



高新科技译丛
通信技术系列

Mobile Intelligence

移动智能

【加】 Laurence T. Yang 等编著 卓力 张菁 李晓光 张新峰 译

WILEY | 国防工业出版社
National Defense Industry Press

高新科技译丛

装备科技译著出版基金资助出版

移动智能

Mobile Intelligence

[加] Laurence T. Yang [澳] Agustinus Borgy Waluyo 编著
[日] Jianhua Ma [澳] Ling Tan [澳] Bala Srinivasan

卓力 张菁 李晓光 张新峰 译

国防工业出版社

·北京·

著作权合同登记 图字：军-2012-091 号

图书在版编目（CIP）数据

移动智能/（加）杨天若等编著；卓力等译. —北京：国防工业出版社，2014.1
（高新科技译丛）

书名原文：Mobile Intelligence

ISBN 978-7-118-08968-4

I. ①移… II. ①杨… ②卓… III. ①移动通信—智能技术 IV. ①TN929.5

中国版本图书馆 CIP 数据核字（2013）第 248157 号

This translation published under John Wiley & Sons. No part of this book may be reproduced in any form without the written permission of the original copyrights holder.

Mobile Intelligence

ISBN 978-0-470-19555-0

Copyright © 2010 by John Wiley & Sons, Inc.

Copies of this book sold without a Wiley sticker on the cover are unauthorized and illegal.

All rights reserved.

本书简体中文版由 John Wiley & Sons, Inc. 授权国防工业出版社独家出版。
版权所有，侵权必究。

国防工业出版社出版发行

（北京市海淀区紫竹院南路 23 号 邮政编码 100048）

北京奥鑫印刷厂印刷

新华书店经售



*

开本 787×1092 1/16 印张 25¼ 字数 481 千字

2014 年 1 月第 1 版第 1 次印刷 印数 1—2000 册 定价 98.00 元

（本书如有印装错误，我社负责调换）

国防书店：（010）88540777

发行邮购：（010）88540776

发行传真：（010）88540755

发行业务：（010）88540717

前 言

如今,随时随地使用手机等移动通信设备的人们随处可见。人们通过个人数字助理设备(PDAs)和具有无线连接功能的笔记本计算机就能连接网络获取信息。这种称为移动计算的应用正在全球范围内迅速增加。各大移动手机生产商竞相发布各自产品的最新版本,就在不久之前苹果公司推出了其有史以来第一款具有革命性的移动手机,即众多用户期盼已久的 iPhone。这对于必须紧跟信息服务以获得商业优势的移动服务提供商来说是一个巨大的挑战。

无线技术的快速发展造就了移动计算模式。它引入了随时随地的计算概念,由此诞生的应用具有广阔前景。但是在移动计算真正实现其潜力之前,仍有一些重要的问题需要解决,比如存储容量受限,断线频率,带宽窄,安全问题,非对称通信费用和带宽以及小屏幕尺寸等。计算智能是一种在研究中用于分析各种问题的强大、不可或缺的方法,它通过学习、自适应或演化计算来构建方案,在某种意义上来说可称为智能方案。这种方法已经用于解决各种不同领域的问题,如安全、市场营销、图像质量预测以及风险评估等。因此,利用它来解决移动计算的问题也颇具潜力。

将计算智能方法应用到移动模式具有广阔的应用前景,二者的结合产生了一个新的概念,也就是所谓的移动智能。移动智能可以使许多应用获益,包括数据库的查询处理、多媒体、网络安全、信息检索、电子商务系统、网络流量与应用以及搜索引擎等。

本书涵盖了移动模式下计算智能各种应用的最新研究进展,包括移动数据智能、移动挖掘、移动智能安全、移动代理、基于位置的移动信息服务、移动环境感知和应用、智能网络以及移动多媒体等。本书将致力于传播并推进移动智能的最新研究。

本书的章节选自移动智能领域杰出学者的特约稿件。全书共划分为 7 个部分,分别是:移动数据及人工智能(第 1~6 章),基于位置的移动信息服务(第 7~10 章),移动挖掘(第 11、12 章),移动环境感知和应用(第 13~15 章),移动智能安全(第 16~19 章),移动多媒体(第 20~23 章)和智能网络(第 24~27 章)。

第 1 部分(移动数据及人工智能)由 6 章构成,每章从不同方面关注无线环境下对移动数据的高效检索。第 1 章是对现有移动 Ad Hoc 网络智能路由协议的综述,对几个关键问题进行了评述,同时也讨论了这些协议如何就数据包吞吐量和延时性能等方面进行评价。该综述是学习移动 Ad Hoc 网络(MANET)无线路由协议的一个非常好的起点。第 2 章提出了 MANET 连通域集(CDS)结构的拓扑管理。书中采用区域算法来减少 CDS 的节点数量,从而可以高效管理网络。第 3 章介绍的是用于优化 MANET 介质访问控制(MAC)机制的一个智能机制。它通过释放未使用的保留信道,提高了信道效率同时增加了信道吞吐量。第 4 章介绍了移动环境中发布/订阅消息的模式,该方法将供应商与用户数据的通信进行分离,从而使移动应用具有隐藏间歇断线的功能。此外,它融入了只选择用户感兴趣消息的过滤机制,对无线网络中的有限可用带宽进行轮流优化利用。第 5 章定义了 MANET 中基于簇的合作缓存和预取机制,以提高查询性能,并对一些开放性问题进行了详细讨论。第 6 章介绍的是移动智能代理的部署,它是可直接提供移动环境数据与服务的一种方法。书中还介绍了移动代理如何使用户随时随地获得信息的基本原理。

第 2 部分(基于位置的移动信息服务)介绍的是利用智能方法解决无线移动环境下基于位置的

信息检索问题。本部分包括四章。第 7 章介绍 MANET 基于簇结构的智能信息服务协议，该协议支持多跳无线网络，具有很多优点，比如开销低，可扩展性、容错性好以及位置信息更精确等。第 8 章研究的是蜂窝以及 Ad Hoc 无线网络中的预测位置跟踪问题。本章阐述了其能够预报移动用户未来可能位置的特点，探究了位置预测的几个关键因素，并定义了一个用于位置跟踪的可行模型。该模型可以在任意无线网络中减少时延和资源消耗，并延长网络的寿命。尽管位置预测问题已经取得了重大进展，书中还是进一步提出了一些重要问题以待今后进一步探究。第 9 章是用于移动数据广播的高效无线索引机制。本章首先对文献中的索引机制进行分类和详尽讨论，而后作者提出了基于单元的分布式空间索引（CEDI）。与合并每个广播项指针不同的是，它将一组数据项的分布式结构进行智能合并，从而可以减少处理窗口查询时冗余数据项的监听，提高能量效率，减少查询访问的时间。第 10 章对无线环境下 Web 2.0 的部署进行了设计，Web 2.0 用于基于定位的服务（LBS），书中对 Web 2.0 如何有利于 LBS、如何工作以及如何实现进行了全面探讨，此外还着重讨论了一些开放性问题，如隐私保护机制的必要性。

