数据包，在转发之前，节点先增加 EODMRP 包头中的跳记数字段值，并

将包 MAC 层的源地址写入 EODMRP 包头中前一跳地址字段中。

最后，如果数据包在 EODMRP 包头之后还有载荷，那么节点检查成员表来决定自己是否为该组播组和组播源的接收者，如果是，则接收该数据包并将其当作收到的组播包进行处理。

3.4.4 对新组播源的处理

当某个组播组 G 的节点 S 产生组播数据包，且 S 并不是当前活动的发送节点，那么显然此时 S 没有对应于 G 的转发表。这种情况下，节点 S 建立并初始化一个新的转发表。表中的数据包发送间隔设为默认值。节点 S 还在包中插入 EODMRP 的包头，并将广播标志设置为全网广播。

发送第 1 个数据包后，节点 S 并不是一产生数据就马上发送，而是将数据包先放置在缓存中。这样可以使网络中只形成对该组和组播源感兴趣的接收者的路由状态。当 S 收到至少一个接收者加入请求包后才开始以正常的组播报向该组发送缓存中的组播数据包。当有新的组播源开始工作时，进行数据包的交换（见图 3-8）。多数来自节点 S 发往组 G 的组播数据包只在已经建立的组播树内传播，但也可能某个对该组播组感兴趣的接收者没有收到来自 S 发送的数据包，为了处理这样的情况，S 使用网络广播而不是树内广播发送后续数据包。全网广播的数据包的发送间隔逐步增加直至到达一定阈值，这样做是为了适应 Ad hoc 网络中无线接口的间歇性消失。例如，在我们的仿真实验中，第一个数据包在网络初始数据包发送 5 秒后以全网广播的形式发送，10 秒后发送的数据包也是作为全网广播发送。这种全网广播只在有数据从 S 发往 G 的时候才产生。

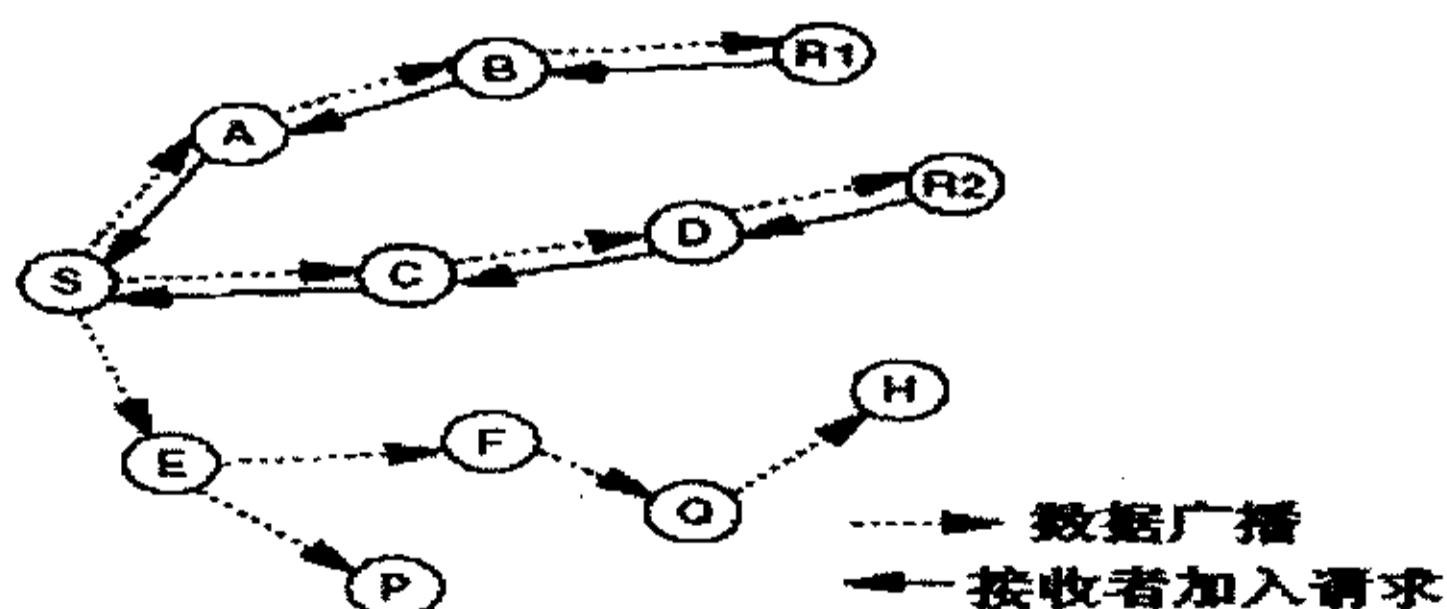


图 3-8：新的组播源 S

3.4.5 对接收者加入请求的处理

当某个节点 R 上的应用请求加入组 G 时，节点 R 的 EODMRP 路由以全网广播的形式发送组播请求包，该请求包的目的地址为组播组地址 G。如果该组是一个特定源的组播组，那么请求包的 EODMRP 包头中包括含该特定的组播源的地址。

包在网络中的传送过程如 3.4.3 所示。但是在特定源的组播环境下，组播源不转发组播请求包。在这种情况下，如果节点收到的组播请求有关该组播源的节点表项，并且关于该组和源的成员表项内容表明该节点是转发节点，那么节点只把组播请求以单播形式发送到节点表中所指的前一跳地址，然后包再沿着指向源的组播树发送。这样可以减少接收者加入请求的开销，并能提高其速度。

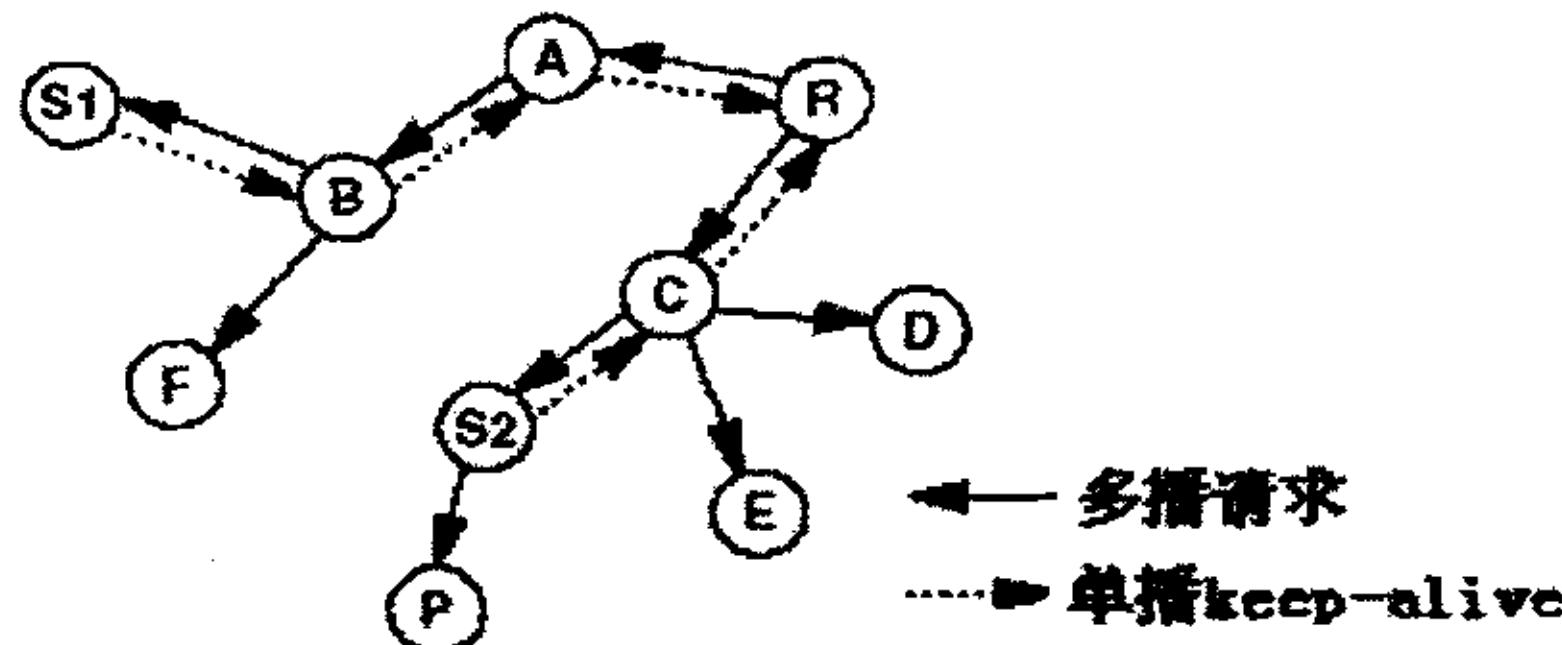


图 3-9：源 S1 和 S2 对接收者 R 的组播请求应答

当组播组 G 的源 S 收到组播请求包后，即向 R 作出应答，表明 S 是组 G 的发送者。该应答可以有两种形式：如果马上发送现有的组播数据包的全网广播，那么 S 选择在该时间之前进行全网广播，并用该广播作为对 R 的组播请求的应答，这种形式的应答非常有效，例如，当许多新接收者试图在同一时间加入该组时，因为 S 会收到来自所有接收者的组播请求，但不能使用单个的网络广播来应答所有的请求。另一种形式的应答是 S 向 R 发送 EODMRP 的 keep alive 包，该包沿着 R 发送组播请求包的路径传输，每个转发该单播 keep alive 包的节点都将该包发送到节点表中对应 R 的条目中的前一跳地址，该地址是当节点转发 R 到 S 的组播请求时建立的（见图 3-9）。当向 R 转发这个单播的 keep-alive 包时，每个节点都更新其节点表中关于 S 的表项，并记录反向到达 S 的前一跳地址字段。

当节点 R 收到组播请求的应答，将对其做如 3.4.5 所示的处理，因

为 R 把这个应答看作来自可能没有和它建立连接的组播源，R 还会向 S 反向发送接收者加入请求包，建立转发状态以与组播转发树建立连接（图 3-10）。

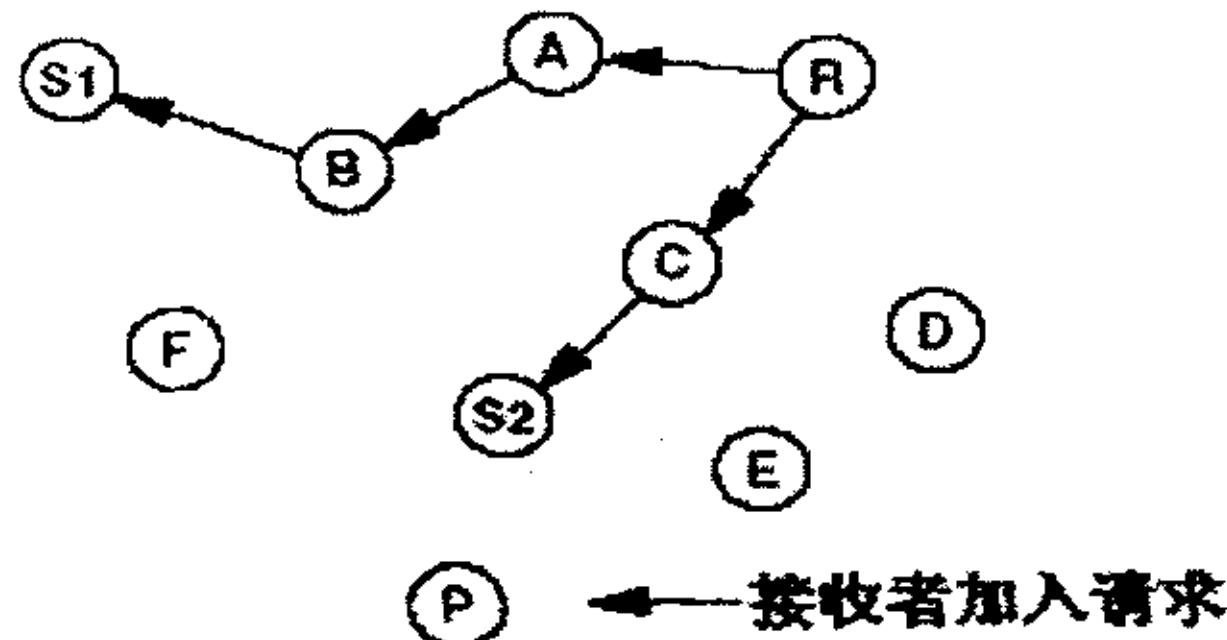


图 3-10：接收者 R 发送接收者加入请求

如果节点 S 通过发送单播 keep-alive 应答来自 R 的组播请求，那么 S 同时设置一个定时器，希望在短时间内收到来自 R 的接收者加入请求。如果 S 没有收到接收者加入请求，会重传对 R 组播请求的应答。如果定时器第二次终止，且 S 仍没有收到来自 R 的接收者加入请求，那么 S 认为通过转发 R 到 S 的组播请求而建立的发送单播 keep-alive 的路径出现故障，此时 S 重新计划其下一次用于应答 R 的全网组播数据包。

3.4.6 对来自新组播源的数据的处理

节点 R 收到组播数据包时，除了按转发进程的要求的转发该包外，还需要检查其成员表中关于该组播源和组播组的内容，以决定自己是否是接收成员，如果是，则把收到的数据包作为组播包处理，并将包按相关协议传送到网络下一层。

如果包以树内广播而非全网广播的形式发送，意味着接收节点 R 正与发送者和组播组保持连接。除非发现了连接失败，否则节点会认为自己一直处于连接状态。

如果包以全网广播发送，或者数据包是单播的 keep-alive 包，且如果接收者 R 当前没有与组播转发树连接，那么 R 就以接收者加入请求包作为应答，使得沿着反向到达 S 的路径上的节点都成为转发节点。节点 R 将接收者加入请求包的发包间隔字段初始化成来自所收到的包的 EODMRP 包头中的相应值，然后接收者加入请求包沿着由收到的组

播数据包或者 keep-alive 包形成的路径发送。每个转发接收者加入请求的节点，如果事先没有该组的成员表项，那么先建立这样的表项，同时将发包间隔时间初始化为接收请求包头中的相关值，每个转发接收者加入请求的节点还需要在起成员表相应条目中设置转发标志。

对于给定的组 G，如果有多个新的接收者，那么到源 S 的路径上就会有多个接收者加入请求包。但为了使每个节点都位于转发路径上，那么节点只需接收或转发接收者加入请求包一次就足够了，这样节点可能会过滤掉除了第一个之外的接收者加入请求包，引起新发送的接收者加入请求包的丢失。为了减少开销但同时又提供对丢失包的可恢复性，每个节点对节点表中记录的最后一个序列号，最多转发 3 个接收者加入请求包。

为了进一步处理接收者加入请求包的丢失，每个新的接收者在发送其接收请求后，设置一个值为接收到的包中的发包间隔时间几倍的定时器，用该定时器触发接收者加入请求。在仿真实验中，使用 3 倍时间间隔值。如果来自新组播源的数据包未到达之前定时器终止，那么发送方重新发送其接收者加入请求，并重置定时器。如果定时器没有收到新的数据包之前再次终止，那么接收方发送新的组播请求，如前文所述。

3.4.7 本地组播树修补

由于网络节点的移动或者无线传输环境的改变，组播组的转发或接收成员有可能与处于转发状态的组播树失去连接。当组播组 G 的转发者或接收者连续几次没有收到组播数据包时，就认为与组播转发树失去连接。

对给定的组播组 G 和组播源 S，节点维护一个连接定时器，当收到来自该组的包，就重置定时器。定时器的值根据最近收到的数据包的 EODMRP 包头的发包间隔值再加上和从源 S 到当前节点的跳记数成正比的时间，这一略微增长的定时器的超时值使得靠近 S 的节点比远离 S 的节点更早发现与 S 失去连接。这个特性并不是协议必需的，但能够提高修补进程的效率。

当某个节点 C 发现与组播树失去连接，就会初始化一个本地修补，如图 3-10 所示，节点 C 首先发送一个修补通知给其组播树的下游节点，这里的下游节点是由每个节点的节点表中定义的前一跳地址。这种关系代表了一系列通过节点 C 收到前一组播包的节点。这些节点以后很

