

大规模 MANET 路由协议 SPDSR 在 Linux 中的设计与实现

郭一辰, 陈靖, 张黎, 黄聪会¹

(空军工程大学信息与导航学院, 陕西西安, 710077)

摘要 针对 MANET 路由协议测试方法主要采用软件仿真而导致结论不准确的问题, 以大规模 MANET 路由协议 SPDSR 为例, 提出基于 Linux 系统的 MANET 路由协议实现方案, 为构建 MANET 路由协议的真实物理测试方法奠定基础。在分析 Linux 路由体系结构基础上, 应用 Netfilter 框架技术和 TUN/TAP 隧道技术, 实现数据分组的拦截和用户空间缓存功能, 启动 SPDSR 路由协议。通过建立路由检查模块, 将路由使用情况传至用户空间, 更新路由表定时器, 避免因路由过期导致的误删除操作, 最终在 Linux 中成功加入 SPDSR 协议。实际组网测试结果表明: SPDSR 协议基于 Linux 的实现方案能够在移动无线自组网中进行正常路由建立和维护, 满足实际应用需要。

关键词 无线移动自组织网络; Linux; 网络过滤器; 动态源路由; 结构化节点; 计算协议

DOI 10.3969/j.issn.1009-3516.2012.04.016

中图分类号 TP393 **文献标识码** A **文章编号** 1009-3516(2012)04-0080-06

无线移动自组织网络(Mobile Ad hoc Network, MANET)是一种不依赖于固定基础设施的无线网络, 其组网方便、快捷, 不受时间和空间的限制^[1]。当前, MANET 技术已经成为一个热门的研究领域, 与之相关的多项技术都得到了人们的广泛关注, 路由协议设计即是其中之一。

及至目前, 人们已经提出了多达 20 种以上的路由协议, 但都不能很好地支持在大规模网络中的应用^[2], 并且新协议只停留在仿真模拟阶段, 实用化进展缓慢。文献[3]中介绍了一种基于结构化 P2P 计算模式的新型 MANET 路由协议 SPDSR; 文献[4]通过仿真验证了该协议在大规模网络中的几个主要性能指标要优于常用协议, 如 DSDV、AODV、DSR。实验结果证明新协议具有良好的应用和推广前景。因此, 本文研究如何使协议脱离仿真平台, 将其移植到物理节点上编译运行。

Linux 是一款开放源码的操作系统, 该系统由许多体积小、性能高的功能模块组成, 不同领域和不同层次的用户可以根据自己的应用需要很容易的对内核进行定制, 实现新的功能。因此, 文章研究将 SPDSR 移植到装有 Linux 操作系统的物理节点(如 PC 机)中, 并详细阐述协议基于 TUN/TAP 隧道技术和包过滤框架 Netfilter 的实现方法。

1 SPDSR 路由协议

SPDSR 是在主流反应式 MANET 路由协议 DSR 的基础上, 通过引入基于 DHT 的分布式命名机制、新型路由表以及一系列优化策略, 在无线移动自组织网络物理拓扑基础上建立一个 P2P 覆盖层网络(P2P Overlay Network)的虚拟逻辑拓扑结构, 有效地建立起一个基于 MANET 架构的完全分布式自组网路由协议^[3]。

¹ 收稿日期: 2012-04-16

作者简介: 郭一辰(1988-), 男, 北京人, 硕士生, 主要从事无线移动自组织网络路由技术研究。

E-mail: ilqwfe@hotmail.com

协议主要由路由发现算法、路由查询算法、路由维护算法和侦听技术、源路由检测机制、PHello 协议等一系列优化策略组成。新路由模型实现了 DSR 和 Chord 算法的结合,并继承了 DSR 所有动态特征和优点。

2 基于 Linux 的 MANET 路由协议的实现方法

2.1 Linux 系统路由体系结构

Linux 操作系统中实现路由功能主要依靠 2 个模块:分组转发功能模块和分组寻路功能模块^[5]。

由于 Linux 系统本身具有完备的分组转发功能,在实现新协议的时候将此模块保留,用于数据分组的收发。分组寻路功能模块既可以在内核空间实现,也可以在用户空间实现。与内核空间相比,在用户空间实现该模块具有开发周期短、代码移植性好、调试方便等优点。因此通过修改分组寻路功能模块并利用 Linux 自带的分组转发功能即可实现新协议。