第 3 部分（移动挖掘）包括两章，分别关于感兴趣模式和移动用户产生数据的知识。第 11 章对移动挖掘进行了全面介绍，是一篇非常好的学习移动物体数据挖掘的应用、当前趋势、问题和方法的参考资料。第 12 章则针对通过网页服务的移动设备上普遍存在的数据挖掘问题进行了讨论。本章阐述了使移动网页服务允许从移动设备远程执行数据挖掘任务并获得相应服务的实现可能性。

第 4 部分（移动环境感知和应用）由三章组成，探讨了智能机制作为移动应用支持环境感知的手段以及它的重要性。环境感知计算的基础原理和框架在第 13 章中介绍，以构成环境感知技术设计与评估的基础。第 14 章介绍的是作者关于普适中间件，用于处理用户环境、位置跟踪的移动代理系统，以及预测技术的开发经验，其中预测技术可以支持员工办公空间的动态分配。第 15 章介绍了环境感知计算中移动代理的应用，对于用户间私人通信问题的解决具有广阔的应用前景。书中作者提出了基于代理的环境感知通信系统，可根据用户的需求和偏好提供私人通信服务，并将提出的系统纳入处理冲突用户政策机制中。

第 5 部分（移动智能安全）由 4 章构成，试图通过采取不同的智能机制来解决移动环境中的安全问题。第 16 章介绍的是 MANET 中现有的路由方法，并指出了它们的缺陷，尤其强调了选择性丢包而非转发的自私或恶意代理。本章定义了一个叫做鲁棒信源路由（RSR）的新机制，该机制采用了先驱报文的概念，指的是沿路径通知所有节点数据包的流以及接收每个包的时限。这样就可以识别出恶意代理进而将其孤立。第 17 章引入了 MANET 检测威胁的在线机制，利用到神经网络的特殊形式即图神经元（GN）。该机制可实现模式识别，其中网络的状况就被当作模式。通过对这些模式实时地收集和分析，来发现网络中的入侵并检测威胁。GN 的一种扩展为分布式层次图神经元（DHGN），书中介绍了其优于标准 GN 的性能。第 18 章对 MANET 的安全路由协议进行了进一步的研究，书中强调的是中间节点间可信度的评估，以提高 MANET 的安全级别。其中引入了一个基于拒绝的信任模型，叫做 SMRTI 或带有信任机关的安全移动 Ad Hoc 网络路由。该模型能够捕获可信证据并通过多种方法对其评估，而后对一个恶意节点进行预测应用。第 19 章详细探讨的是基于定位访问控制系统的重要性、设计和需求，着重强调了隐私问题，并定义了一个叫做相关性的度量标准，用于度量位置的隐私性和准确性。一些示例的研究结果表明，该技术可以获得不错的效果。

第 6 部分（移动多媒体）注重于移动环境下智能方法用于多媒体检索的应用，共有四章。第 20 章讲的是具有语音 XML（VOICE）功能的智能移动环境感知的推荐系统。特别示例了餐馆推荐系统，它可以根据用户的偏好为用户有效提供一系列餐馆进行选择，并对用户选定的餐馆给出方向指示。第 21 章主要探讨移动计算中多媒体内容的高效管理与检索，提出了 MoVR 系统并作了详细介

绍。该系统采用了智能机制用于个人视频检索，包括分层马尔可夫模型调停（HMMM），该系统对 HMMM 文件进行组织建模来捕捉并存储个人用户的访问历史和偏好，从而可提供“个性化建议”，此外模糊关联概念使得用户可通过框架做出选择。为提高处理效率，在服务器端计算操作量大，而移动客户端的主要任务是管理检索到的媒体以及当前查询的用户反馈。该系统还支持移动设备的信息缓存。同样，书中也示例了一个移动足球视频的系统样本。第 22 章介绍的是普适时尚计算机（UFC），UFC 是一款可穿戴型计算机，可用于无处不在的计算环境中。该系统可支持不同通信链路（如蓝牙、Zigbee）的互操作，以及用户界面和 UFC 的直观使用。采用的用户界面叫做 i-Throw 或直接输入设备，可以识别手势、方向，并利用手势来控制普适设备。书中也介绍了开发的系统样本及其有效性。第 23 章对移动设备上采用的多种不同视频编解码器的能量消耗进行了深入、系统的测试，并给出了实验数据的总结，该总结对选择移动视频的最优编解码器及其参数很有帮助。

第 7 部分（智能网络）从移动计算向前进一步到普适计算层面，此处计算的概念是与环境紧密联系的，通常包括无线传感设备。该部分由四章组成，介绍的是用于解决无线传感器网络（WSN）不同问题的先进技术。第 24 章着重探讨了 WSN 数据存储的问题，讨论了现有解决方案并作了全面对比。文章结尾提出了 WSN 数据存储相关的开放性问题和进行了发展趋势展望。在典型 WSN 设置中部署了大量传感器设备，嵌入跟踪机制很有必要，尤其是在传感器被放置在恶劣环境中人们很难进行控制的情况下。为此，第 25 章通过引入 COLLECT 机制提供了一种解决方法。COLLECT 的设计融入了邻近三角、事件决策以及边缘传感器选择等过程，它可以快速地检测事件并予以跟踪。书中也探讨了文献中的其他方法以及待解决的开放性问题。另一个需要克服的重要问题是 WSN 攻击，一次 WSN 攻击会导致网络不稳定甚至陷入瘫痪。对此，第 26 章提出了一个分布式拒绝服务（DDoS）攻击的模型。DDoS 被视为洪泛攻击，它利用的是网络线路间速率与节点间处理能力的不对称性，目的在于操纵网络的流量密度，使其攻击对象或整个网络丧失功能。书中提出了一个基于自组织图（SOMs）的集中式算法来检测这种攻击。SOM 神经网络通过网络流量模式进行训练，模式包括攻击模式和正常情况。所提出的 SOM 方法可以在网络流量稳定的环境中很准确地检测出攻击。最后，第 27 章介绍了 WSN 中另一个关键问题，即能量效率。书中提出了一个名为得票图神经元（VGN）的模式识别方法，它利用传感器节点间的合作来检测事件的发生，并引入了一个新的高效节能的模板匹配方法。这种方法可以使传感器节点在活动模式和睡眠模式之间转换，通过动态控制节点之间的协作来达到节省能量的目的。

作为一个非常崭新的研究领域，移动智能带来了非常多的研究机会，使研究者、工程师和开发人员能够开发出更多有趣的应用。随着移动用户知识的增多，我们相信，随之而来的应用和系统必将更加密切地贴近我们的日常生活。