可能发现因为连接定时器值的增长而导致失去连接的情况。

为了向节点 C 的下游节点转发修补通知，每个节点只接收和转发与节点表中的前一跳地址相匹配的数据包。另外，序列号和每个节点的节点表中的附加比特值用来避免重复发送修补通知。

发送完修补通知后，节点 C 在进行本地修补前先要经过一个修补延迟的等待周期，在这个等待时间内，如果节点 C 收到来自上游节点的对于同一组和源的修补通知，那么 C 放弃自己的本地修补。

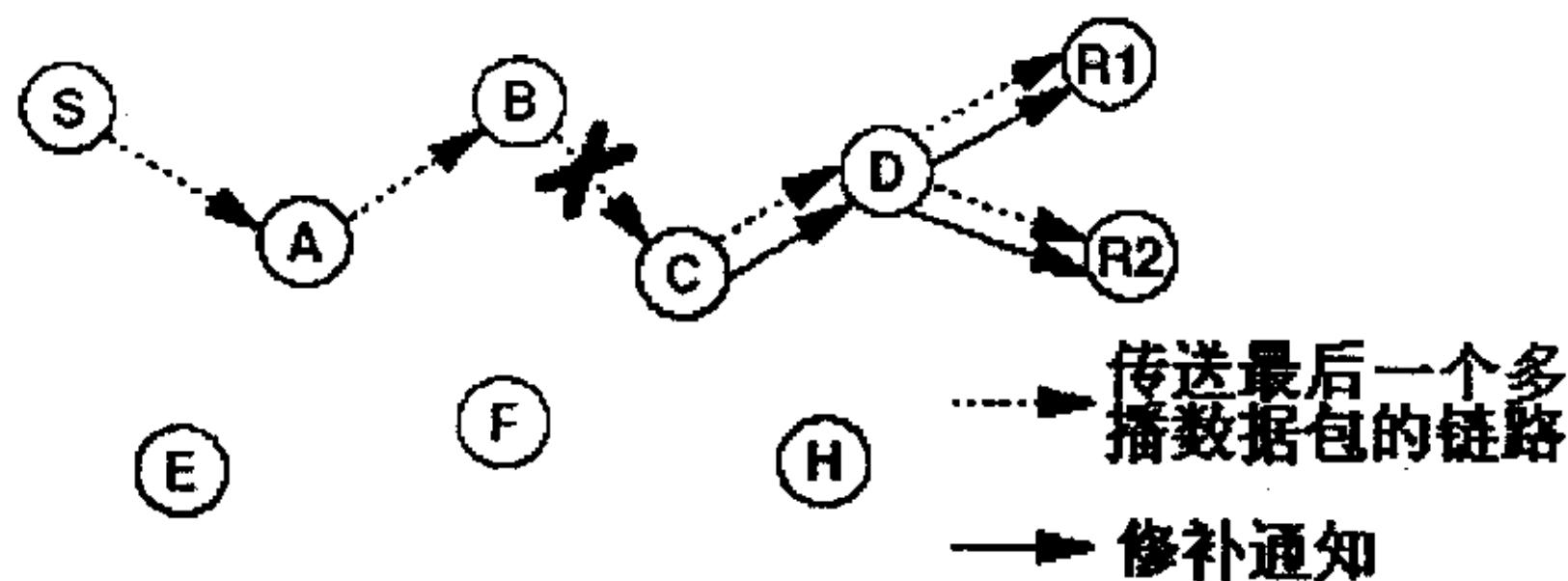


图 3-11：节点 C 的下游节点初始化本地修补

修补通知包有两个作用。首先，通知节点 C 的下游节点正在进行本地修补进程，告诉这些节点不要初始化其本地修补进程。另一个作用是再次检查到节点 C 的链路是否真的中断，转发树中所有节点 C 的下游节点都会收到修补通知，如果从 C 到其父节点 B（图 3-11）的链路没有真正断开，那么 B 也可能收到该修补通知。修补通知包中，C 列出其当前父节点的节点地址。如果这个父节点收到修补通知，会认为直连的子节点（节点 C）正在进行本地修补。然后父节点发送一个单跳的修补通知给 C，使其终止当前的本地修补。当组播组的接收成员收到修补通知后，将连接定时器的大小延迟一段时间，这段时间由估测的本地修补时间决定。

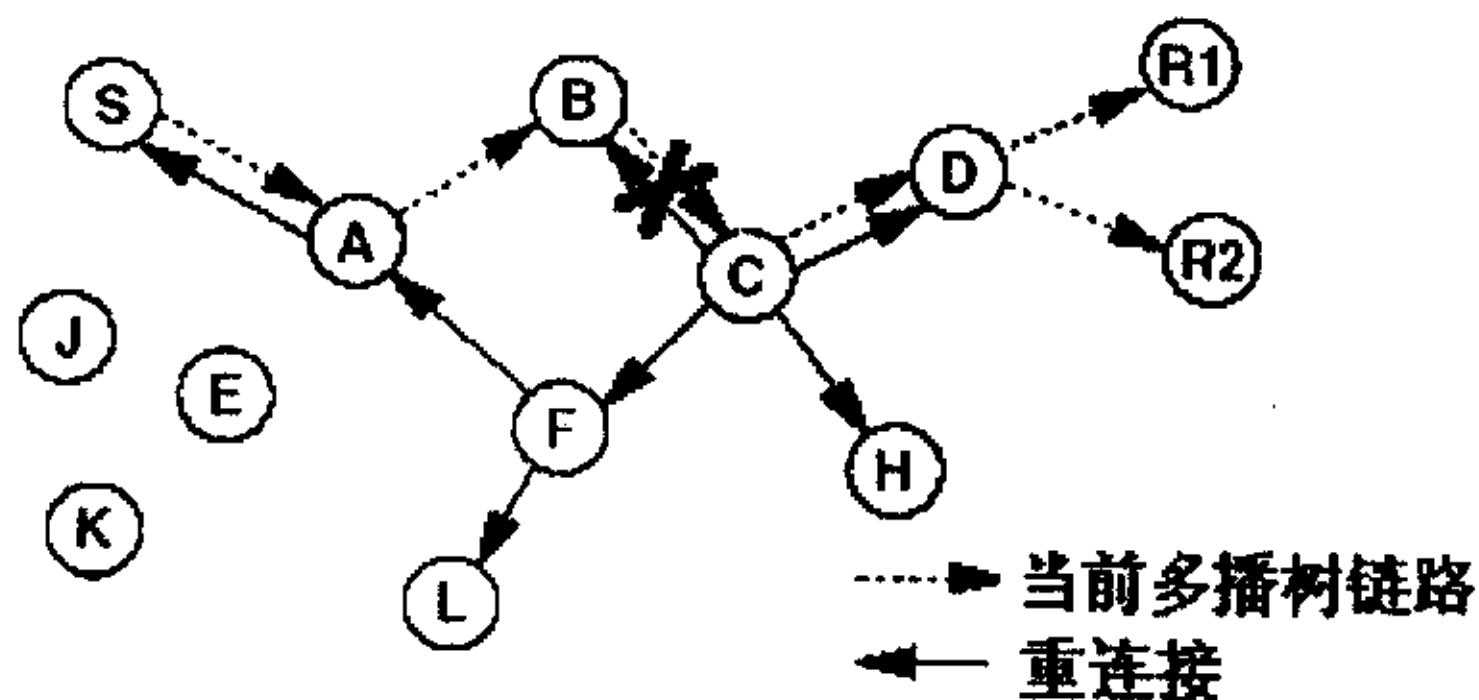


图 3-12：失去连接的节点 C 发送重连接包

经过上述一个延迟后，如果节点 C 还没有收到上游节点的修补通知，那么节点 C 以全网广播的形式发送一个单跳的重连接包（如图 3-12），重连接包定义了正在进行本地修补的组播组和组播源，重连接包的跳记数（TTL）限制该广播只到达 C 的邻近节点。

在决定是否转发重连接包时，除了对网络广播的普通处理外，作为组播组 G 和组播源 S 的转发节点，还必须对这些包进行一些特殊的处理，这类节点（如图 3-12 中的节点 A）如果没有收到本次修补通知，那么就认为自己是修补节点 C 的上游节点，同时还认为与组播树的源 S 仍然保持连接。节点不是将重连接包以有限跳数在全网转发，而是将包的 TTL 值重新初始化为默认值，并将该包以单播形式发送到其父节点。此时，该包不再作为全网广播包处理，而是被每个节点向其父节点转发，直至到达 S，如果节点实际上并不是修补节点 C 的上游节点，那么当单播重连接包到达节点 C 时，C 将丢弃该包。

如果重连接包到达节点 S，节点 S 以重连接应答包作为响应，如图 3-13 所示。这个重连接应答包沿着重连接包到达 S 的路径以单播发送。

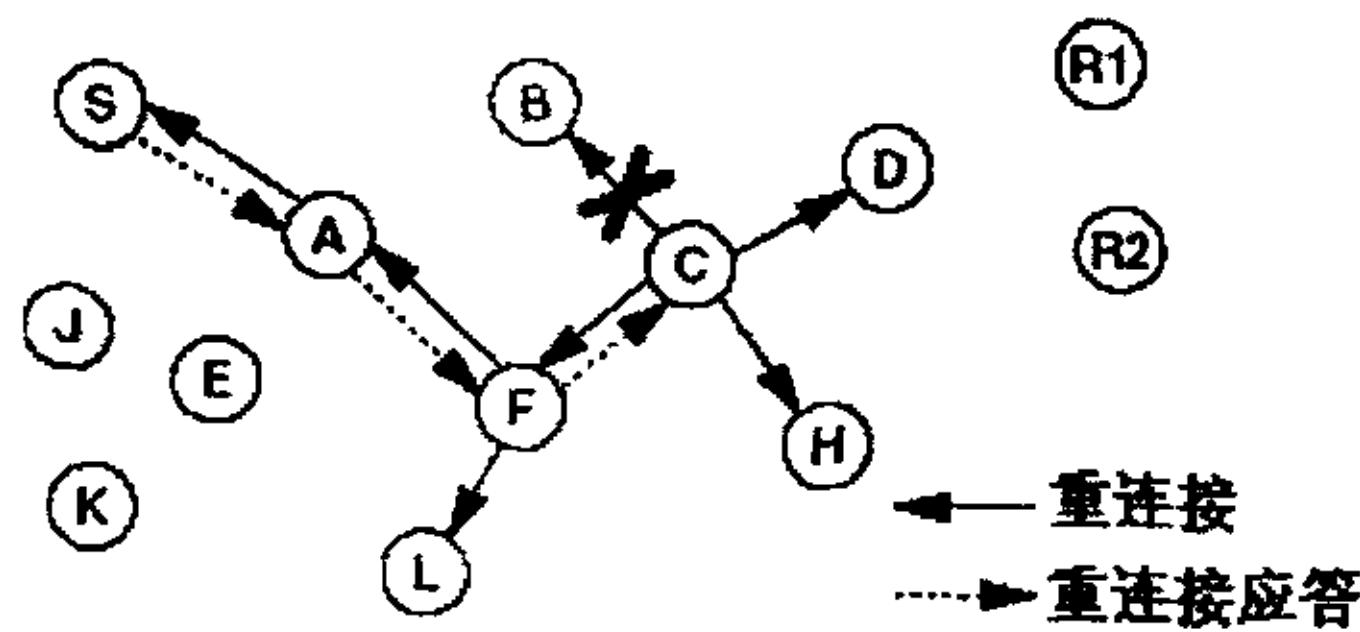


图 3-13：源 S 用重连接应答作为响应

重连接包到节点 C 的转发路径上的每个节点都成为组播组 G 和组播源 S 的转发者，并在各自的成员表中建立相应的条目。

3.4.8 由接收者初始化的修补

如果本地修补成功，那么组播转发树将和接收成员重新建立连接，进行数据的收发。如果连接定时器在组播组某个接收成员 R 处终止，则说明本地修补失败，这可能是因为网络中节点的大量移动对本地修补定义 TTL 造成困难。此时，节点 R 通过用和原来一样的方式再次加入组播组来执行本地修补，如 3.4.5 中所述。

每个接收者跟踪由于失去连接而需要进行重新加入组播组的次

数。当这个次数到达一个阈值时，接收者在接收者加入请求包 EODMRP 包头中设置“高速移动”的标志，当组播源收到几个含有这个标志的接收者加入请求时，将后面的组播数据包的发送方式改为全网广播。大量的重加入说明组播状态不能适应网络中节点的高速移动，只有通过广播才能成功发送数据。经过一段时间的广播后，随着网络中节点移动速率的降低，路由协议重新回到正常工作模式。

3.4.9 组播树的修剪

组播组 G 和组播源 S 的组播转发树的每个转发节点在决定不需要进行组播转发时会自动终止其转发状态并退出组播树。同样，组播源 S 在无下游接收成员时也能够自动停止发送组播数据包。发送节点继续在后台不定期的发送全网广播，但只有当收到至少一个新的接收者加入请求包时才会继续发送组播数据包，如 3.4.4 中所述。这一机制有助于从转发树中修剪由于下游节点不存在或失去连接而使得转发状态失效的节点。

状态终止的依据是组播数据包是否被收到该包的节点继续转发。为了决定每个数据包的状态，节点 B 希望发现至少一个从 B 收到数据包的其他节点 C 继续转发该数据包，当节点 C 收到并转发数据包时，在转发前，C 将所收到的包的 MAC 层的源地址（如节点 B 的地址）拷贝到 EODMRP 包头的前一跳地址字段中。如果节点 B 发现 C 在传输这个包，即认为这是在确认自己应该继续转发后续的组播包，这样节点 C 就可以继续接收到数据包。另一方面，如果 B 没有接收到续传组播数据包的确认信息，那么就认为没有必要继续留在该组播组的组播转发树中。这类似于被动应答技术。

3.5 总结

总而言之，EODMRP 的创新之处在于：

- 在全网范围内发送非周期性的控制包，进行端端节点探测和路由表交换。
- 根据数据发送的状态调整工作方式，实现高效的链路故障探测和无用路由的自动终止。
- 通过在组播树上发送有限的 keep alive 来处理突发源，这样可以识别缺乏数据和链路故障。

- 使用被动应答的方式提高组播树的自动修剪性能。
- 如果没有接收者，组播源只是不定期的广播现有的数据，并不传输其他数据或控制信息。
- 可以不借助 GPS 或其他定位信息及额外的控制信息来发现过高的移动速率，如果发现这样的情况，EODMRP 暂停当前工作转而以广播的方式发送所有数据包，一段时间后，当网络中的节点移动速度降低到一定程度，EODMRP 再次尝试用组播路由进行更高效的工作。

第四章 EODMRP 的性能分析与评价

4.1 仿真实验工具

随着计算机网络应用范围的迅速扩展，为了提供健壮的（robust）适应性强的计算机网络，必须对网络拓扑做出改进或创新，甚至对旧网络协议进行改进或者提出新的更高效的网络协议。网络仿真软件用于包交换数据网络中的数据流特性以及拥塞控制的研究，提供了指定网络特性的方法并仿真网络的行为。

当前较著名的仿真软件有 MIL3 公司版权所有的商业软件 OPNET (Optimized Network Engineering Tools) 和 Berkley 开发的源码公开的免费仿真软件 ns (Network Simulation)^[31]，ns 特别适合 TCP/IP 协议的仿真。出于性价比的考虑，选择 ns 作为 EODMRP 的仿真实现工具。

Ns 是一个事件驱动 (event-driven) 的网络仿真软件，用于网络研究领域，其用 C++ 编写的仿真引擎具有可扩展性，C++ 程序中使用了美国麻省理工学院 (MIT) 的对象工具命令语言 (Object Tool Command Language, OTcl)^[22] 语言作为仿真程序的命令与配置接口。

Ns 最初并不包含对无线网络的仿真功能，1998 年卡内基梅隆大学 (Carnegie Mellon University, CMU) 对其进行了扩展，增加了仿真无线网络的功能^[23]。

CMU 对 ns 的扩展为仿真的物理层、链路层以及网络层提供了新元件 (element)。如物理层的无线电传播模式 (radio propagation mode)、全向天线 (omni-directional antenna)、共享介质的网络接口 (shared media network interface)，数据链路层的 IEEE 802.11 (Distributed Coordination Function , DCF) MAC 协议、带载波检测多路访问 (carrier-sense multiple access , CSMA) MAC 协议、无线 ARP 协议 (Address Resolution Protocol)，网络层的无线路由协议 (如 EODMRP, DSDV, TORA, AODV)。此外还有专门用于生成随机场景 (scenario) 的软件，场景规定了移动节点的数量、移动和静止频率、移动速度、移动方向等。使用这些新元件，可以对无线子网、无线 LAN、或者 Ad hoc 无线网络进行细致精确的仿真。以下将把 CMU 扩展后的 ns 网络仿真软件简称为 CMU—ns。