2.2 路由协议基于 Linux 的设计

2.2.1 设计思路

Linux 路由体系结构是按照有线网络路由协议的工作方式来实现的,这些协议都是主动路由协议。这种体系结构对 MANET 主动路由协议,如 DSDV、WRP 来说可以完全适用,但对于 MANET 按需路由协议,如 DSR、AODV 就会出现^[6]。因为按需的路由协议只保存网络中部分节点的路由信息,按照 Linux 路由体系结构的固有处理机制当内核路由表中查找不到可用路由时将丢弃数据分组,这就使处在用户空间的路由守护程序无法发挥作用。另一方面,用户空间维护一个存储最近被使用路由的路由列表,对应每一个路由表项都有一个定时器,当某条路由被使用时该定时器将被重置,若在定时器规定的时间内该条路由未被使用则将其删除。但用户空间无法获得内核中路由使用情况信息,不能更新用户空间的陈旧路由。

SPDSR 是一种基于 P2P 技术的按需路由协议。因此,针对 Linux 网络系统对数据分组处理的特点,本文利用 Linux 的 TUN/TAP 隧道原理和 Netfilter 框架解决在 Linux 路由体系中实现协议存在的限制。

2.2.2 TUN/TAP 隧道技术

TUN/TAP 设备^[7]可以虚拟一个网络节点,为内核中没有路由的数据分组提供暂时的接收端。TUN/TAP 并不是一个物理设备,他只是为数据分组提供一条通往用户空间的“隧道”。用户空间收到“隧道”传送过来的数据分组后,路由守护程序将其进行缓存并启动路由算法。若查找成功则调用 socket 函数创建原始套接字 raw socket 将数据分组返回至内核;若查找失败则删除缓存中的数据分组。在 Linux 中为使用 TUN/TAP 设备必须包含特定的头文件 linux/if_tun.h,并通过 ifconfig 对该设备进行配置,最后将缺省路由设置为 TUN/TAP 设备接收。

2.2.3 Netfilter 框架

Netfilter 是 Linux 2.4 内核提供的包过滤框架,可以在内核空间非常高效地进行包过滤^[8]。Netfilter 为每种网络协议定义了一套钩子(hook),这些钩子注册在数据包流过协议栈的几个关键点上,内核的任何模块都可以对每种协议的一个或多个钩子进行注册。当数据包流经 Netfilter 框架中一个 hook 点时,该点将调用各模块注册的回调函数,从而实现对数据包的检查、修改、丢弃等操作。