本书的读者不仅可以学到移动计算不同方面的内容，还能看到相关问题的智能解决方法以及移动模式中的其他方面。

祝您阅读愉快。欢迎读者评论及反馈，谢谢！

Laurence T. Yang
Agustinus Borgy Waluyo
Jianhua Ma
Ling Tan
Bala Srinivasan

目 录

第 1 部分 移动数据及人工智能

第 1 章 移动 Ad Hoc 网络的尖端路由协议综述	1
第 2 章 Ad Hoc 网络的连通支配集拓扑控制	16
第 3 章 一种面向移动 Ad Hoc 网络提高信道利用率的智能方法	27
第 4 章 发布/订阅系统的移动性	39
第 5 章 移动 Ad Hoc 网络中自适应协同缓存的跨层设计框架	54
第 6 章 面向移动 Agent 应用的研究进展	66

第 2 部分 基于位置的移动信息服务

第 7 章 KCLS: 基于簇的定位服务协议及其在多跳移动网络中的应用	87
第 8 章 蜂窝和 Ad Hoc 无线网络中的预测位置跟踪	99
第 9 章 一种移动计算环境中面向空间数据分发的高效空间索引方案	116
第 10 章 下一代基于位置的服务: 将定位与 Web 2.0 融合	132

第 3 部分 移动挖掘

第 11 章 面向移动对象数据库的数据挖掘	146
第 12 章 在小型设备上通过 Web 服务的移动数据挖掘	163

第 4 部分 移动环境感知和应用

第 13 章 上下文感知: 形式化基础	171
第 14 章 智能办公室项目的体验	182
第 15 章 提供增强通信服务的基于代理的架构	198

第 5 部分 移动智能的安全

第 16 章 MANET 路由安全性	212
第 17 章 移动 Ad Hoc 无线网络的在线威胁检测方案	233
第 18 章 有信任纠葛的安全移动 Ad Hoc 网络路由	251
第 19 章 基于位置的接入控制系统的隐私管理	266

第 6 部分 移动多媒体

第 20 章 支持 VoiceXML 的智能移动服务..... 285

第 21 章 移动设备上的用户自适应视频检索 296

第 22 章 一个基于位置服务环境中具备 i-Throw 设备的普适时尚计算机..... 310

第 23 章 有关移动多媒体播放能量效率的探讨..... 323

第 7 部分 智能网络

第 24 章 无线传感器网络中以数据为中心的高效存储机制..... 331

第 25 章 无线传感器网络的跟踪 344

第 26 章 无线传感器网络中 DDoS 攻击建模与检测..... 358

第 27 章 无线传感器网络中的高能效模式识别..... 376

第 1 部分 移动数据及人工智能

第 1 章 移动 Ad Hoc 网络的尖端路由协议综述

1.1 引言

传统的网络基础设施难以在即时的场景中提供计算机网络服务,而移动 Ad Hoc 网络(MANETs, Mobile Ad Hoc Networks)的出现使这成为了可能。移动网络通常是基于无线通信的,因此很多建立在有线网络基础上发展的技术无法直接应用于移动网络。特别地,由节点构成的移动网络不依赖于任何基础设施,可以短时间内在空中形成网络。形成移动网络的主要挑战在于节点的移动性、无线介质的特性、小型移动节点的能量约束以及节点可能在网络周期内随时加入或离开网络的不确定性。

由于节点的移动性,网络中路由的生命周期通常很短。有线网络中的节点通常是在固定的地理位置上,与有线网络不同,无线网络中的路由在数据通信会话的整个期间可能出现,也可能不出现。无线介质的约束性是指,节点的任何通信都是通过数据包广播完成的。由于无线介质的带宽相较于有线网络来说要窄很多,带来的问题就是几乎所有通信都会由于数据包的洪泛而占用大量的网络带宽。此外,对有限带宽的争夺会导致数据包的冲突和重传,进一步造成可用带宽的浪费。MANET 中的节点通常由电池供电,在网络对话过程中可能难以充电,这也进一步增强了无线介质的约束性,也即节点不应该把能量浪费在因冲突丢失的数据包重传上。节点消耗在传送和接收数据包上的能量几乎是相等的,所以对传往其他节点的数据包进行监听也是种能量的浪费。

无论对于有线计算机网络,还是无线计算机网络,数据包的路由选择都是最基本的操作之一。MANET 中的所有应用都依赖于可靠和有效的数据包路由选择。因此,设计出适应于 MANET 约束性并能为更高层应用提供支持的路由协议极其重要。过去 10 年来,为设计出有效的 MANET 路由协议,各方学者对此做了大量的研究工作。互联网工程任务组(IETF, Internet Engineering Task Force)^[1]的 MANET 工作组正在考虑将这些协议中的某些进行标准化。但由于文献所提出的路由协议不尽相同,这对于 MANET 工作组来说相当困难。目前,工作组已将四种协议纳入了考虑范围,并为这些协议的互联网草案公开征求意见。

本章旨在对面向 MANET 的不同路由协议给出一个简明而全面的概述。本章将特别关注 MANET 工作组纳入考虑范围的协议,因为这些协议是目前为止所提出的路由协议中最为有效和可靠的协议。同时也将从历史的角度来讨论路由协议的演变和发展,并就其他协议中独特而有趣的理念进行探讨。本章中所讨论协议的更多细节可参考 Belding-Royer 和 Toh 的论文^[2]以及 Perkins 的著作^[3]。

本章的其余部分安排如下:1.2 节讨论路由协议的分类;1.3 节讨论主动路由协议的分类;1.4 节就反应式路由协议进行讨论;1.5 节介绍一些其他类别的协议;1.6 节就以上内容做出总结。

1.2 MANET 路由协议的分类

所有针对 MANET 设计的路由协议的主要目的均是获得高性能，如高吞吐量、低延迟、各节点的低耗能等。然而上述目标往往都是自相矛盾的，因为路由协议为了满足其中一方面的性能要求时可能必须要牺牲另一方面的性能要求。例如，假设所设计的路由协议是为了确保数据包传输时具有较低的延迟，这也可能是在网络中节点间传输多媒体内容时服务质量（QoS, Quality of Service）的需求。如果单个节点想要快速地传递或发送数据包到目的节点，它们就必须对网络拓扑结构有着明确的认识，并且尽可能准确地知道通往目的地的路径。不过，要想收集到精确的拓扑信息，就需要节点间交换局部视图。换句话说，每个节点都应时常将自身邻域内的情况告知其他节点，这样所有的节点都能获得最新的网络拓扑信息。这种类型的信息交换是通过发送控制消息（该名字用于区分实际的数据包）来完成的，并且整个过程需要节点消耗大量的能量。因此，路由协议可能不得不以牺牲电池能量为代价来换取低延迟。

所有的路由协议都隐含地设定 MANET 中的节点在传送包时是与其他节点相互配合进行的。从包的角度来看，MANET 中的节点可分为三类：发送节点、接收节点和转发节点，转发节点的任务是竭尽所能将数据包发送至目的地。MANET 的安全问题不在本章范围内，我们假设网络中的所有节点都是可信的。读者可参考 Pirzada 等的论文^[4]，相关参考文献中会涉及更多有关 MANET 安全性问题的探讨。