图 4-1(a) 是 CMU—ns 中一个典型无线移动节点的结构示意图。为了能够在同一个仿真环境中同时运行单播和组播路由协议，本文对 CMU—ns 移动节点的

结构做了改进，除了对节点使用组播代理外。还加入了可以同时处理单播和组播数据包的代理。改进后的移动节点结构如图 4-1(b)所示。

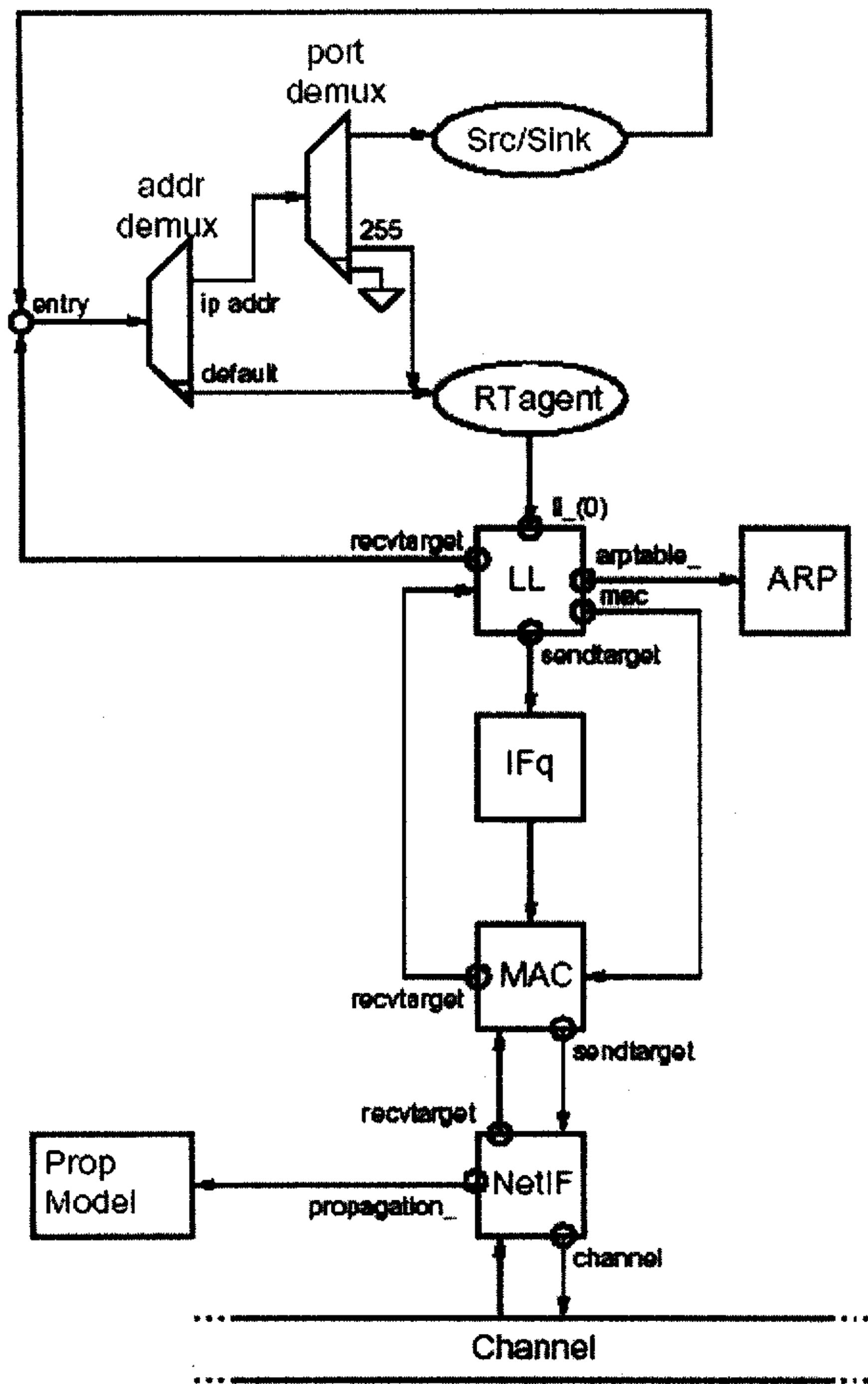


图 4-1(a)：CMU-ns 移动节点的结构示意图

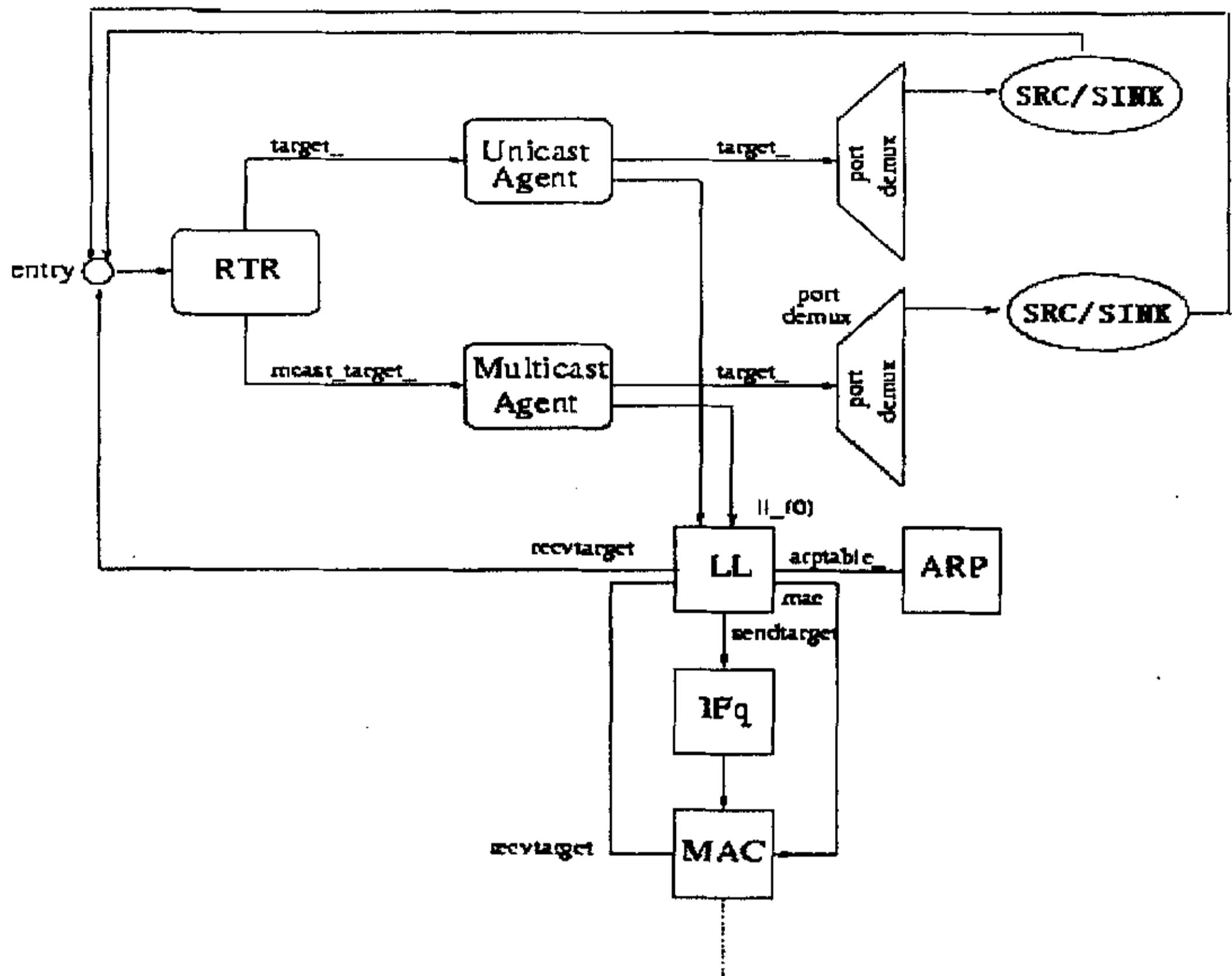


图 4-1(b)：改进后的 CMU-ns 移动节点结构示意图

4.1.1 信息处理

4.1.1.1 发出的信息包

从移动节点向外发出的信息包被交给该移动节点的入口点 (entry point) 处理，入口点把信息包转发给具有单播和多播区分功能的RTR，RTR查看节点的地址信息，如果是单播地址，那么把信息包传递给单播代理（即图4-1(b)中的unicast agent）。如果是多播地址，则把信息包传递给多播代理（即图4-1(b)中的multicast agent）。单/多播代理中的地址多路选择输出器检查信息包的目的IP地址，如果与本节点的地址一致，则将信息包传递到端口多路选择输出器 (port demultiplexer，即图4-1(b)中的port demux)。路由协议负责在信息包的包头设置下一跳节点的地址，然后把信息包向下传递给数据链路层协议（即图4-1(b)中的LL）。如果下一跳地址是IP地址，则LL对象会请求ARP对象把IP地址翻译成硬件地址。如果ARP对象当时没有该IP地址到硬件地址的映射，它就会将信息

包放入队列中，同时通知LL对象以硬件广播地址发出一个ARP请求包，进行地址解析（address resolution）。当得知信息包的下一跳的硬件地址之后，信息包就被放入接口队列（interface queue，即图4-1(b)中的IFq）中。MAC对象在适当的时候从接口队列的队头取出信息包，从网络接口中发送出去。网络接口在信息包的公共包头中记录一些特性值（如发送接口的位置）之后，把信息包发送到信道（即图4-1(a)中的Channel，图4-1(b)中被忽略）上，然后由信道为信道中所有的节点复制该数据包。

4.1.1.2 接收的信息

发送到信道上的每一个信息包都会有一个副本被送至该信道上的所有网络接口中，送达时间是根据节点之间的距离和光速计算出来的信息包的第一个比特（bit）到达该网络接口的时间。每一个接收网络接口都在信息包的公共包头中记录下本接口的特性值，然后唤醒传播模型(propogation model，即图 4-1(a) 中的 Prop Model）。传播模型根据发送接口和接收接口记录下的特性值计算出信息包到达接收接口时的功率。然后接收接口根据自己的特性值与信息包当时的功率计算自己是否确实能够接收到该信息包，如果能，就把信息包交给本节点的 MAC 层进行处理。如果 MAC 层完整正确地接收到了信息包，它就把该信息包传递给本移动节点的入口点。如果本移动节点是信息包的目的节点，则地址多路选择输出器将把信息包传递给端口多路选择输出器，然后由端口多路选择输出器把信息包交给合适的接收代理（sink agent）。如果本移动节点不是信息包的目的节点，该信息包就会被传递到地址多路选择输出器的缺省目标上，由路由代理（routing agent）为信息包设置下一跳的地址，然后传递给链路层。

4.1.1.3 移动节点上的用户层代理（user level agent）

移动节点载着代理（agent）在网络中移动并为代理提供路由层服务。有好几种方法可以把代理附着在移动节点上。在CMU-ns的方案中，建立、配置以及控制这些代理的命令位于通讯模式（communication pattern）文件中。通讯模式文件可以直接手工建立，也可以通过ad-hockey用菜单选项 Construction-Tools::SchedulPackets建立。Ad-hockey是一个用Perl/Tk编写的程

序，它可以辅助建立CMU-ns使用的场景文件以及将仿真跟踪文件可视化。作为一个说明性的例子，下面是CBR通讯模式范例文件中的一行代码：

```
set cbr_(4) [$ns_ create-connection CBR $node_(6) CBR $node_(7) 0]
```

其中create-connection是simulator对象的一个方法，定义在ns-src/tcl/lib/nslib.tcl文件中。该方法先创建CBR源和CBR接收器，然后调用节点的attach方法把它们分别与各自对应的节点关联起来。对于移动节点，create-connection方法调用位于cmu/scripts/mobile_node.tcl:MobileNode::attach方法中的如下代码：

```
$agent target [$self entry]  
$dmux_ install $port agent
```

上述代码做了两件事：1）为代理把target值设置为MobileNode的entry方法的返回值。（以后代理就把数据包发送到target）2）设置端口多路选择输出器，把目标端口号等于\$port值的数据包传递给代理，其中\$port是分配给代理的端口号。对于一般的MobileNode，方法entry的返回值是对地址分类器（address classifier）的引用。

如果用手工直接建立，首先可以采用如下命令行方式产生节点的组播通信模式：

```
./ns scripts/mcast_cbrgen.tcl -type cbr -nn 100 -rate 4 -ng 2 -nspg 2 -nrpg 10  
-leave_mg 1 -join_dur_r 1 > cbr_100_4x64_2x5x10_mcast_ns_rl
```

上例中type参数给出业务类型，nn参数给出节点数，rate参数给出节点最大移动速率，ng参数给出组播组的数量，nspg给出每组组播源的数量，nrpg给出每个组播组中的接收者的数量，leave_mg说明接收者在仿真过程中，除了可以加入组之外，是否可以离开该组，join_dur_r参数指明接收者在组中的停留时间是固定的还是任意的。产生的多播通信模式被存入文件cbr_100_4x64_2x5x10_mcast_ns_rl中用于仿真。

仿真实验中，除了需要多播通信模式文件，还需要产生一个节点移动场景文件，该移动模型采用一种称为随机路点（random waypoint）的模型^[24]，下列命令可以手动产生一个移动场景文件：

```
./setdest -n 20 -p 2.0 -s 10.0 -t 200 -x 500 -y 500 >scen-20-test
```

其中-n指定仿真节点的数量，-p指定停顿时间，-s指定最大速率，-t设定仿真时间，-x和-y设定仿真场景的范围，产生的移动场景文件为scen-20-test.