MANET 的主要路由协议按其收集网络拓扑信息方式的不同可分为两类：主动式和反应式。主动式协议在包传递过程中通过在整个网络中积极传播拓扑信息来降低延迟。但这也带来了不利影响，无线介质中的大量可用带宽都用于发送控制信息。因此，在设计主动协议时会面临这样一个挑战性的难题，即如何才能既减小控制消息在网络中的影响，又使包传递过程达到可接受的延迟水平。有关此问题的讨论将在 1.3 节中进行详细论述。

反应式协议是通过减少网络中的控制消息从而最大限度地减少带宽浪费的一种协议。它不依赖拓扑信息的主动收集，而是采取按需方式来寻找路径。不过这种方法往往会加剧包传递的延迟。因此在设计反应式协议时的挑战性难题是，如何既能降低延迟，又能保持少量的控制消息。采用反应式协议的移动节点往往是使用间接方法，例如监听过往的无线数据流来提高对网络拓扑信息的认知。本章将在 1.4 节对该问题进行深入探讨。

此外还有一些既能结合主动式协议和反应式协议的优点又能摒除其缺陷的协议。这些协议在每个节点的小范围邻域内采用主动协议，在整个网络范围内采用反应式协议，这样一来就能够管理控制消息的数量。区域路由协议（ZRP, Zone Routing Protocol）是其中最为值得关注的，将在 1.5 节中简要对其进行讨论。

在 1.5 节中还将讨论一类叫做链路反向协议的重要协议。这些协议基于在 MANET 中保持有根有向无环图（DAG, Directed Acyclic Graph）这一简单思想，其主要目的在于保持 DAG（因此也就保有路径），不过这些协议的额外开销通常也是非常大的。

1.3 主动路由协议

主动路由协议通过主动地交换本地拓扑消息，来收集尽可能多的 MANET 信息。目的序列距离向量（DSDV, Destination Sequenced Distance Vector）协议^[5]是最早的 MANET 协议之一，也是最著名的主动路由协议之一，该协议对控制消息的开销巨大，因此随着更有效的反应式协议的出现，如

动态源路由 (DSR, Dynamic Source Routing) 和 Ad Hoc 按需距离向量协议 (AODV, Ad Hoc On-demand Distance Vector), DSDV 便逐渐失宠。不过, 包传送过程中的低延迟仍是主动协议最具吸引力的一个方面。近年来, 相当数量的研究工作都旨在降低 DSDV 协议的开销, 其中最为成功的是最优链路状态路由 (OLSR, Optimal Link State Routing) 协议^[6], 目前它已被纳入 MANET 工作组考虑范围。本节中将首先讨论 DSDV 协议, 随后就 OLSR 协议进行探讨。

1.3.1 目的序列距离向量协议

首先, 本节列出关于 MANET 路由协议中的几个特点, 它们同样适用于本章讨论的所有协议。路由协议是 MANET 中每个节点执行的一种分布式算法, 也就是说, 每个节点对数据执行的是它们在本地产生的本地协议副本。此外, 这种协议的分布式执行旨在达到整体性能目标, 如高吞吐量、低延迟和减少能量总开销等。下面用名词包和消息来表示无线介质中节点发送的包。

所有的路由选择都是由单个节点基于本地路由表所做的决定, 从这种意义上来说, DSDV 是一种表驱动协议。DSDV 协议分为两个部分, 即保持本地路由表的实时更新和根据路由表计算路径。首先讨论第二部分, 因为这将能明确节点如何通过执行 DSDV 协议找到最佳路径。随后会讨论节点如何更新其路由表。

首先, 假设每个节点在各自的路由表中都已完成对网络拓扑最新信息的收集。给定一个源节点 S 和一个目的节点 D , 路由表的目的是根据 S 和 D 之间的某种度量, 寻找到最佳路径。在 MANET 中, 中间节点 (如除 S 和 D 以外的节点) 负责将接收到的包从 S 转发到 D 。因此, 从中间节点 I 的角度来看, 最佳路径就是从 I 起始、以 D 为目的地的一条路径。

本节将从抽象的图论角度对路由表进行分析, 这种分析方法同样适用于其他类似的表。将 MANET 中的节点看作所提出的图形中的节点, 如果两个节点位于彼此的传输范围内, 那么两者之间就会有链路或者一条边界。实际上, MANET 中相邻节点彼此的传输范围可能并不相同, 因此节点间的链路可能不是双向的。为了使协议便于描述, 将情形简化为节点间的链路总是双向的。不过, 当链路是单向时, 我们的描述也可依情况进行适当的修改。现在, 可以将路由表视作图形的邻接矩阵, 代表底层的 MANET。可以在矩阵的第 i 行和第 j 列交叉处, 用 1 表示节点 i 和 j 间存在边界, 用 0 表示不存在边界。

现在, 可以在邻接矩阵利用 Dijkstra 最短路径算法来寻找最短路径。假设源节点 (每个节点对包来说都是潜在的源节点) 或中间节点 I 必须为包找到到达目的地 D 的最佳路径。节点对其路由表运用 Dijkstra 最短路径算法, 找到通往 D 的最短路径, 这个最短路径必须经过该节点的相邻节点。接下来, 我们的任务是将包转发到这个相邻节点。相邻节点依次对其路由表运用 Dijkstra 最短路径算法找到通往目的地的最佳路径, 并且重复转发数据包的过程。假设在使用 Dijkstra 算法寻找最佳路径时, 跳数是唯一的度量; 不过也可以用一些与无线介质相关的度量。这些度量可以是链路成本带宽、链路延迟等。最短路径的选择如图 1.1 所示说明。

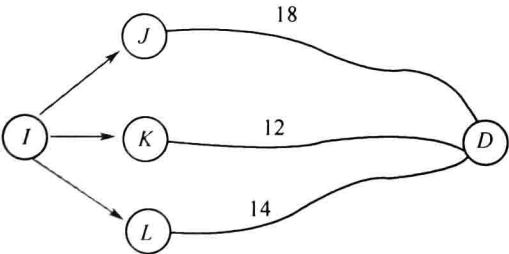


图 1.1 DSDV 协议中转发节点选择示意图

节点 I 运用 Dijkstra 最短路径算法在路由表中找到了通往目的地 D 的三条不同路径。由于花费在 K 和 D 之间路径的成本最少, I 选择相邻节点 K 作为包的下一个跳点。节点的 ID 显示在节点内部, 从节点 J 、 K 、 L 出发的路径成本显示在其路径上。

现在将关注焦点转向 DSDV 协议的另一个部分，即拓扑信息的收集。如果期望节点通过运用 Dijkstra 最短路径算法寻找到最佳路径，那么每个节点的路由表就应该随时进行更新。值得注意的是，MANET 中不是所有节点的路由表中包含的信息都是正确的。换句话说，由于节点的移动性，相邻节点间已经被记作链路的所有链路在协议执行期间可能会随时消失。比如 MANET 中有三个节点 i 、 j 、 k ，节点 i 有可能记录了节点 j 、 k 之间的链路信息。不过，通常来说这种信息是比较旧的，在节点 i 运用此信息来计算最短路径时，由于节点 j 、 k 的移动性，两者之间的链路可能已经断开。

值得注意的是，集中式算法和分布式算法是不同的。实际上，集中式算法在执行前具有完整的输入，而分布式算法中每个节点只有部分输入，并且节点不得不根据这些部分输入来执行算法。除此之外，由于节点的移动性，MANET 的路由协议甚至无法获得正确的部分输入。因此，MANET 的任何路由协议都不是传统意义上的精确算法。从某种意义上来说，路由协议是运用不完整的、部分错误的输入来试图达到最优结果的一种算法。

执行 DSDV 协议的节点通过交换各自的路由表来更新它们对网络拓扑的认知，如一个节点 i 和它的路由表。假设 i 目前有三个相邻节点 j 、 k 、 l 。如果三个节点中的任意节点，如 j ，移出了 i 的传输范围，那么 i 的路由表将会更新。这种类型的变化将会触发 i 面向网络中所有节点的广播，其他节点也将依据已经改变的拓扑结构来更新自己的路由表。 i 将自己的路由表发送给当前的相邻节点，这些相邻节点再将其发送给它们的相邻节点，以此类推，广播就是通过这样的方式完成的。如果节点 m 接收到上述来自节点 i 的广播，那么 m 的路由表也会改变，之后再由 m 发起广播。

由于节点移动性而引发的节点局部拓扑结构的改变，使 MANET 中的所有节点都需要广播它们的路由表，显然，这是一个费力的过程。带宽的一大部分都消耗在了这些用于更新的数据包上，特别是在高移动性场景中。对此，在原先 DSDV 协议的基础上提出了一些减少开销的建议，将 DSDV 发送路由表的所有更新信息改为只发送更新的增量。增量更新方式的思想是，只广播路由表中更新的部分，而不是广播整个路由表。但是，事实上每个更新都会洪泛至整个网络，即使在中度移动性场景中，就数据包的吞吐量而言，协议也会变得低效。

我们通过论及 DSDV 协议的另外一个重要特征，即包的序列号分配，来对整个讨论进行总结。从节点 i 向 MANET 内所有节点广播数据包 p 的结果是，另一节点 j 会得到 p 的多重副本。除此之外，如果 i 在两种不同情况下发送了两种不同的路由表更新内容，那么将不能保证这两个数据包以正确的顺序抵达 j ，也就是说，先发送的数据包有可能晚到达。由于 DSDV 依赖正确的拓扑信息进行路由选择，所以节点利用最新信息来更新其路由表就变得尤为重要。确保节点能够利用最新信息的途径之一就是在当前时间对每个数据包做时间戳。如果 j 收到从 i 发送的两个数据包， j 可以通过时间戳来判断哪个数据包比较新。不过这种方法只在所有节点保持时钟同步的情况下才适用。时钟同步在分布式系统中并不易实现，特别是在 MANET 中，所以并不期望节点能够访问任意基础设施。另一种方法是使用序列号作为逻辑时钟。每个节点都对数据包进行戳记，这样数据包在广播时就会附带一个递增的整数值，称这个整数值为序列号。任意节点 j 收到从 i 发送的两个数据包时，都可以通过比较数据包序列号的大小来判断哪个数据包比较新。

1.3.2 最优链路状态路由协议

虽然 DSDV 协议因其寻找路由时的低延迟而备受关注，但是路由表广播也带来了控制包高开销的弊端。任何广播都会洪泛于整个网络中，并且会占用无线介质中的大部分可用带宽。OLSR 协议是致力于降低 DSDV 协议控制开销从而提高数据包传输吞吐量的一种相对较新的协议，它主要通过两种途径对 DSDV 协议进行改进：一是降低更新信息的长度；二是降低广播的影响。下面就这两种途径分别进行讨论。

执行 OLSR 协议的节点所广播的内容不再是路由表，而是链路状态。假设节点 i 目前有三个相邻节点 j 、 k 、 l 。如果其中一个节点，如 k ，移出了 i 的传输范围，那么拓扑结构将发生改变，并且 i 会将这一改变通过广播通知给其他节点。注意， i 是有能力通知其他节点链路 $i-k$ 已经断开这一信息的，并且其他节点也可以通过这个信息来更新自己的路由表。因此，相较于路由表，广播链路状态信息能更好地将控制消息保持在低量状态。

OLSR 协议主要的改进之处在于它减轻了数据包广播的影响。对于节点 i ，由于无线传输是一种全方位式的传输方式，所以所有相邻节点都会收到节点 i 发出的数据包广播。但让 i 的所有相邻节点再转播数据包是不必要的，只要令 i 的相邻节点的子集进行数据包转播即可。OLSR 协议利用相邻节点的子集来更远地传播数据包，这样的子集称为多点中继。严格地说，多点中继这一概念并不是 OLSR 协议的一部分，更确切一点，它是用于无线网络中降低数据包广播量的一种概念。可以通过一个例子详细说明多点中继的作用。节点 i 有三个相邻节点 j 、 k 、 l ，这三个相邻节点称作 i 的单跳相邻节点。自 i 的两次传输可得到两跳相邻节点。假设 i 有 6 个两跳相邻节点 a 、 b 、 c 、 d 、 e 和 f 。显然，每个两跳相邻节点都至少有一个 i 的单跳相邻节点作为其相邻节点（否则它们就不能成为 i 的两跳相邻节点）。如之前所提到的，节点 i 的多点中继，记作 $MPR(i)$ ，是 i 的单跳相邻节点的子集，那么 i 的所有两跳节点即存在于 $MPR(i)$ 的子集中。为了说明这个问题，假设 k 和 l 统称为 a 、 b 、 c 、 d 、 e 和 f 的相邻节点，选择 k 和 l 作为 $MPR(i)$ 的元素，也就是说只有这两个节点转发 i 的数据包广播。这时， j 就无需广播来自 i 的数据包。 MPR 子集将由 MANET 中的所有节点通过跟踪它们的单跳和两跳相邻节点来选出。结果表明利用 MPR 集的方式可以极大减少广播开销。通过多点中继进行广播的过程，如图 1.2 所示。