4.1.1.4 CMU-ns 中的 EODMRP 节点

图4-2所示为CMU-ns中EODMRP节点的结构。为简单起见，这里讨论单代理下的工作流程。EODMRP移动节点将地址多路选择输出器与路由协议代理合并成一个对象（EODMRPAgent）。这是因为在将来的实现中，EODMRPAgent可能会利用普通数据包夹带路由信息，而如果不把EODMRPAgent放在节点的入口处，则数据包就有可能不经过EODMRPAgent。这就意味着EODMRPAgent必须知道本节点的IP地址，这样目标地址为本节点的数据包才可能被向上传递给端口多路选择输出器。

由于使用EODMRP路由协议的移动节点与使用其它路由协议的移动节点具有不同的结构，所以创建EODMRP移动节点需要使用与创建一般的移动节点不同的方法。在cmu/EODMRP/EODMRP.tcl文件中定义了MobileNode的一个子类——SRNode，使用它来创建EODMRP移动节点。具体地说，

cmu/EODMRP/EODMRP.tcl文件中的过程create-mobile-node中调用“set node_(\$id) [new SRNode]”而不是调用“set node_(\$id) [new MobileNode]”来创建EODMRP移动节点。

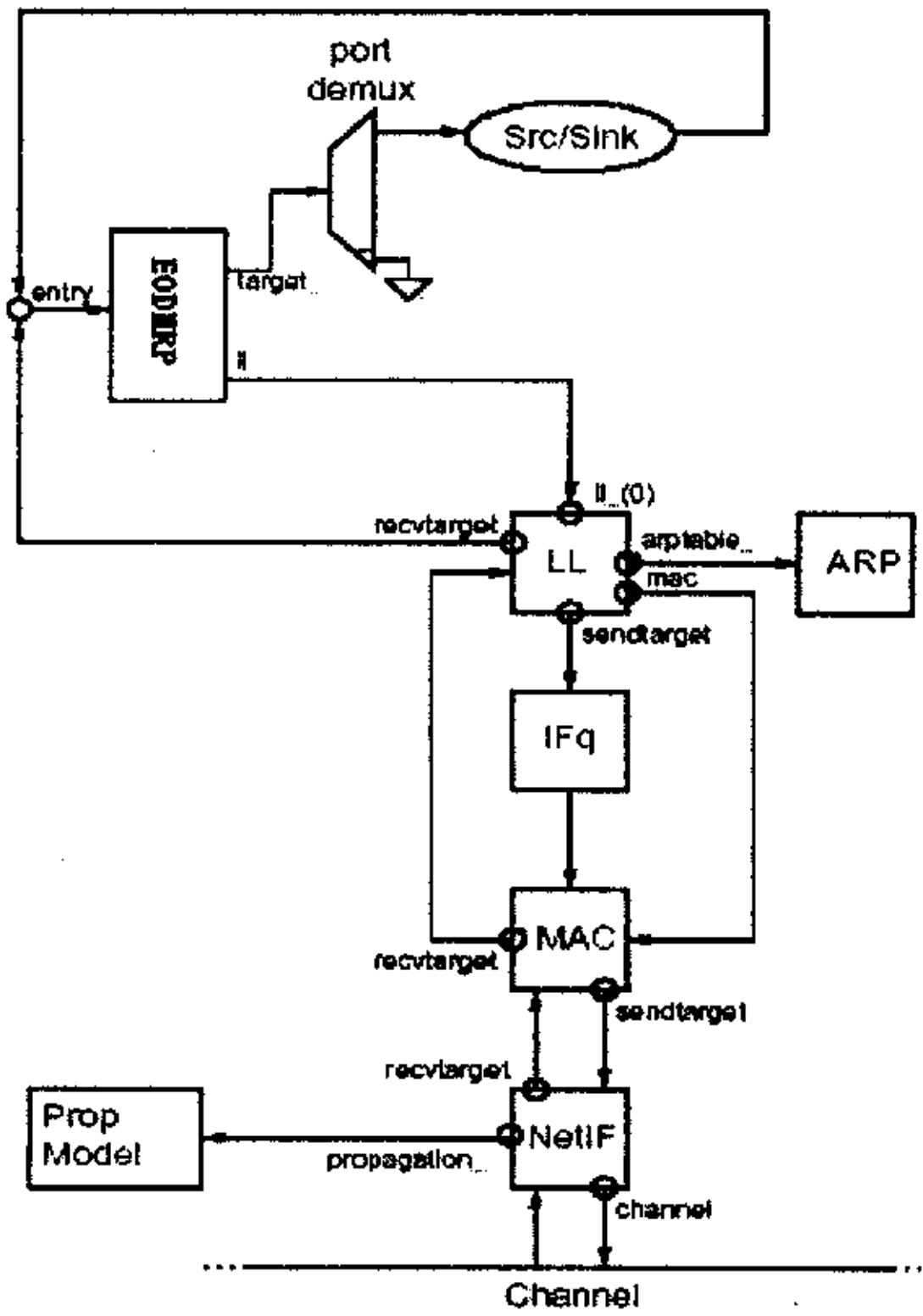


图4-2: CMU-ns EODMRP节点的结构示意图

4.1.1.5 EODMRP 对 CMU-ns 的改进

本文的EODMRP是在CMU-ns的ODMRP路由协议的基础上实现的，主要对其中的两个类进行了改进，其一是EODMRPAGent，它对应于图4-2中的EODMRP，该类原来实现的是ODMRP路由协议，笔者在其中添加了端到端的路由探测的机制。其二是MobiCache，该类原来实现的是ODMRP路由协议所使用的路由缓存，笔者在其中添加了使用多条路径的功能。此外，由于EODMRP需要发送路径探测包，它不同于标准的EODMRP路由包，因而需要特殊标识，所以还对结构hdr_sr做了修改，添加了一个标识字段。

需要注意的是，EODMRPAGent 与 MobiCache 是关联紧密的两个类，它们虽然各司其职，但又互相协作和调用，共同实现了完整的 EODMRP 路由协议。EODMRP 所增加的多径路由功能也是由它们合作实现的。比如，当 EODMRPAGent 要为应用业务发送数据包时，它调用 MobiCache 的 findRoute 方法为该数据包寻找合适的路径，这时 MobiCache 就要为 EODMRPAGent 提供多路径服务，即为发往同一目标主机的一系列数据包提供不同的路径，所提供的路径的总体更换频率由路径权重的计算中权重计算公式的系数 R 控制。另一方面，EODMRPAGent 要为 MobiCache 的多条路径发送路径探测包，且当路径探测包返回后，还要对 MobiCache 中所保存的路径特性值做更新或修改。

4.1.2 EODMRP 运行参数的设置

在为EODMRP寻找最优参数以及对各种参数对EODMRP的性能影响的过程中，Tcl带来了很大便利。比如，为了控制EODMRP路径探测包的发送频率，笔者在类EODMRPAGent (class EODMRPAGent{...}) 中定义了两个变量：double mul_timer_interval_ 和 int ping_timer_counts_ 前者是实现的一个定时器的周期，后者指定经过多少个定时器周期之后将发送路径探测包，二者的乘积即为 EODMRP 路径探测包的发送周期。在EODMRPAGent::EODMRPAGent()构造函数中，将上述两个变量与Tcl的两个变量进行绑定：

```
bind("mul_timer_interval_", &mul_timer_interval_);
bind("ping_timer_counts_", &ping_timer_counts_);
```

于是，在文件EODMRP.tcl中，可以通过如下语句为上述C++程序中的两个变量赋值：

```
Agent/EODMRPAgent set mul_timer_interval_ 0.1
```

```
Agent/EODMRPAgent set ping_timer_counts_ 20
```

仿真程序开始运行时，会执行EODMRP.tcl文件中的命令，这就把EODMRP路径探测包的发送周期设置为2秒钟。为了确定最优的路径探测周期，只需修改EODMRP.tcl文件中的两条赋值语句末尾的两个数字，保存EODMRP.tcl文件之后再次运行仿真程序。通过观察、比较、分析仿真输出结果，确定先后两个路径探测周期的优劣。在此基础上，继续实验其它的路径探测周期，最终找到最优化。顺便指出，在上述实验过程中，每次修改eodmrp.tcl文件后，并不需要重新编译C++程序，这就大大加快了实验速度。

4.1.3 数据处理

CMU-ns 输出的仿真跟踪文件(trace 文件)数据量大、数据种类多，而且可读性比较差。为了便于观察性能、发现规律，重新编写了 mcast_totals.pl 来处理所得到的仿真数据。统计仿真跟踪文件中被关心的数据，如应用程序的通讯量、丢包率、延迟等等。通过 mcast_totals.pl 得到的文件，能够很直观的反映仿真中的一些主要数据，以下是通过 mcast_totals.pl 处理后的部分数据结果。

t.mrtr_tx = [2195 88346]	% total control packet transmissions
t.mrtr_rx = [550952 41229404]	% total control packet receptions
t.mrtr_fx = [39523 3080551]	% total control packet forwards
t.df_tx = [259 26936]	% # of data floods transmissions
t.df_rx = [331986 34526544]	
t.df_fx = [25542 2656368]	
t.uka_tx = [91 3640]	
t.uka_rx = [317 12680]	
t.uka_fx = [227 9080]	
t.rj_tx = [270 7830]	
t.rj_rx = [689 19981]	
t.rj_fx = [420 12180]	
t.ms_tx = [60 1740]	
t.ms_rx = [82736 2399344]	

```

t.ms_fx = [ 5940 172260 ]
t.ka_tx = [ 212 6784 ]
t.ka_rx = [ 86529 2768928 ]
t.ka_fx = [ 4958 158656 ]
t.rn_tx = [ 1223 39136 ]
t.rn_rx = [ 29977 959264 ]
t.rn_fx = [ 494 15808 ]
t.rc_tx = [ 40 1160 ]
t.rc_rx = [ 18559 538211 ]
t.rc_fx = [ 1823 52867 ]
t.rr_tx = [ 40 1120 ]
t.rr_rx = [ 159 4452 ]
t.rr_fx = [ 119 3332 ]
t.ack_tx = [ 0 0 ]
t.ack_rx = [ 2231295 214204320 ]
t.ack_fx = [ 172207 16531872 ]
t.cbr_tx = [ 28965 1853760 ]
t.cbr_rx = [ 552865 35383360 ]
t.cbr_fx = [ 532752 51144192 ]
t.cbr_rx_total = [ 552865 552865 ]           % total cbr packet receptions
t.cbr_srx_total = [ 561423 561423 ]         % # cbr packets that should have been
                                              received over all receivers
t.cbr_opt = [ 806280 0 0 ]                  % # of packets transmitted along a given
                                              path length % in the optimal case
                                              (the path length is 0, 1, 2 etc.
                                              corresponding to the position of the
                                              number in the % array)
t.cbr_act = [ 127359 98745 0 0 ]            % number of packets actually forwarded
                                              along a path of a certain length
t.cbr_dif                                     % difference between t.cbr_act and t.cbr_opt
t.cbr_lat                                     % average latency experienced by each receiver

```

通过 mcast_totals.pl 的处理，我们能够更容易的根据仿真结构分析和改进路由协议。

4.2 仿真实现

4.2.1 仿真 Ad hoc 环境

仿真采用 Ad hoc 网络的移动模型和网络业务模型。移动网络的范围为 1500 米×300 米，节点数为 50 和 20，仿真时间为 300 秒。每次仿真执行一个包括所有 Ad hoc 网络节点移动的场景，所有应用层的通信由节点产生，并可以重现，这样就可以将同一场景应用于不同的协议测试中。

仿真环境中的节点采用一种称为随机路点的(Random Waypoint)的模型^[24]，每个节点从一个随机选择的位置开始，随机地选取一个目的地，然后以一定的速度向该目的地行进，在到达该目的点之后，停顿一段时间，然后再向另一个随机的目的点行进，每个节点在一次仿真时间内独立重复这种移动方式。

仿真实验中，有一个节点的最大移动速度为 20 米/秒，其他节点的最大移动速度为 1 米/秒，对每个最大移动速度，选择了 6 个不同的停顿时间（分别为 0, 30, 60, 120, 200 和 300 秒），停顿时间 0 表示网络中所有节点都连续移动，而停顿时间 300 表示静止的网络。对于每个停顿时间和最大节点移动速度，随机产生 10 个不同的场景，并以这 10 组场景下的平均值作为最后结果。

仿真环境中的应用层多播源产生 CBR 业务，每个节点每秒产生 100 个 256 字节的包。选择这一发送速度能够最好的测试协议在 Ad hoc 网络中传送数据包的性能。

本文使用了 2 种不同的多播组，多播源和多播接收节点的组合方式。前 2 种情况中，包括一个多播源和 10 个多播接收节点。在不同的节点最大移动速度下，使用场景容易观察理解和分析的多播环境中路由协议的性能。第 3 种情况中包括 2 个多播组，每组 2 个多播源和 15 个接收节点。这种情况用来测试路由协议在多个大型多播组的场景下的性能。所有情况下，多播源发送数据和多播接收节点加入多播组都在同一随机产生的时间，该时间值在 0 到 180 秒之间。

4.2.2 性能评价度量

以下为对 EODMRP 和 ODMRP 进行性能评价的度量：

投递率：即接收端收到的由应用层的多播源产生的多播数据包数对发送端发出的多播数据包数的比率。例如，在有 1 个发送者和 2 个接收者的多播组中，2 个接收者收到的多播包的总数应该是发送端产生的多播数据包数量的 2 倍；如果产生了 100 个多播包，接收端收到 150 个包，那么投递率就为 0.75。

负载开销：转发每个多播包所对应的控制包的开销。这个度量值反映了路由协议的基本多播转发效率。在实验中，针对目标网络的每一条链路的利用率，在每个采样时间段内进行统计计算，链路利用率以采样时间段内的传输速率与链路容量之比来计算。计算公式如下：

$$B_R(n,i) = [B_r(n,i)/T] / B_N,$$

其中 $B_R(n,i)$ 是第 n 条链路在第 i 个采样时间段的链路利用率。 $B_r(n,i)$ 是第 i 个采样时间段内第 n 条链路传输的数据总量； T 是采样时间段的长度； B_N 是第 n 条链路的链路容量，即带宽。

端一端延迟：每个包从发出到被成功接收的端到端的延迟，

包括在 buffer 中等待可用路径的延迟，每个输出网络接口中的延迟，MAC 层的竞争延迟，以及线路中的传输延迟等。该值由多播接收者单独计算。在实验中，目标流在包一级的延迟计算。每一个包的时间戳和序列号都被传输端和接受端记录下来。延迟被作为数据包发送和到达之间的间隔来计算。计算公式如下：

$$R(n) = [T_r(n) - T_{(s)}(n)],$$

其中 $R(n)$ 是第 n 个包的端端延迟， $T_r(n)$ 是第 n 个包的收到时间戳， $T_{(s)}(n)$ 是第 n 个包的发送时间戳。

投递率和端一端延迟是 best effort 业务最重要的参考指数。在 Ad hoc 路由性能研究的文章中常常作为最主要的性能参数。另外，由于我们采用基于 probing 的主动测量的机制来控制负载平衡，probing 的负载也应考虑。

4.2.3 EODMRP 和 ODMRP 的仿真参数

EODMRP 的仿真实验中，数据广播周期为 30 秒，缺少数据时等待时间间隔的乘积因子为 1.2，连续丢失两个包即触发无连接探测。

4.3 仿真结果

对每个 EODMRP 和 ODMRP 的仿真，使用了 6 个不同的停顿时间，2 个不同的节点最大移动速度，和 3 个不同的多播组，多播源及多播接收者的组合方式。仿真结果中的每个结果都是各种场景和仿真参数下 10 次仿真的平均值。

图 4-3 给出了 EODMRP 和 ODMRP 的包投递率，该场景有 20 个节点，1 个多播源 10 个接收者。仿真结构图的 Y 轴表示包投递率大小，数值为 0 到 1.0 之间。EODMRP 和 ODMRPD 对产生的多播树数据包的投递率都能达到 60% 以上，甚至在高速移动的网络（停顿时间小）中也是如此。EODMRP 的平均投递率比 ODMRP 高大约 10%，而且在控制包开销方面，EODMRP 也大大低于 ODMRP（图 4-4）。ODMRP 除了在网络上频繁发送广播，重建转发组，还在建立冗余数据发送路径方面产生了大量的开销。而改进后的 EODMRP 只需少量的控制开销进行数据。

如前所述，ODMRP 在网络的节点上建立转发状态，当该节点不需要继续进行转发时，并不是马上终止转发状态而是再经过一段固定时间。这段时间设置为加入请求广播间隔时间的倍数，这样可以保证广播包的丢失不会破坏多播数据的发送。但是，因为新的节点也可能成为多播组中的转发者，而已经建立的转发节点仍然还有转发标志，可以继续转发组中的多播包。所以这种机制会引起网络中产生的冗余状态。当 ODMRP 产生的冗余随着丢包而增加时，会给网络和相关节点的能量消耗带来负担。随着负载的增加，网络性能下降，包传输延迟显著增加（图 4-5）。对 EODMRP 而言，当多播树发生链路故障需要修补时，也会建立冗余路径，但多播树中不再包括链路故障前的多播树中的那些节点。但当下游节点中不存在对该多播组数据感兴趣的成员时，EODMRP 中的转发节点会自动终止当前状态。

路由协议在单多播源的另外一个场景（50 个节点）下，当接收者增多时，处于转发状态的节点也会增多，即转发状态节点的密度增加。这样自然就产生了更高的冗余性，两种协议在进行多播数据转发时都利用了这个冗余性。同时，因为大量的转发节点要求与大量的接收节点建立连接，所以 ODMRP 中可以成为冗余转发节点的节点数量就减少了。图 4-7 中所示的包的开销是通过将结果按接收者数量标准化后得到的，所以在接收者数量越多，所显示的开销就越低。停顿时间为 30 秒的情况下，发送相同数量的包，ODMRP 的开销是 EODMRP 的 7 倍。

在节点最大移动速度为 10 米/秒的情况下，EODMRP 和 ODMRP 的投递率都有所上升。因为移动速度的降低使得节点冗余要求降低，因此包投递率的表现要优于 20 米/秒的情况，其开销也低于 20 米/秒的场景。在 EODMRP 中，这些数值的降低是因为多播树链路故障的减少，而在 ODMRP 中，则主要因为冗余的减少。