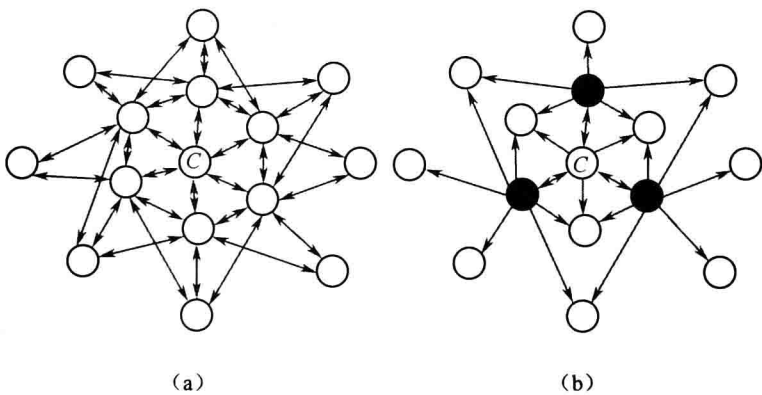


图 1.2 简单洪泛及经多点中继洪泛的图示说明

图 1.2 (a) 展示的是简单洪泛。每个节点在接受到最初由中央节点 C 发送的包时都会再广播一遍此包。两节点之间的双向箭头表示互相接收对方的广播（假设此为对称通信链路）。通过使用多点中继节点（黑色节点），数据包的数量大大减少，图 1.2 (b) 展示的便是这一现象。图中剩下的点表示的是 C 的单跳相邻节点和两跳相邻节点的所有相邻节点。当非多点中继节点的单跳相邻节点不进行广播时，数据包总量将大大减少。

OLSR 协议目前正在接受 IETF 的 MANET 工作组的审核，该协议具有高吞吐量和低延迟的理想特性。

1.4 反应式路由协议

反应式路由协议的主要设计初衷是降低主动路由协议的控制数据包开销。由于这类协议不能主动维护路由表，因而也不能按照需求尽快地寻找到路径，通常在寻找路径时会出现延迟。可用带宽将不再用于主动协议中控制包的常规洪泛，而是在反应式协议中为数据包传输所用，因此包传输将获得高吞吐量。不过，反应式协议在有需求时也需要数据包洪泛或广播。下面讨论两个最重要的反应式协议，即 DSR 协议和 AODV 协议。

1.4.1 动态源路由协议

DSR 协议^[7, 8]分为两个不同的阶段，即路由发现和路由维护。若源节点 S 欲寻找一条通往目的节点 D 的路径，这时就会开始路由发现阶段。一旦通往 D 的路径被找到，当 S 利用发现的路径传输数据包时，就开始了路由维护阶段。下面将详细讨论这两个阶段。

源节点 S 通过向其相邻节点发送路由请求（RREQ, Route Request）包来启动路由发现阶段。RREQ 包有一个标识符，包括一个源节点、一个目的节点和一串连续的整数 ID 号。如果中间节点 I 接收到了 RREQ 包，但并不知道通往目的节点 D 的路径，那么它将采取以下两种行动之一。若 RREQ 包是新的（即 I 以前从未见过此 RREQ），那么 I 将 ID 号附加在 RREQ 的包头之后，再将包广播给相邻节点。同时， I 也将这个包存入列表中，这样就可以将这个包的标识符与之后接收的 RREQ 包进行比较。若 RREQ 包是旧的，也就是说 I 之前接收过此包， I 就会将这个包丢弃。

如果 I 已知一条通往目的地 D 的路径（下面会讨论 I 是如何知道路径的），节点 I 将发起一个路由回复包（RREP, Route Reply），具体过程是这样的，将 I 到 D 和 S 到 I 的路径附加到 RREP 包头中，并且将包发回相邻节点，相邻节点就会接收到相应的 RREQ 包。注意，现在 RREP 包的包头中已有了累积的从 S 到 I 的路径。因此，任意接收到 RREP 包的中间节点都会确切地知道该向哪个相邻节点发回 RREP 包。最后，源节点 S 接收到 RREP 包。至此，路径发现阶段结束。如果中间节点都不知道通往 D 的路径，RREQ 包就会到达目的地 D （所给目的地与 S 一样处于网络相同连接部分），并且 D 会发出 RREP 包。如果源节点 S 在指定的一段时间内没有接收到路由回复，那么它可以为数据包分配一个新的 ID 号，之后发出新的 RREQ 包。DSR 协议的路由发现过程如图 1.3 所示。

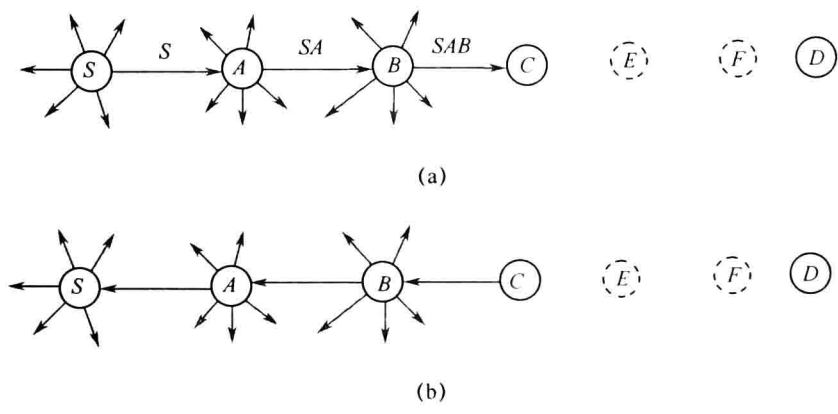


图 1.3 DSR 协议路由发现阶段的图示说明如图 1.3 (a) 所示， S 通过广播 RREQ 包来开始路由发现阶段。节点周围的箭头表示全方位式广播。每个节点将通往源路径的 ID 附加在 RREQ 包头中。节点 C 经过节点 E 和 F ，具有一条通往目的地 D 的路径。如图 1.3 (b) 所示， C 向 S 发回 RREP 包。每个节点通过检查源路径来确定 RREP 包的下一个跳点。

路由发现阶段最为重要的地方就是将所有中间节点的 ID 附加到 RREQ 包和 RREP 包的包头中，有时也称 ID 列表为源路径。执行 DSR 协议的节点并不能维护路由表，因此需要这样一种机制，当接收到 RREQ 和 RREP 包后，可以通过某个节点决定转发包的去向。这种情况对于 RREQ 包来说较为容易，因为它需要被广播至所有相邻节点。但是，RREP 包的目的是传送数据包至发起路由发现阶段的源节点 S ，因此，每个接收 RREP 包的中间节点都需要获悉最初发送相应 RREQ 包的前一个节点（朝向 S ）。将源路径附加至 RREP 包头的目的就是提供此信息。此外，RREQ 包也需要携带源路径，因为发起 RREP 包的节点需要从相应的 RREQ 包中复制此信息。

一旦 S 接收到一个 RREP 包，它便具有了一条通往 D 的路径，并会获悉这条路径上所有节点的

ID 号, 也就能根据这条路径开始数据包的传输。 S 将整个源路径附加至每个数据包的包头中, 这样任意中间节点都沿着这条源路径准确地向其相邻节点转发数据包。对路由发现阶段而言, 维护已发现的路径是十分重要的。但是由于节点的移动性, 路径在数据传输中有可能被破坏。例如, 假设路径中的其中一个链路是 $G-H$, 这里 G 和 H 是链路的终端节点。如果现在 H 移出了 G 的传输范围, 那么从 S 到 D 的整条路径将被破坏。在这种情况下, 节点 D 通过使用包头中的源路径, 发送路由错误 (RERR, Route Error) 包至 S 。RERR 包在除了此时的目的是通知 S 路径已被破坏之外, 其他与 RREP 包十分相似。现在 S 有两种选择, 如果有备用路径的话就开始使用备用路径; 如果没有, 则开始新一轮的路由发现来寻找新路径。