图 4-9, 4-10, 4-11 显示了高负载场景的结果，该场景中包括 2 个多播组，每组 2 个多播源，15 个接收者。因为场景中每个多播组存在 1 个以上的多播源，因此 ODMRP 中的转发节点还要为其他多播源产生的数据做转发，这样就产生了额外的冗余和开销。特别值得注意的是，包不仅向接收者做转发，还转发到组中的其他多播源。

在这些场景中，网络的负载要明显高于前几个场景，包投递率也下降。由于 ODMRP 的冗余性，在这些场景下，显示了比 EODMRP 略好的性能（图 4-9）。但是转发每个数据包时，ODMRP 所用的发送次数多于 EODMRP（图 4-10）。EODMRP 除了能保持低开销外，还能在单多播源的情况下保持较低的端端延迟（图 4-11）。

当网络中接收节点占较大比重时，ODMRP 的性能表现较好。尽管 EODMRP 在网络接收成员较为分散的情况下路由性能达到最佳，但在这种情况下，EODMRP 的表现也不错。很难找到使 ODMRP 象 EODMRP 一样同时有高投递率和低开销的参数。ODMRP 对节点移动，特别是只有少数节点参与网络数据包转发时的移动尤为敏感。而 EODMRP 的对节点移动的良好的适应性使得它能够获得低开销，高投递率和低延迟的特性。

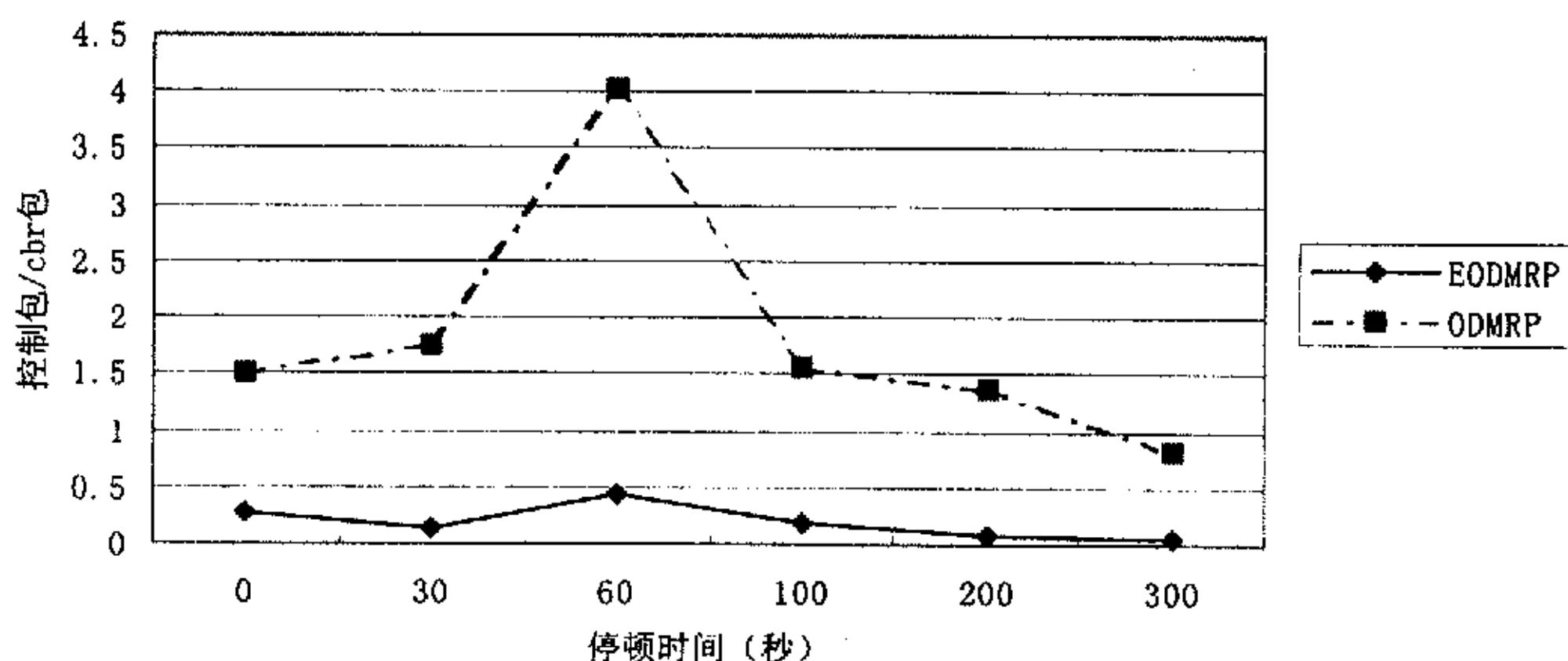


图 4-3：EODMRP 和 ODMRP 的包投递率
(20 个节点下，1 个组播组，1 个组播源，每组 10 个接收者)

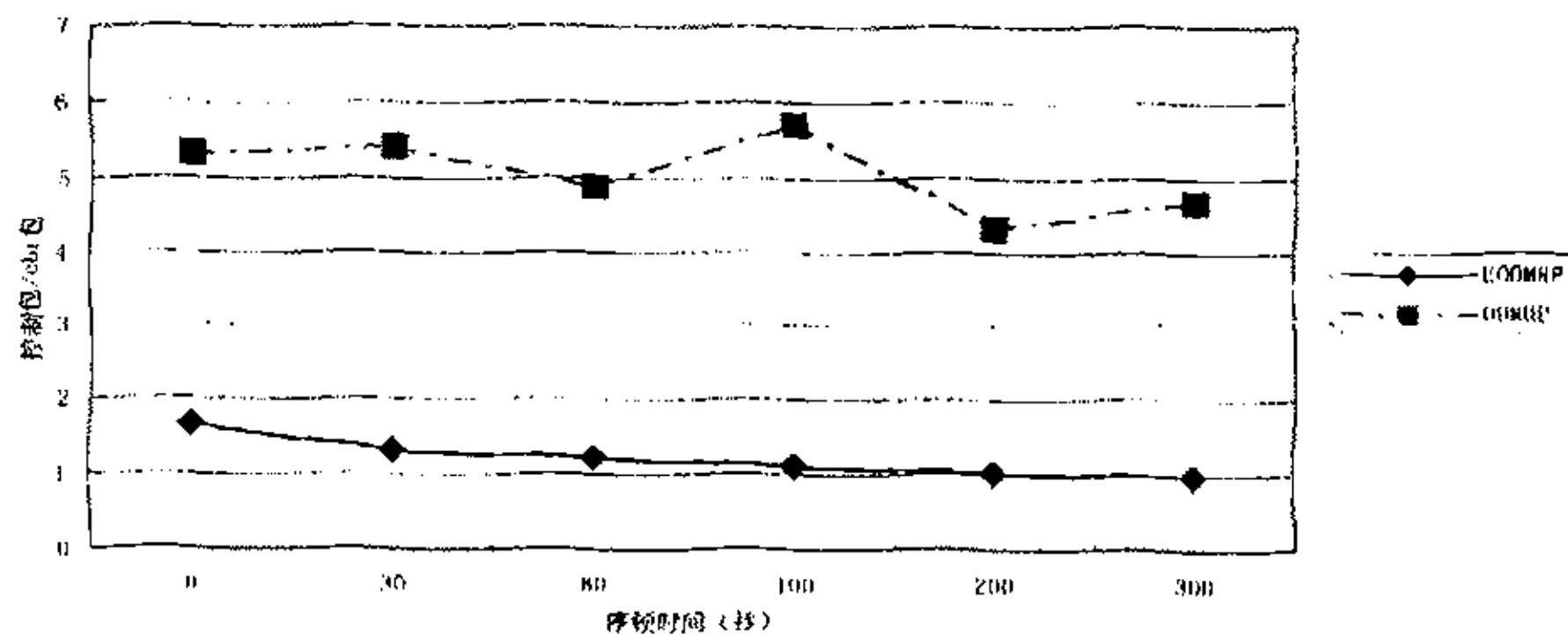


图 4-4: EODMRP 和 ODMRP 的控制开销
(20 个节点下, 1 个组播组, 1 个组播源, 每组 10 个接收者)

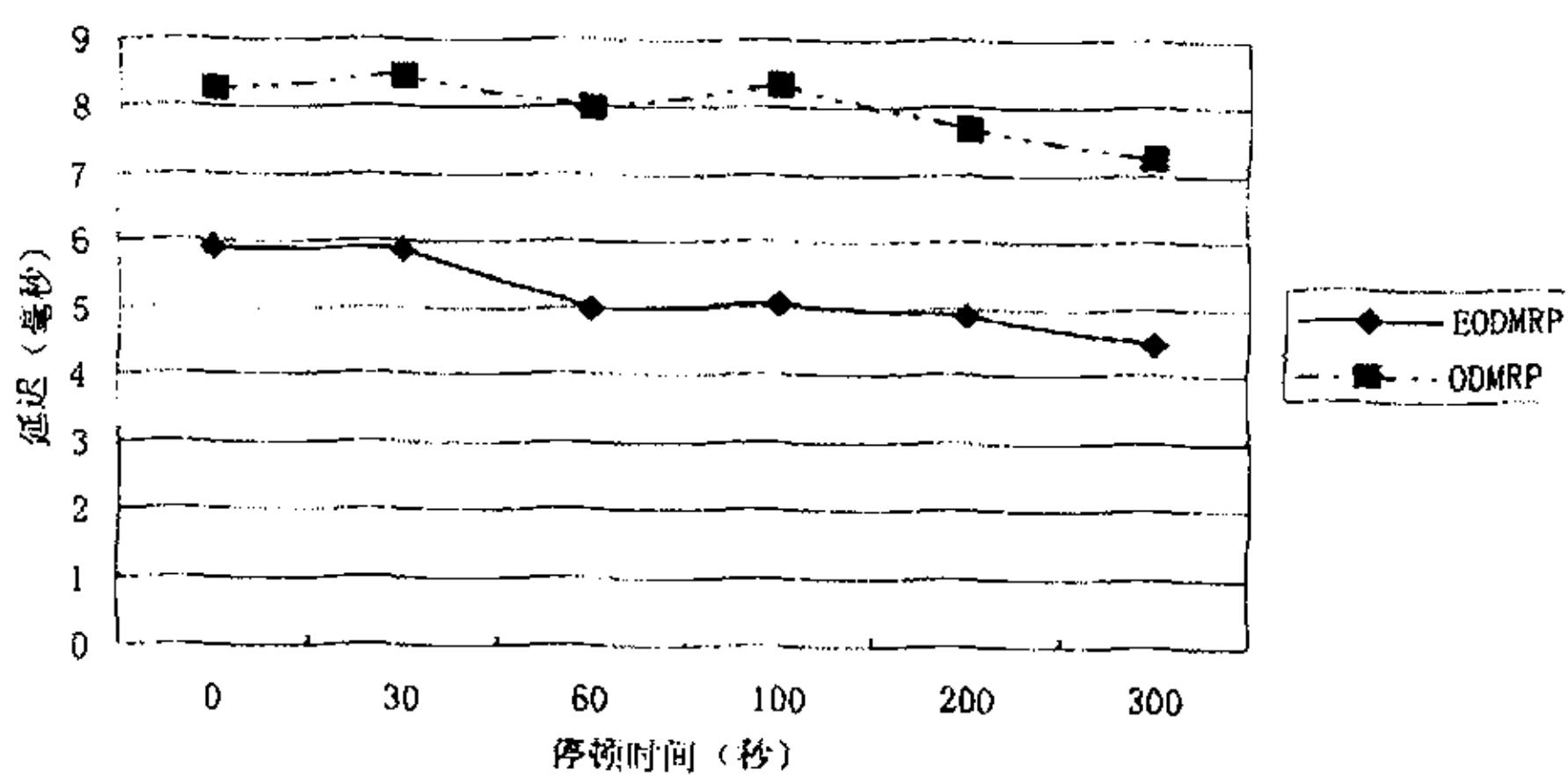


图 4-5: EODMRP 和 ODMRP 的传输延迟
(20 个节点下, 1 个组播组, 1 个组播源, 每组 10 个接收者)

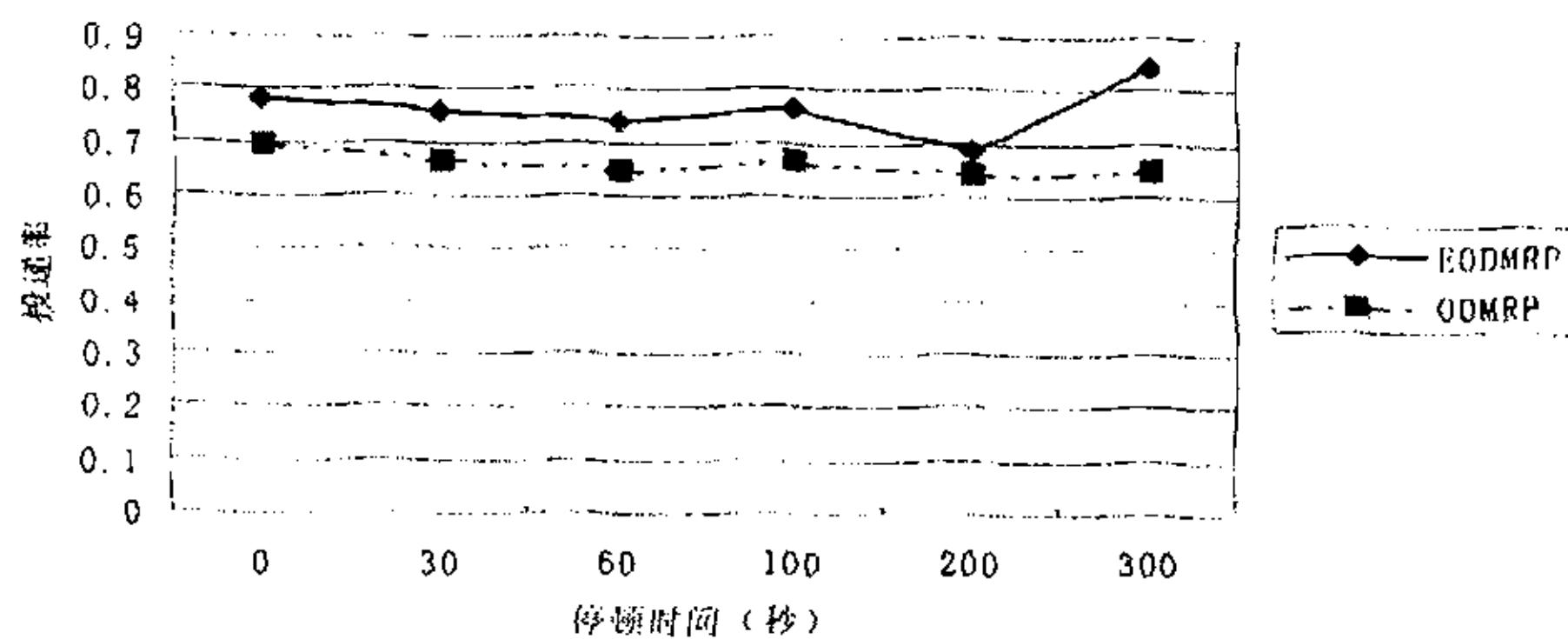


图 4-6: EODMRP 和 ODMRP 的包投递率
(50 个节点下, 1 个组播组, 1 个组播源, 每组 10 个接收者)

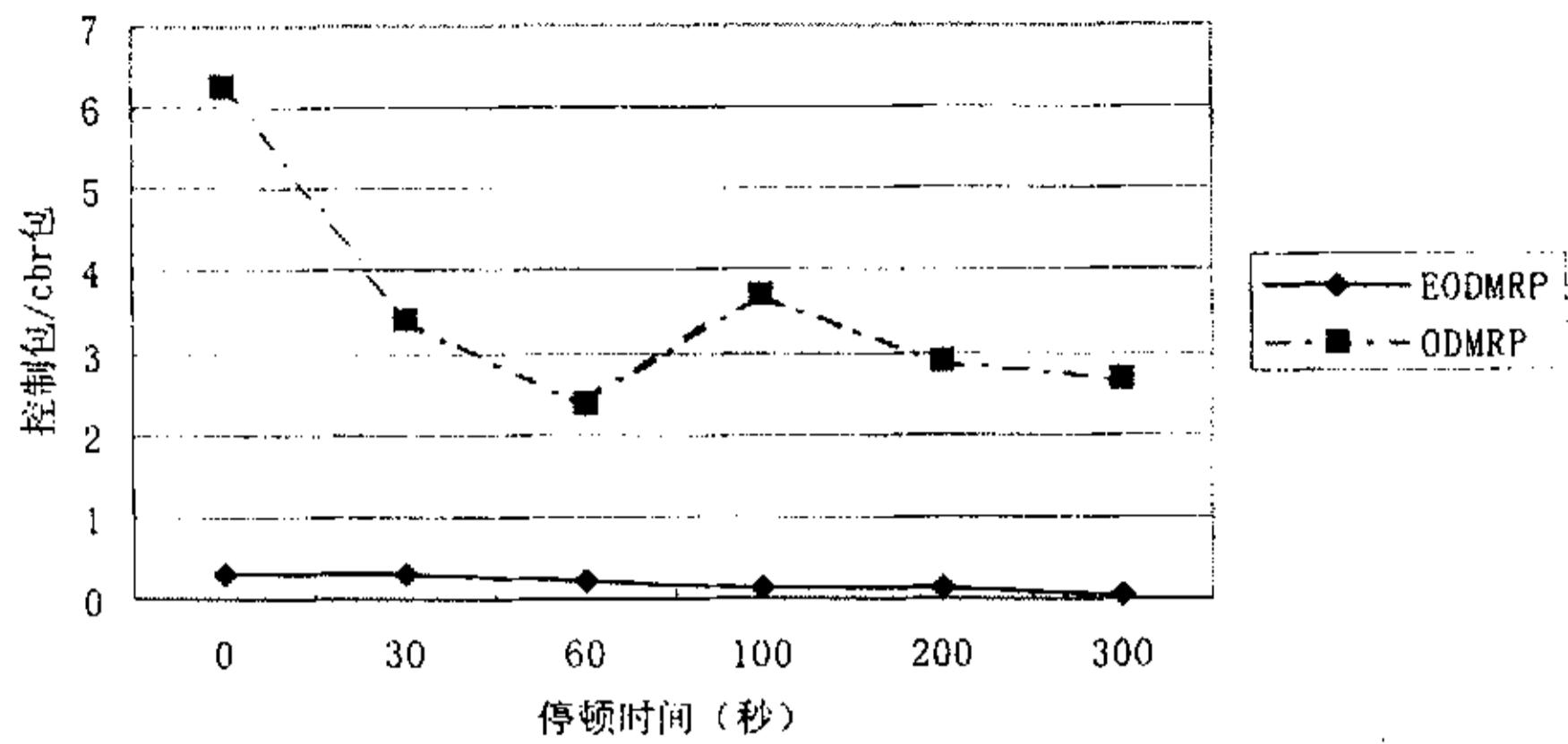


图 4-7: EODMRP 和 ODMRP 的控制开销
(50 个节点下, 1 个组播组, 1 个组播源, 每组 10 个接收者)

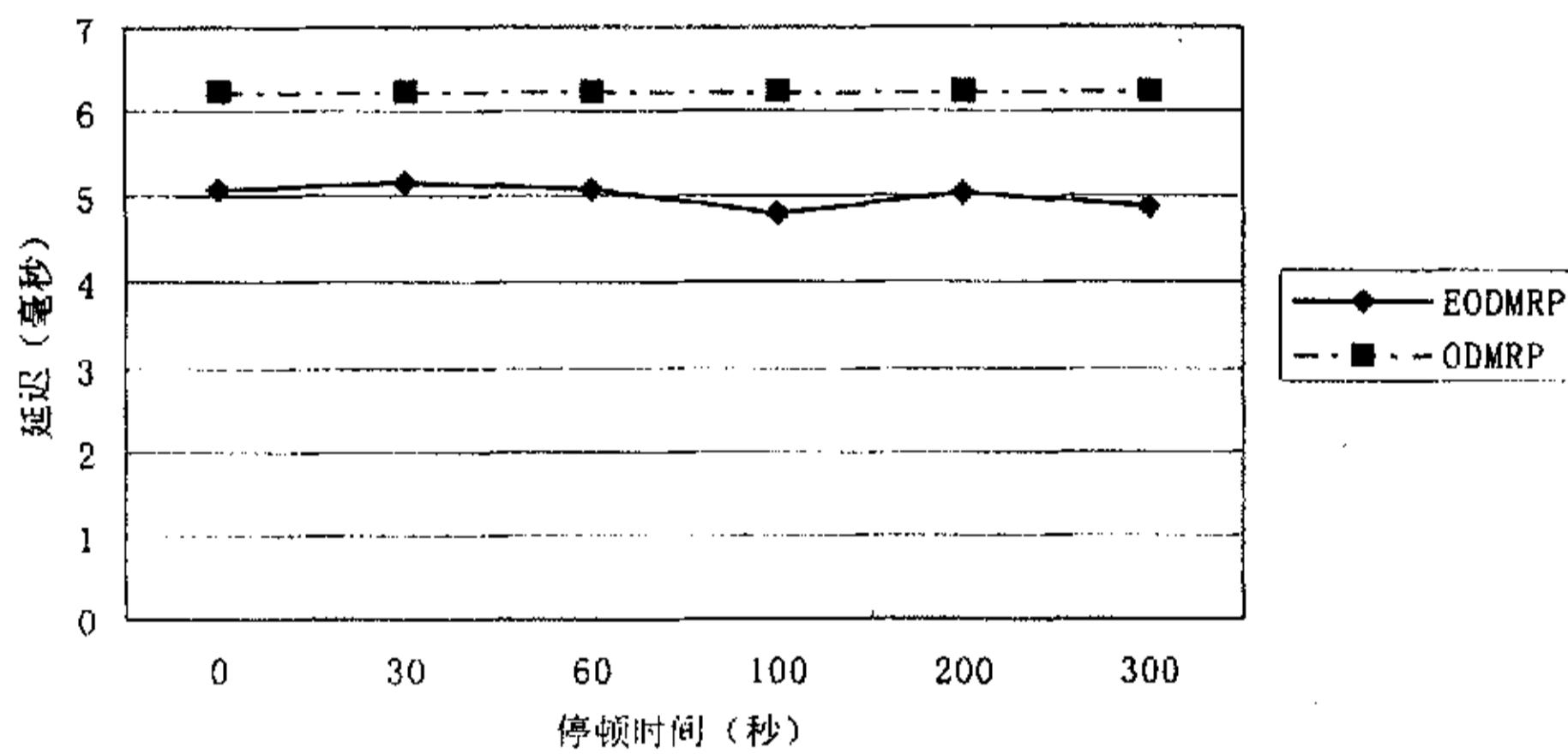


图 4-8: EODMRP 和 ODMRP 的控制开销
(50 个节点下, 1 个组播组, 1 个组播源, 每组 10 个接收者)

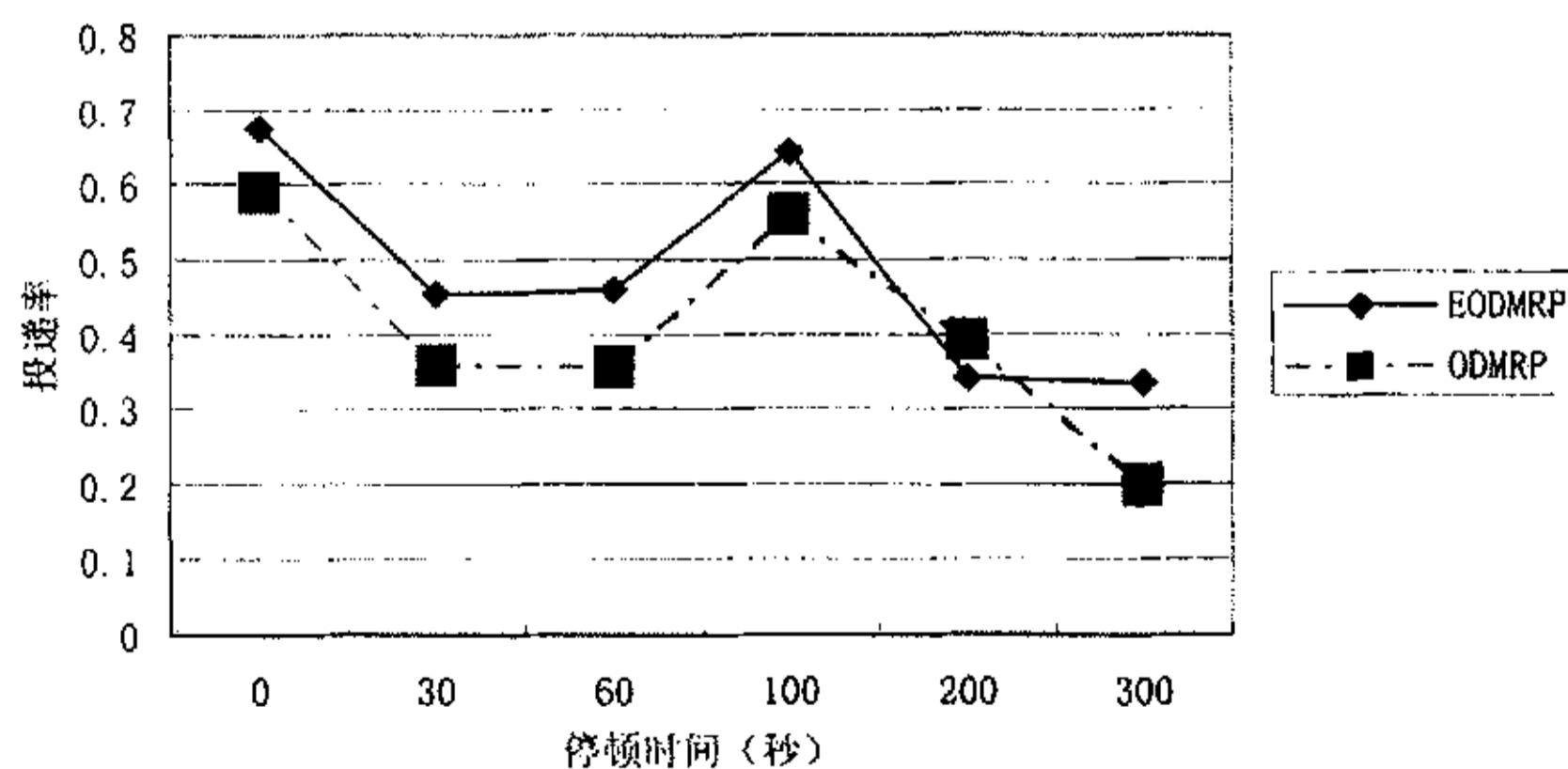


图 4-9: EODMRP 和 ODMRP 的投递率
(50 个节点下, 2 个组播组, 2 个组播源, 每组 15 个接收者)

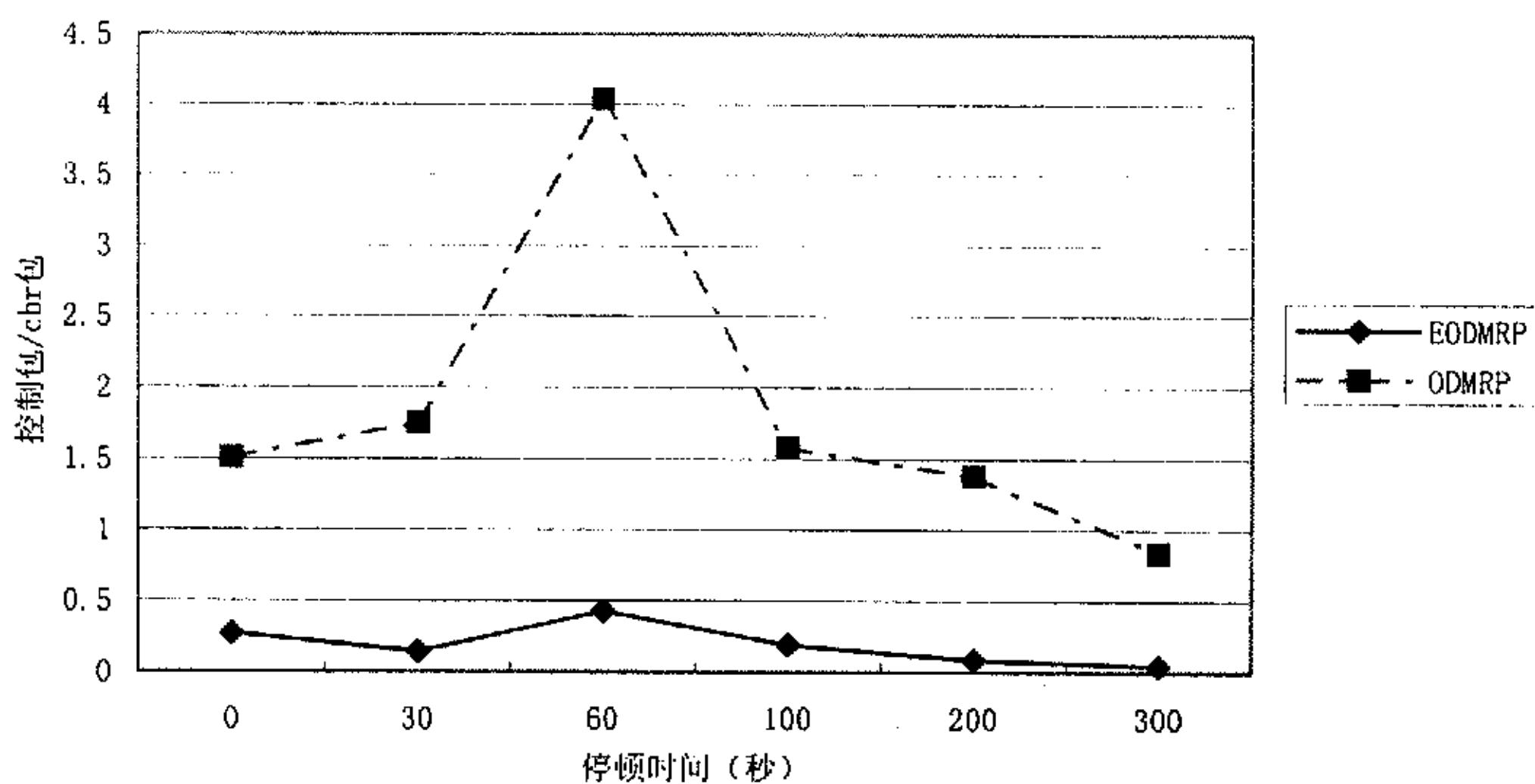


图 4-10: EODMRP 和 ODMRP 的控制开销
(50 个节点下, 2 个组播组, 2 个组播源, 每组 15 个接收者)