到目前为止, 本节提到了执行 DSR 协议的节点无需使用路由表。但是, 实际上在 MANET 中重复运行路由发现阶段代价十分高昂, 因为每一次路由发现本质上都是 RREQ 包在整个网络的洪泛。因此, DSR 运用一种叫做路由缓存的数据结构来降低洪泛带来的影响。路由缓存的目的在于储存节点搜集的任何信息, 这些信息要么来自于接收或转发的包, 要么来自于过往的数据流。特殊情况下, 节点可能利用 IEEE 802.11 标准中所允许的混杂模式操作, 即节点能够无意中监听到非预期的数据流。路由缓存的工作可以通过一个简单的例子来加以说明。考虑一个节点 I 和目的地 D 。假设 I 作为中间节点, 负责转发源节点 P 的资料包到 D , 假设 I 到 D 的路径同时也经过其他中间节点 E 、 F 和 G 。现在 I 在其路由缓存中储存了路由片段 $I-E-F-G-D$ 。如果 I 从其他源 S 接收到去往目的地 D 的 RREQ 包, 它将会利用已缓存的路径发送 RREP 包给 S 。

但是, 由于 MANET 中节点具有移动性, 缓存路径提供的信息可能是不正确的。例如, 在上述例子中, 假设节点 G 移出了 F 的传输范围, 此时链路 $F-G$ 断开。但是 I 并不知道链路被破坏的消息。如果现在 I 发送 RREP 作为对 S 发来 RREQ 的回复, 那么它会报告路径已被损坏, 就会带来源自 S 的进一步的路由请求这一难题。一种有效的策略是暂停该缓存的旧入口, 因为由于网络的移动性, 旧入口很可能会无效。ns-2 仿真器^[9]就是采用上述策略, 将存在超过 20s 的缓存入口暂停。

DSR 协议也采用其他一些优化措施, 以降低控制包开销和提高吞吐量。由于篇幅限制, 这里只讨论其中两种。第一种优化叫做包抢救。如前文所提到的, 中间节点在路径中检测到链路破坏后, 会发送 RERR 包给源节点。此外, 中间节点试图通过路由缓存中的备用路径将数据包传送至目的地。换句话说, 中间节点为了抢救数据包, 通过备用路径将数据包传送至目的地。当节点转发数据包且不幸与其相邻节点遭遇链路故障时, 包抢救降低了这些节点资料包丢失的可能性。由于在收到通知路径损坏的 RERR 包以前, 源节点会一直发送数据包, 因此网络中会存在大量的数据流, 此时中间节点的缓冲区可能溢出。

另一种重要的优化与回复路径请求发送的 RREP 有关。在某种情况下, 有些节点的路由缓存是最新的, 同时很多不同的节点可能也正处于发送路由回复的状态。但是, 由于路由回复, 网络可能会洪泛。因此, DSR 利用了这样一种策略, 即节点在发送路由回复前等待一段随机的时间。如果节点无意中监听到了另一节点已对相同路由请求发送了路由回复, 或者源节点已开始使用备用路径, 该节点便不再发送路由回复。这种优化大大降低了网络的开销。

就吞吐量方面来说, 即使在高移动性情景中, DSR 都是最为高效的协议之一。此外, DSR 通常能够通过其路由机制寻找到最短路径。但是, DSR 有一个缺陷就是在每个包中都用到了源路径。随着路径长度的增加, 源路径中 ID 号的数量也随之增加。由于无线介质通常支持尺寸相对较小的包, 如果路径长度太长, 就不可能在一个单独的包里保存整个源路径。另一方面, DSR 无法保证数据包在无线介质中按正确的顺序传输, 所以将源路径划分到多个数据包的方法也不能解决问题。因此, DSR 对于规模较小的路径在 10 跳左右的 MANET 来说是一种十分有效的协议。

1.4.2 Ad Hoc 按需距离向量协议

AODV 协议^[10, 11]通过消除控制包和数据包中源路径的方式来克服 DSR 协议的主要弊端。此外, AODV 是首个支持在 MANET 中进行多播的协议。至此, 本节已经从单播角度讨论了路由的相关内容, 即单个源节点发送数据包至单个目的节点。但是, 这只是应用于典型网络中的其中一种通信类型。在多播中, 单个源节点有可能意图将相同的包发给多个目的节点(称作多播组)。广播是多播的一种特殊情况, 这时源节点是向网络中的所有节点发送相同的数据包。在网络中, 通过多播组中找到从源节点到所有节点的单播(或一对一)路径来支持多播。不过, 发现和维持多个节点的独立路由的代价通常是十分高昂的。对于多播组来说, 其组内成员若能有一个共同的路由机制将是非常可取的。AODV 巧妙地解决了上述问题, 本节将在后面对其进行讨论。

AODV 在某种意义上来说是一种表驱动及反应式协议。每个执行 AODV 的节点都有一个路由表, 不过这个路由表是本地的, 并且只包含其相邻节点的信息。与 DSDV 相反, 这里不需要通过全网广播来更新路由表。像 DSR 一样, AODV 也有两个阶段, 即路由发现和路由维护。路由发现阶段与 DSR 十分相似, 不同之处在于 AODV 不使用源路径, 并且每个 RREQ 包由数据包的源节点戳记一个序列号。源节点 S 利用 RREQ 包的四个不同字段, 分别为自身的 IP 地址、目的地 IP 地址、自身的序列号和目的地 D 的最后一个已知序列号。 S 有可能从过去转发的包中获得 D 的序列号, 但是目前 S 还没有通往 D 的路径。数据包中包含 D 的最后一个已知序列号的目的在于把 S 获得的关于 D 的信息质量通知给其他节点。假设 S 通向 D 的序列号为 seq_1 , 另一个节点 I 通向 D 的序列号为 seq_2 。假设 I 现在接收到一个从 S 到目的地 D 的 RREQ 包。两个序列号均为整数, I 可以对其进行比较。如果 $seq_1 > seq_2$, 则与 I 相比, S 获得了有关 D 的最新信息, 由于 I 中包含的信息较旧, 从而 I 不需要对此 RREQ 包做出应答。相反, 如果 $seq_1 < seq_2$, 则与 S 相比, I 获得了有关 D 的最新信息。因此, 与 DSDV 协议相似, 序列号被作为逻辑时钟使用。

AODV 使用这样一种机制: 在路由发现阶段, 一条路径上的每个节点都设置前向和反向路径。现在通过一个实例来说明这种机制。假设中间节点 J 从另一节点 I 处接收到 RREQ 包, 并且此 RREQ 包的源是节点 S 。 J 在其路由表中设置反向路径入口, 并且将 I 作为入口的目的地。如果 J 未来接收到对此 RREQ 包作出回复的 RREP 包, 则此反向入口帮助 J 为数据包安排好了途经 I 、通往 S 的路径。 J 同时保存了此反向路径入口的当前时间。反向路径入口若在指定的生命周期内未被使用, 那么它将被清除。