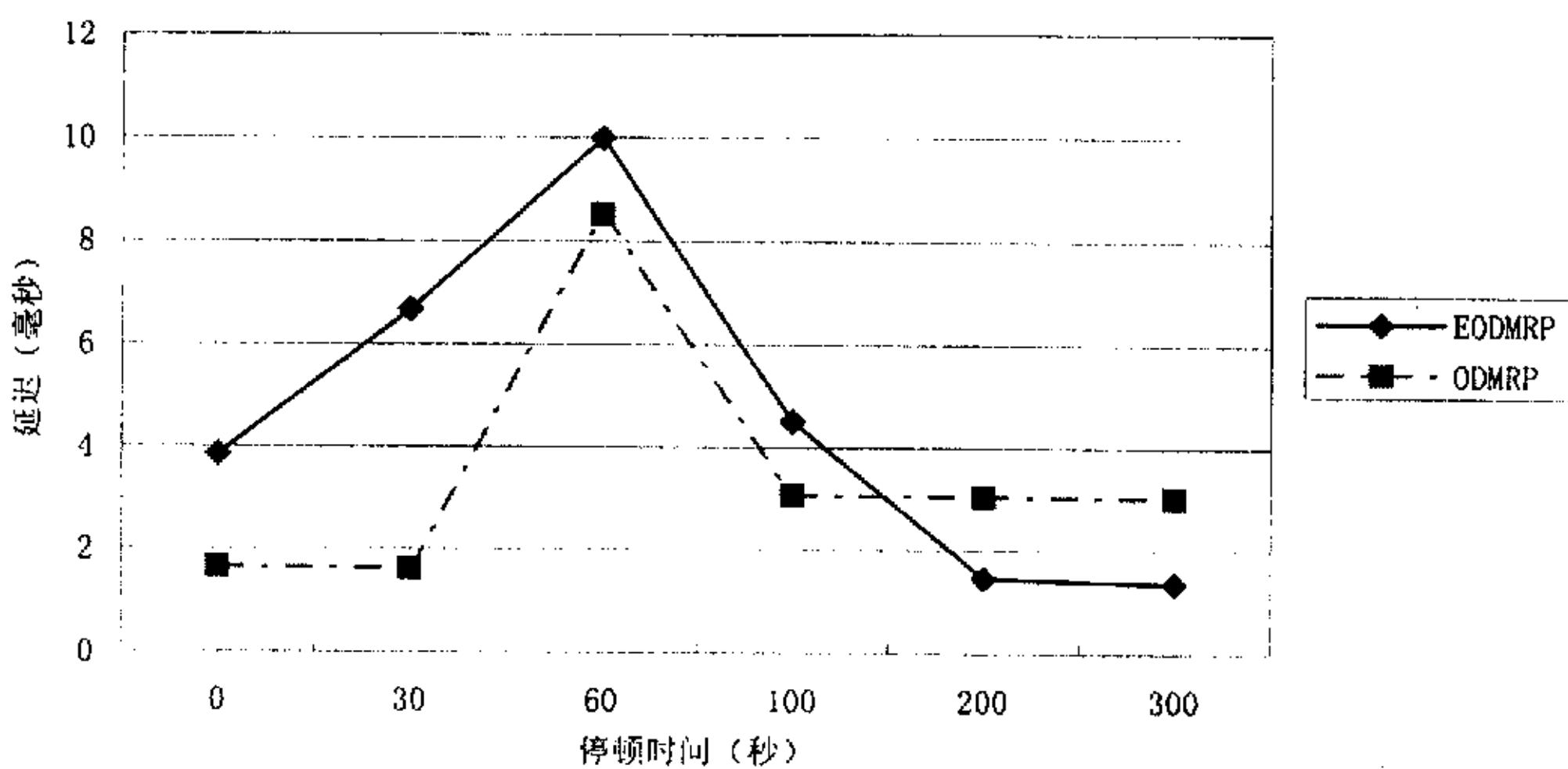


图 4-11: EODMRP 和 ODMRP 的传输延迟
(50 个节点下, 2 个组播组, 2 个组播源, 每组 15 个接收者)

4.4 对仿真中基于组播的网络延迟测试的讨论

数据包延迟对网络的整体性能有极大的影响，确定延迟的原因和位置等有利于设计、控制和管理网络。网络测试分析可分为内部法和外部法。内部法直接获取数据包延迟、丢失等信息。外部法通过获取端到端的数据包延迟、丢失等信息来确定网络内部的延迟、丢失、拥塞状况等。内部法有许多不足之处，容易加重网络负载、可扩展性不好、不同的网络产品难于协调一致等。而外部

法不需要网络内部各个结点协调一致，只需要边界端点处的延迟、丢失等信息。在使用外部法的测试与分析技术中，^[3,4]通过实验了解端到端的性能现象；^[1,2]基于单播的方法来测试与分析，扩展性不好，在八个结点的的网络上，最坏时带给网络的负载为 $O(N)$ ；虽然 mtrace 是一种基于组播的测试工具，最坏时带给网络的负载为 $O(N)$ ，但它要求所有的路由器对其提出的要求作出响应。

基于以上原因，Cascales, Presti F, Lo^[4,5]等人提出了端到端的基于组播的延迟、丢失分布推导分析法。它具有可扩展性好、不需网络内部结点上的信息等优点，但有两个不足之处：(1)未能解决怎样控制从源结点发送的测试数据包数的问题；(2)没有根据问题的具体特点设计出有效地降低时间复杂度的方法。

本文所用的测试方法中解决了该方法的以上两个不足之处。实验结果表明，本文提出的改进的测试分析法不仅能准确地控制测试数据包数，还使时间复杂度大为降低。从而既减轻了带给网络的负载，又使该测试分析法更高效适用。

4.4.1 模型及框架

按照^[4,5]的方法。由物理网络构造出逻辑组播树，使得除根结点和叶结点以外，每个结点至少有两个子结点，测试数据包从根结点以组播的方式发送出去。今 $r=(V, L)$ 表示此逻辑树， V 是所有结点的集合， L 是所有有向连接的集合，根结点记为 o ， $U=V\setminus\{o\}$ 为中间结点及目的结点的集合，结点 j 的所有儿子的集合记为 $d(j)$ ，其父结点记为 $f(j)$ 。如果 j 是 k 的子孙，则记为 $j < k$ ，“ $<$ ”是 V 上的一个偏序关系。根结点是源结点，所有叶结点只是目的结点。对于任意 $k \in V$ ， D_k 是取值在扩展的正实数集 $R' \cup \{\infty\}$ 上的随机变量，它表示数据包从 $f(k)$ 到 k 的时延， ∞ 表示数据包在此连接上丢失。 $D_o = 0$ 。假设 D_k 相互独立，记 $Y_k = \sum_{l \leq k} D_l$ ，它表示数据包从根结点到结点 k 的传输时延。把 D_k 的取值范围进行离散化，使其在 $(0, q, 2q, \dots, i_{\max}q, \infty)$ 上取值， Y_k 在 $\{0, q, 2q, \dots, i_{\max}q, \infty\}$ 上取值，其中 q 为离散尺寸， i_{\max} 为离散数。 D_k 的分布记为 A_k ， $\mu_k = 1 - a_k(0)$ 为连接 k 的利用率。 Y_k 的分布为 A_k ， $A_k(i) = \sum_{j=0}^i a_k(j) A_{f(k)}(i-j)$ 。

令 $A_0(0) = 1$ 。 $(r, k) = (V(k), L(k))$ 表示以第 k 个节点为根的子树， $R(k) = RI V(k)$ ， $\Omega_k(i)$ 表示随机事件 $\{\min_{j \in R(k)} Y_j \leq iq\}$ ，其概率为 $Y_k(i)$ 。 (r, a) 称为延迟树，如果 $\forall k \in U$ 均有 $a_k(0) > 0$ ，则称此延迟树为标准延迟树。

4.4.2 基于多播的延迟分布估计及性质

现在考虑从根结点以组播的方式发送 n 个数据包。本文的主要目的是仅由端到端的测试数据 $(Y_{k_n})_{k \in U, j=1, 2, \dots, n}$ 分析出网络内部各连接上的延迟特性。

定理 1^[3]: 令

$$\Delta = \{a = a_k(i) : a_k(0) > 0, \sum_{i=i_{\max}}^n a_k(i) \leq 1, k \in U, i \in \{0, 1, \dots, i_{\max}\}\},$$

$$\Theta = \{\gamma = \gamma_k(i) : \exists \sigma \in \Delta | \gamma = \Gamma(a), k \in U, i \in \{0, 1, \dots, i_{\max}\}\},$$

则 Γ 是从 Δ 到 Θ 的连续可微且有反函数的双射。以后用 $\hat{\gamma}$ 文来估计 γ , 其中产的表达式见式(8), 用 $\hat{\alpha}(i) = (\Gamma(\gamma))_k(i)$ 来估计 $\alpha_k(i)$ 。

α 的表达式. 当 $i=0$ 时, $A_k(0)$ 由式(4-1)解出。

$$(1 - \gamma_k(0)/A_k(0)) = \prod_{d \in d(k)} (1 - \gamma_d(0)), k \in U \setminus R \text{ 时}; \quad \gamma_k(0) = A_k(0), \quad k \in R. \quad (4-1)$$

$$\beta_k(0) = \gamma_k(0)/A_{f(k)}(0), k \in U. \quad (4-2)$$

当 $i > 0$ 时, $A_k(i)$ 式由式 (4-3) 和式 (4-4) 解出。

$$\gamma_k(i) + A_k(0) \left\{ \prod_{d \in d(k)} [1 - [\gamma_d(i) - \sum_{j=1}^{i-1} \beta_d(i-j) A_k(j) - \beta_d(0) A_k(i)]/A_k(0)] - 1 \right\} + \sum_{j=1}^{i-1} A_k(j) \left\{ \prod_{d \in d(k)} [1 - \beta_d(i-j)] - 1 \right\} + A_k(i) \left\{ \prod_{d \in d(k)} [1 - \beta_d(0) - 1] \right\} = 0, k \in U \setminus R \quad (4-3)$$

$$A_k(i) = \gamma_k(i) - \sum_{j=0}^{i-1} A_k(j), k \in R \quad (4-4)$$

$$\beta_k(i) = \{\gamma_k(i) - \sum_{j=0}^{i-1} A_{f(k)}(j) \beta_k(i-j)\}/A_{f(k)}(0), k \in U \quad (4-5)$$

把所有的 $A_k(i)$ 都解出来以后, $\alpha_k(i)$ 即可求得

$$\alpha_k(i) = \begin{cases} A_k(0)/A_{f(k)}(0), & i = 0 \\ [A_k(i) - \sum_{j=0}^{i-1} A_{f(k)}(j) \alpha_k(i-j)]/A_{f(k)}(0), & i = 1, 2, \dots, i_{\max} \end{cases} \quad (4-6)$$

4.4.3 延迟分布测试过程

整个测试分析过程如下:

```

PROC main () {
    For (i=0,1,...,imax)
        发送 n0 个测试数据包;
        infer_delay(1,i);
        发送 m 个测试数据包,
        计算前后两组 γ 的值;
    While (前后两组 γ 差的绝对值 > σ)
        {
            发送 m 个测试数据包;
            计算新的 γ ;
        }
        Find_y(1);
    PROC find_y(k)
        {for (j ∈ d(k)){
            Yj = find_y(i);
            for(m=1,2,...,n)
                Yk[m] = min(Yk[m], Yj[m]);
        }
        For (i=0,1,...,imax)
    }
}

```

```

 $\hat{\gamma}_k(i) = n^{-1} \sum_{m=1}^n l_{\{\hat{Y}_{k,m} \leq iq+q/2\}};$  Newton 法计算出  $X$ , 用 Horner 法计
return( $\hat{Y}_k$ );
}

PROC interface_delay(k,i){
    if (i == 0)  $A_k(i)$  = solveAK1(k);
    else  $A_k(i)$  = solveAK2(k,i);
 $\beta_k(i) = \{\gamma_k(i) - \sum_{j=0}^{i-1} A_{f(k)}(j)\beta_k(i-j)\} / A_{f(k)}(0);$  算多项式的值;
 $\alpha_k[i] = \{A_k[i] - \sum_{j=1}^i A_{f(k)}[j]\alpha_k[i-j]\} / A_{f(k)}(0);$   $\hat{A}_k[0] = \hat{\gamma}_k[0] / X;$ 
for( $j \in d(k)$ )
    infer_delay(j,i);
}

PROC solveAK1(k){
     $X = \hat{\gamma}_k[0] / \hat{A}_k[0];$  return ( $\hat{A}_k[0]$ );
    由式 (1) 以  $X_0 = 1$  为初始值, 用
}

```

PROC solveAK2(k,i){
 {
 以 $X^{(0)} = 1$ 为初始点, 用变尺度
 法求出 $f'(X)$ 的第一个极小值点
 X_0 , 用 Horner 法计算多项式的值;
 由式 (4-4) 以 X_0 为初始点用
 Newton 法计算出 $\hat{A}_k[i]$, 用 Horner
 法计算多项式的值;
 return ($\hat{A}_k[i]$);
 }
}

4.5 结论

本章提出了提高 ODMRP 性能的方法, 通过使用移动和链路状态预测, 路由和转发组可以在拓扑改变前重建。对刷新时间的选择避免了不必要的控制包的传输引起的带宽浪费。本文使用的新的路由选择算法能够使路由在最长时间内保持有效。稳定路由的使用进一步减少了控制包引起的负载。被动应答和重传机制被用于提高应答传输的可靠性。另外, 本文还提出了减少获得路由延时的方法。

仿真结果显示这几种思想都不同程度的提高了基本 ODMRP 的性能。低速链路中, 发往信宿的数据包越多, 产生的控制信息相对越少。在高速链路中, 控制包的利用率大大提高, 减少了端端延时。因此, 改进后的 ODMRP 对于移动网络有着更好的适应性。

第五章 Ad hoc 网络的 QoS 保证

5.1 Ad hoc 网络中的 QoS 保证策略

随着 Internet 技术的快速发展，人们对于 best effort 的服务不再满足，如何获得更多的带宽，如何减少错误，如何降低延迟及其抖动，使得 QoS 相关的研究，包括 Integrated Service (IntServ/RSVP)^[30]、Differentiated Service^[25]以及 MPLS^[26]等，成为重要的网络研究课题。目前的 Ad hoc 路由协议中，多数都是在近几年间提出的，只针对基本的 best effort 业务，并没有考虑到 QoS；另外，关于提供 QoS 保证机制的研究也较少。本章旨在对当前有线网络中保证 QoS 机制分析的基础上，提出适合于 Ad hoc 网络的 QoS 保证机制，其中包括信令机制和 QoS 路由协议。

5.1.1 IP 网络中 QoS 保证的几种机制

IETF 已经建议了很多服务模型和机制，以满足 QoS 的需求。其中主要有：综合业务 (IntServ/RSVP) 模型^[30]，区分型 (DiffServ) 业务模型^[25]协议标记交换 (MPLS)^[26]，QoS 路由^[27]等。IntServ 的特点是资源预留，实时应用在传输数据前必须首先建立通道和预留资源；RSVP 是其中用来建立通道和预留资源的协议。在 DiffServ 中，把分组加以标记，产生不同的级别，每个级别的分组得到不同级别的服务。MPLS 是一种面向连接的前向转发策略，在进入 MPLS 作用域时给分组赋予一定的标签，随后分组的分类、转发和服务都将基于标签完成。QoS 路由在路由寻径时会加入一定的 QoS 约束，如带宽或延迟的要求，以满足一定的 QoS 需要。

5.1.2 Ad hoc 网络 QoS 策略研究

随着 Ad hoc 网络的不断发展，在 Ad hoc 网络传送不同类型业务的需求与日俱增，并且在某些实时应用的场合（如传送语音、视频图像等信息），不但需要正确的路由信息，还需要提供有一定保障的 QoS。这些要求对带宽较低、通信线路质量都相对受限的 Ad hoc 网络而言，显得有些苛刻。所以，如何在 Ad hoc 网络中合理地利用无线网络资源，实现有针对性的 QoS 保证具有现实意义。本节中，我们针对 Ad hoc

网络的具体特点，简单分析现有各种 QoS 机制在 Ad hoc 网络中的适用情况。

5.1.2.1 Ad hoc 网络中支持 QoS 的难点

与单跳的蜂窝网络不同，Ad hoc 网络中无线节点之间通过多跳的无线连接进行通信。所以在 Ad hoc 网络中，不仅要考虑单跳情况下的 QoS 保证，还要保证无线多跳路径上的 QoS。另外，由于 Ad hoc 网络自身的一些特点，已有的一些固定网络中支持 QoS 的方法不能直接移植到 Ad hoc 网络，而必须加以适当的改造才能应用。综上所述，Ad hoc 网络中支持 QoS 主要包括以下难点：

- **Ad hoc** 网络的网络资源极其有限

Ad hoc 网络的特点是信道带宽较低，并且无线信道质量不稳定。虽然可以通过改进编码方法，增加信号发射功率以及提供冗余路径等方式来提高信道质量，然而更强的编码会导致带宽减少；提供冗余路径会加重网络的负担；增大功率则会增加分组冲突的概率。所以，信道质量问题的解决通常会引起网络进一步拥塞；

- **Ad hoc** 网络中的信道访问存在竞争

无线信道是广播共享形式的，多数 Ad hoc 网络采用随机的信道访问机制，如载波检测多址（CSMA），同时 Ad hoc 网络中存在终端隐藏和终端暴露的问题^[36]。因此，引入大量控制分组的方法不再适用，因为那样会带来更多的冲突，降低系统的整体性能。

- **Ad hoc** 网络拓扑结构动态变化

要减轻网络拓扑结构变化对 QoS 的影响需要改进媒体访问控制（MAC）层协议以及依靠能够快速生成新路径的路由协议。

5.1.2.2 Ad hoc 网络 QoS 策略讨论

1. IntServ 与 Ad hoc 网络

IntServ 的概念是源于电信网和 B-ISDN，亦即本质上是选择虚电路的机制，如 RSVP 就是一种信令机制负责建立和维护链路的连接，由路由器进行相应的资源预留，如分类排队等，以支持每一连接的 QoS 要求。在此机制的基础上，IntServ 提供基于每个流的定量的 QoS。在 Ad hoc 网络中应用 IntServ 方法的优缺点讨论如下：