假设节点 M 接收到其相邻节点 N 发来的 RREP 包。 M 在其路由表中创立了前向路径入口, 将 N 设为其前向路径的目的地。如果源节点 S 发送数据包, M 可将这些数据包通过 N 发送给 D 。只有当节点 J 具有到 D 的未到期路径, 且 D 具有更高序列号(与 RREQ 包中的序列号相比)时, 节点 J 可用 RREP 来回复 RREQ。RREP 包沿反向路径入口将 RREQ 传往 J 。图 1.4 说明了 AODV 协议中的前向和反向路径入口。

在已建立的路径上进行数据传输时, 若因链路故障出现路径损坏, 这时就进入了路由维护阶段。AODV 协议的 RERR 包发送过程与 DSR 协议十分相似, 只不过执行 AODV 协议的节点在链路损坏的情况下不去抢救包。当源接收到 RERR 包时, 它将开启一个新的路由发现阶段。

原始的 AODV 协议中, 每个节点对每条路径只保存单一的前向路径入口。一旦发生路由损坏, 路由发现阶段就要马上开始, 此外还会产生大量的控制包开销, 因此协议会变得很低效。这时候, 一种叫做多路径 AODV (AOMDV, Multipath AODV)^[12]的 AODV 协议的变体应运而生, 在此协议中, 每个节点为每条路径都保存多个前向路径入口。例如, 假设源节点 S 请求了一条通往目的地 D 的路径, 产生的 RREQ 包经过中间节点 J , 那么 J 就有可能在以后接收到多个对此 RREQ 包回复的 RREP 包。假设 J 接收到三个相邻节点 U 、 V 和 W 发来的 RREP 包。在原始 AODV 协议中, J 会选

择三个节点中的一个在路由表中对其维持一条前向路径入口，而舍弃其他两个。而在 AOMDV 协议中，对于上述情况， J 会同时维持三个相邻节点的前向路径入口。不过，当相应的 RREP 包被接收到时， J 会对这些前向路径入口做时间标记，如果入口在使用前已经比较旧，那么将被暂停使用。假设当 S 开始传输数据时， J 选择其中一个入口，如 U ，来转发数据包。如果之后 U 移出了 J 的传输范围，破坏了 $J-U$ 链路和通往目的地 D 的路径，那么 J 将开始使用另一个前向路径入口来转发包到 D ，比如到 V 的链路，只要这条链路不是很旧就可以。因此，在 AOMDV 协议中，不需要因 $J-U$ 链路被破坏而向 S 发送 RERR 包。不过，如果 J 的路由表中的最后一个前向路径入口遭到损坏，那么仍需发送 RERR 包。AOMDV 协议与 AODV 协议相比，在低数据包开销方面有着更优越的性能。图 1.5 用来说明 AOMDV 协议的前向路径入口。

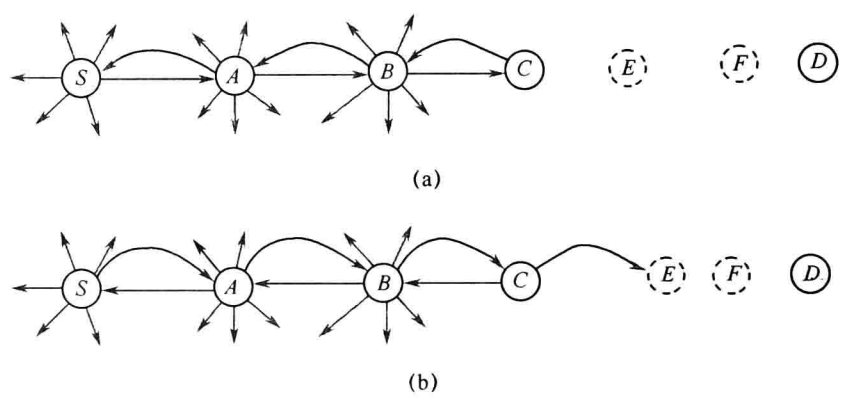


图 1.4 AODV 协议中路由发现阶段前向和反向路径的建立过程说明。图 (a) 中，自源节点 S 的 RREQ 包的传播由图中的直线箭头表示。每个节点通过在其路由表中记录接收到 RREQ 包的相邻节点的 ID 来建立反向路径入口。图中反向路径入口用粗体弯曲箭头表示。图 (b) 中，当通往目的地 D 的路径存在时，节点 C 通过发送 RREP 包来发起路由回复。当节点接收到自其相邻节点发来的 RREP 包时，节点通过在其路由表中记录接收到 RREP 包的相邻节点的 ID 来建立前向路径入口。RREP 包的传播由图中的直线箭头表示，前向路径入口由图中的粗体弯曲箭头表示。注意，由于像 E 、 F 这样的节点是通往 D 的路径中的一部分，因此具备前向路径入口。

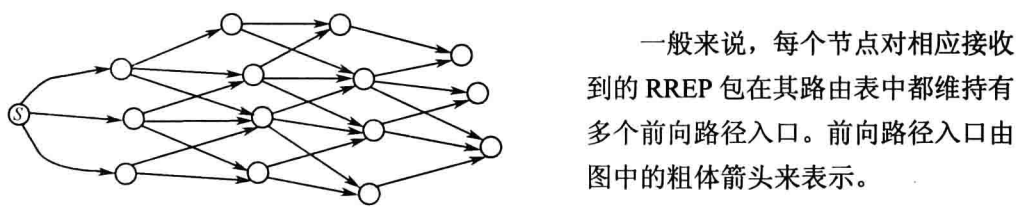


图 1.5 AOMDV 协议的示意图

接下来，讨论 AODV 如何支持 MANET 中的多播。AODV 的目的在于保持每个节点多播组的多播树。一个多播树包含两类节点：多播组成员和多播树成员。从多播组的观点看，MANET 中的其他所有节点都是非树节点。多播树的成员不是多播组的成员，不过在保持多播树的连接时会需要它们。保持多播树的连接是十分重要的，这样当多播树成员接收到数据包后，可以将包发送给任意其他成员。此外，用树代替图还可以避免路由环路。

多播树可以看作是一个单一实体，其任务是在 AODV 中进行路由发现。任何 RREQ 包中都应该包含一个目的节点的 ID。节点 I 在以下两种情况下可以发送 RREQ 包到多播组 M ：一是 I 欲将 RREQ 包发送到 M 中的某节点处（数据请求）；二是 I 欲加入 M （加入请求）。针对数据请求的情况，任何具有通往多播组的当前路径的节点都可以发送 RREP 包来回复 RREQ。当前路径的含义与前面讨论过的原始或单播 AODV 是一样的，并且，通往多播组的路径指的是可以到达多播组中任意成员的