可扩展性：IntServ 提供逐个流的控制，因此，在连接的每一个节

点中，每一个流的状态都必须保存，这会造成中间节点路由器的存储和计算开销的增加，这就是众所周知的 IntServ 的可扩展性问题。然而，Ad hoc 网络是规模较小的网络，目前只用作端网络（stub network），因此可扩展性问题在 Ad hoc 网络中不突出。

信令：信令协议一般包括三部分：建立连接，撤销连接和维护连接。Corson^[37]指出在网络能力受限的网络中，如无线网络，信令等控制开销所占比重会很大。因此，对于 Ad hoc 网络来说，用于维护连接的开销甚至会超过建立连接所需的开销，所以类似 RSVP 之类的信令系统不适用于无线网络。Ad hoc 需要的是轻便的（lightweight）信令系统。

路由机制：IntServ 对路由器提出了很高的要求，路由器一般应包括以下四种机制：分类程序（classifier）、分组调度程序、RSVP 和准入控制机制。这样，中间各个路由器的处理开销会很大。一种观点认为对于电源受约束的 Ad hoc 节点，IntServ 处理开销大，因而耗电量大，不适用。但是，本文作者认为对 Ad hoc 网络而言，带宽能力不足是主要矛盾，因此，应当充分挖掘 Ad hoc 中主机的处理计算能力，以补偿 Ad hoc 网络带宽的能力限制。

2. DiffServ 与 Ad hoc 网络

DiffServ 的产生是为了克服在 Internet 主干网上使用 RSVP 和 IntServ 的困难，降低每个节点（路由器）处理的复杂程度，从而增强可扩展性。DiffServ (DS) 通过一种优先级机制来保证 QoS，这种 QoS 同 IntServ 的 QoS 相比要松散一些，常常是聚集的 QoS。在 Ad hoc 网络中应用 DiffServ 方法的优缺点讨论如下：

Ad hoc 网络中每一个节点既是主机又是路由器，从这一意义上讲同 DiffServ 是类似的。但是，DiffServ 需要建立确定的 DS 域，在 DS 域的边界节点（boundary node）需要实现复杂的调节（conditioning）功能，具体包括分类（classification）、标记（marking）和整形（shaping）等操作。在此基础上，核心路由器按照 PHB (Per-Hop Behavior) 进行分类转发。但是，这种模式在 Ad hoc 网络中有很大的局限性：(1) 确定 DiffServ 的域边界很困难：每个节点都在移动，由谁来作为边界路由器？(2) 边界路由器的通信失败会对网络造成很大的影响，降低了网络的可靠性。

另外，从应用的对象来看，DiffServ 是为骨干网或传输网（transit network）设计的，而 Ad hoc 是一种端网络，两种网络有着很大的差别。

所以，DiffServ 不适用于 Ad hoc 网络。

3. Ad hoc 网络 QoS 策略的选择

实现 Ad hoc 中的 QoS 保证，首先应当选择一种 QoS 保证机制。在以上各小节讨论的基础上，研究在 Ad hoc 网络中策略的选择。

RSVP 为每个单独的流分配端到端的网络资源，它能提供具有绝对保证的 QoS，但是由于可扩展性等原因而没有在 Internet 上得到广泛的应用。在 Ad hoc 网络中，带宽是受限的，并且节点处理的数据流量较小，因此在 Ad hoc 网络中采用资源预留协议是一种较为合理的方法。但是由于标准的资源预留协议引入了过多的控制分组，会给 Ad hoc 网络带来过大的传输开销，所以应对其进行适当简化，以适合 Ad hoc 网络的特定环境。为了支持资源预留协议很好的工作，应该选择 QoS 路由协议为预留协议寻找满足 QoS 需求的路由。

Ad hoc 网络中由于其固有的移动性，有可能经过很短的一段时间，当初的预留就不能再被保证了。因此，Ad hoc 网络中要实现 QoS，必须是快速自适应的。一种可行的方法是在 Ad hoc 网络中定义一种带内信令协议，它可以通过在数据分组中携带控制信息的方法来传递信令，而不引入独立的控制分组，采用这种方法既不会给网络带来很大的额外负担，又能适应网络拓扑结构的动态变化，使得数据流连接能够在 Ad hoc 网络环境下快速地建立、恢复和释放。

Ad hoc 网络这样的自治系统主要作为端网络组网，所以，可扩展性不是主要问题。即使将来随着应用的发展，容量的增大，可扩展性问题凸现时，仍可以通过其他方式解决，如引入代理、接入点或者边界点的处理方式。因此，可以选择 per-flow 的控制方式，以提高 QoS 保证的效果。

以下简单总结一下本文所选择的 Ad hoc 网络的保证策略，那就是首先选择 QoS 路由，在此基础上引入 IntServ 机制中的 per-flow 的控制，信令则采用带内信令系统。需要说明，由于无线及移动的特殊性，Ad hoc 网络中不可能实现真正的 hard-guaranteed 服务，受环境的限制，只能实现 soft-guaranteed^[28] 服务。

5.2 结论

本章讨论了 Ad hoc 网络中的 QoS 的策略问题，并在对几种常见的 Internet QoS 机制讨论的基础上，分析了各种 QoS 机制在 Ad hoc 网络中

的适用情况。作者认为，将带内信令系统同QoS路由结合进行per-flow的控制是在Ad hoc网络中实现QoS的一种很有前景的方法。

第六章 结论与展望

随着无线技术的发展, ad hoc 无线网络技术越来越显示出其巨大潜力, 同时, 组播和无线技术的结合更加为无固定中心设备的应用场景提供了很好的支持。

目前组播环境下的无线网络设计和实施的一个重点是提高网络中的路由效率, 以有效的利用无线网络中有限的资源。本文研究了ODMRP 协议在组播环境下的工作情况, 并在此基础上提出了能更好的适用于组播环境的 EODMRP 协议。

EODMRP 协议中, 我们将 ODMRP 中的 mesh 结构改变为组播树结构。只要网络中有至少一个发送者和接收者, 就将建立对应于某一特定源的组播转发树。EODMRP 监控组播源的业务模式, 并据此来发现组播树中的链路故障。组播源暂停数据发送时, 为了启动对组播树的链路故障监控, EODMRP 以递增的间隔时间发送有限个 keep alive 信息, 当组播源在一定时间内不再发送数据, 那么 keep alive 信息也被终止, 整个组播树也随之终止。组播源发送模式的严重改变意味着组播源在一段时间内不工作, 此时如果继续维持网络路由将成为一种浪费。EODMRP 还能够自动对无用的组播树的枝叶部分进行修剪, 修剪与否不依靠显式的修剪信息, 而根据是否收到来自下游结点的被动应答。

EODMRP 还会以全网广播的方式非周期的发送组播数据包, 这种广播不具备核心路由的功能, 不是协议机制所必需的, 当这种非周期机制对于提高路由收敛效率有很好的作用。

本文所提出的 EODMRP 还通过 Probing 机制提高了系统的寻径能力, 系统的容错能力以及对路由差错的及时修复能力, 作者认为 EODMRP 时非常适合于频繁移动的 ad hoc 网络的一种路由方式。

仿真实验结果表明, ad hoc 网络中, EODMRP 在投递率, 端端延迟和负载开销方面都有了很大的提高, 尤其是网络拓扑频繁变化时, 其性能提高更为明显, 因此 EODMRP 协议是适合于 ad hoc 组播网络的路由协议, 有着很好的应用前景。

EODMRP 的局限性在于对组播树的依赖, 问题的解决可能不仅仅

要依赖于路由层，而是更可能要同路由层以下的协议更好的协同工作，如怎样将路由协议同 MAC 层更好的结合起来，这是一个研究的方向。组播路由与传输层协议的协同工作也是需要研究的问题，IP 组播使用 UDP，任何主机都可以向某个组播地址发送 UDP 包，并且底层组播机构将传送这些 UDP 包到所有组成员。这使得组播安全性问题和组播的可靠性问题一样难以解决。组播 ad hoc 网络路由的缺点归结为一点就是其控制的复杂性，但是相信其巨大挑战的背后隐藏着的是巨大的资源空间。

参考文献

- [1] D.Towsley. MINC: multicast inference of network characteristics project.1999,<http://gaia.cs.umass.edu/minc>.
- [2] S.Casner, The mtrace (8) manual page,1998,
<http://ftp.parc.xerox.com/pub/net-research/ipmulti/>.
- [3] R.Cascares, N.G.Duffield, J.Horowitz . Multicast-based inference of network-internal loss characteristics. Information Theory, IEEE Transactions on.1999. vol.45(7):2462~2480.
- [4] N.G.Duffield, J.Horowitz, F.L.Presti . Multicast topology inference from measured end-to-end loss. IEEE Transactions on Information Theory.2002. vol.48(1):26~45.
- [5] S.E Deering.Multicast Routing in Internetworks and Extended LANs. ACM SIGCOMM Computer Communication Review, 1988, vol.8(4):55~64.
- [6] J.Moy, Multicast Routing Extensions for OSPF, Communications of the ACM, 1994, vol.37(8):61~66.
- [7] T.Ballardie, P.Francis, J.Crowcroft. Core Based Trees (CBT) . In: Conference proceedings on Cmmunications architectures,protocols and applications. San Frocisco, CA:ACM press,1993.85~95.
- [8] S.Deering, D.L.Elstrin. The PIM architecture for wide-Area multicast routing, IEEE/ACM Transactions on Networking, 1996, vol.4(2):153~162 .
- [9] J.B.David A.Maltz, J.Jetcheva. A performance comparison of multi-hop wireless ad Hoc network routing protocols. In: Proceedings of the 4th annual ACM/IEEE International Conference on Mobile Computing and Networking.Dallas,USA:ACM Press, 1998. 85~97.
- [10] S.R. Das, R.Castaneda, J.Yan. Comparative performance evaluation of routing protocols for mobile, ad hoc networks. In:Preceedings of 7th Internatioal Conference. Lafayette,LA,USA,1998.153~161.
- [11] P.Johannson, T.Larsson. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking.Seattle,WA,USA:ACM Press,1999. 195~206.
- [12] E.Bommaish.AMRoute:Ad hoc multicast routing protocol, Internet -Draft, draft-talpade-manet-amroute-00.txt, 1998, Work in progress.

- [13] C.W.Wu, Y.C.Tay, Ad hoc multicast routing protocol utilizing Increasing id-numbers (AMRIS) Functional Specification, Internet -Draft, draft-ietf-manet-amris-spec-00.txt, 1998, Work in progress.
- [14] J.J.Garcia-Luna-Aceves, E.L.Mardruga.The core-assisted mesh protocol. IEEE Journal on Selected Areas in Communications, 1999,vol.17(8):1380~1394.
- [15] S.J.Lee. Routing and multicasting strategies in wireless mobile ad hoc networks:[A dissertation for the doctor's degree].USA:UCLA, 2001.
- [16] S.J.Lee, W.Su, J.Hsu. A performance comparison study of ad hoc wireless multicast protocols.In:Proceedings of the nineteenth Annual Joint Conference if IEEE Computer and Communications Societies. Israel, 2000.555~574.
- [17] T.Pagtzis, P.Kirstein, S.Hailes. Operational and Fairness issues with connection-less traffic over IEEE802.11b. In: Proceedings of IEEE International Conference on Communications 2001.Helsinki,Finland, 2001.1905~1913.
- [18] 王雷, 无线移动 ad hoc 网络多径路由研究: [博士学位论文].天津: 天津大学,2001.
- [19] 梁冰, MANET 实验床的设计与实现: [硕士学位论文].天津: 天津大学,2002
- [20] S.J.Lee, M.Gerla, C.K.Toh, A simulation study of table-driven and on-demand routing protocols for mobile ad-hoc networks. IEEE Network, 1999,vol.13(4):48~54.
- [21] C.Huitema, Project felix: Independent monitoring for network survivability.1999, <ftp://ftp.bellcore.com/pub/mwg/flex/>.
- [22] B. Welch. Practical programming in Tcl and Tk, Prentice Hall,1980.
- [23] J.Meggers. Multicast communication in ad hoc networks. In: Proceedings of 48th IEEE Vehicular Technology Conference. Ottawa, Ont., Canada,1998.372~376.
- [24] D.B. Johnson, D.A.Maltz. Dynamic source routing in ad-hoc wireless networks. Mobile Computing, 1996,vol4(3):153~181.
- [25] S.Blake, D.Black, M.Carlson. An architecture for differentiated services, RFC 2475, 1998.
- [26] E.Rosen, A.Viswanathan, R.Callon. Multiprotocol label switching architecture, RFC 3031, 2001.
- [27] S.Crawley, R.Nair, B.Rajagopalan. A framework for QoS-based routing in the Internet, RFC2386, 1998.

- [28] S.G.Chen, K.Nahrstedt. An overview of quality of service routing for next-generation high-speed networks: problems and solutions. IEEE Network, 1998,vol12(6):64~79.
- [29] S.J.Lee, W.Su,M.Gerla. The design,implementation, and performance evaluation of the on-demand multicast routing protocol in multihop wireless networks. IEEE Network,2000, vol.14(1): 70~77.
- [30] R.Braden, L.Zhang, S.Berson, Resource reservation protocol (RSVP), RFC 2205, 1997.
- [31] K.Fall, K.Varadhan. Ns notes and documentation, 1999, <http://www.mash.cs.berkeley.edu/ns/>.
- [32] F.A.Tobagi, L.Kleinrock. Packet switching in radio channels: Part II-the hidden terminal problem in carrier sense multiple-access modes and the busy-tone solution. IEEE Transactions on Communications. 1975,vol. 23(12): 1417~1433.
- [33] L.Ji, M.S.Corson. A lightweight adaptive multicast algorithm, In: Proceeding of IEEE Global Telecommunications Conference. Sydney, NSW,Australia,1998.1036~1042.
- [34] W.Richard Stevens, UNIX Network Programming volume 1 Networking APIs: Sockets and XTI, Prentice-Hall International, Inc. 1997.

作者攻读硕士学位期间发表的论文和参加科研的情况

发表论文

1. Chen Ling,Lu Kaining, Jin Zhigang, Enhancement of the On-demand Multicast Routing Protocol in the Ad hoc Network, Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering CCECE 2002,1511~1515.
2. 鹿凯宁，陈凌，天津市教育科研宽带骨干网的规划和建设，中国教育网第九界学术年会论文。
3. 鹿凯宁，陈凌，PATHCHAR及其在网络带宽测量中的应用，中国教育网第九界学术年会论文。
4. 陈凌，鹿凯宁，基于负载平衡的无线ad-hoc网络路由，2002亚太高速网络会议论文。
5. Liang Bing,Chen ling, Jin zhigang, Lu Kaining, Improvement and Experiments of an On-demand multicast Routing Protocol in an Ad hoc Network Testbed,Proceedings of future telecommunication conference 2001,479~483.

参与的课题

与美国 cisco 公司合作项目，“cisco 公司-天津大学-渥太华大学无线通信实验室”。项目还得到国家自然科学基金和天津市政府信息办的支持。

致 谢

本论文是在导师鹿凯宁教授的亲切关和悉心指导下完成的，两年多来导师对我学习工作上的循循善诱和谆谆教诲，以及在生活上对我的关怀，使我终生难忘。导师严以律己、宽以待人的思想品格，对科学事业勤勤恳恳、呕心沥血的科学态度和严谨的治学作风无不让我产生深深的敬意，也每每成为我前行的动力。在此谨致以崇高的敬意和诚挚的感谢。

论文的顺利开展很多得益于金志刚博士的帮助和指导，同金博士的很多讨论使得对问题的理解更加深刻，在此表示衷心的感谢。论文完成过程中还得到了硕士生周慧勇、涂翔云、及小梅、陈泽华和杜蓓蓓的协助以及网络中心的杨晋生、陈慰国等诸位老师的指导以及胡宇明、胡晔、刘青竹等工作人员的大力配合，在此一并致谢。

同时衷心感谢参加本论文评阅和答辩工作的专家们。

最后感谢我的家人多年来对我的关心、支持和鼓励，同时衷心感谢所有关心、支持和帮助过我的老师、同学和朋友们。