3 SPDSR 协议基于 Linux 的实现方案

针对 Linux 路由体系结构特点,提出 SPDSR 在 Linux 中的具体实现方案见图 1。

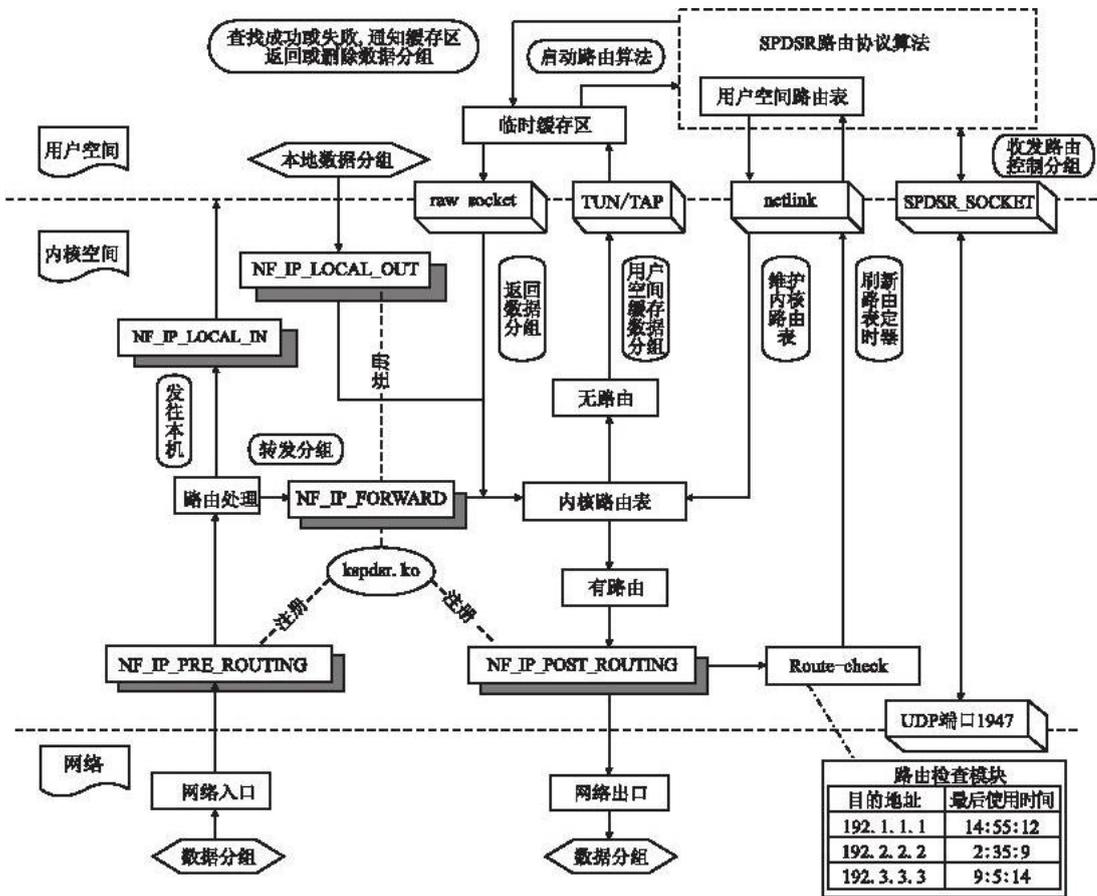


图1 SPDSR在Linux中实现方案

Fig.1 SPDSR implementation in Linux

3.1 对网络数据分组的处理

系统从网络中接收到的数据分组首先要通过 Netfilter 框架的 hook 点 NF_IP_PRE_ROUTING, 在进行校验和、目的地址转换等操作后交由路由代码处理, 判断分组传送方向, 若发往本机则将该分组传至 NF_IP_LOCAL_IN 点进行 INPUT 包过滤后发往上层应用, 否则交由 NF_IP_FORWARD 点处理。kspdsr.ko 是自定义的一个可加载内核模块, 通过将其回调函数在 NF_IP_PRE_ROUTING、NF_IP_LOCAL_OUT、NF_IP_POST_ROUTING 点调用函数 nf_register_hook 注册, 可在数据分组流过时对其进行操作。

通过对数据分组目的地址和内核路由表中的路由进行精确匹配, 判断内核是否包含可用路由。若存在则根据路由信息和规定的网口, 将数据分组传送至 hook 点 NF_IP_POST_ROUTING, 执行源地址转换操作后发送到网络中; 若内核路由表中无可用路由, 调用 open 命令打开设备/dev/net/tun, 同时创建一个结构体 ifr, 设置该结构体中的 ifr_name 即设备名称为 tun_spdsr, ifr_flags 即网络设备属性为 IFF_TUN|IFF_NO_PI。该设备将数据分组发送至用户空间中的临时缓存区进行存储, 并以目的 ip 为关键字映射到散列表(hash)中, 供之后的查询操作。

3.2 SPDSR 路由功能设计与实现

临时缓存区收到 TUN/TAP 传送的数据分组后通知路由守护程序启动 SPDSR 路由查询过程, 发送路由查询请求 RLOC。RLOC 的数据结构为:

```

struct spdsr_rloc_opt {
    u_int8_t type; // 分组类型
    u_int8_t length; // 分组数据长度
    u_int8_t id; // 分组识别码
    u_int8_t max_hop; // 最大查询次数
    u_int32_t source_nid; // 源节点身份标识符
    u_int32_t target_nid; // 目的节点身份标识符
    u_int32_t addrs[0]; // 节点身份标识符列表

```

};

SPDSR 路由协议由多个功能模块组成,其核心模块为 kspdsr, ChordCycle, PRTable 等。各模块主要功能见表 1。

表 1 SPDSR 主要功能模块
Tab.1 SPDSR main function module

子模块名称	功能描述
kspdsr	内核可加载模块,其回调函数在 Netfilter hook 点注册,控制数据分组的处理流程
ChordCycle	Chord 环维护模块
PRTable	SPDSR 路由表维护模块
HASH	将节点映射到 Chord 环的 HASH 构造模块
spdsr_rreq	路由发现消息处理模块
spdsr_rrep	路由回复消息处理模块
spdsr_rloc	路由查询消息处理模块
spdsr_rerr	路由错误消息处理模块
spdsr_chord	Chord 维护模块(控制环内节点加入和退出)
spdsr_socket	SPDSR 控制信息收发模块
spdsr_timer	SPDSR 定时器处理模块

本文定义了一种套接字 SPDSR_SOCKET,调用 bind 命令与 UDP 端口绑定,端口号为 1947,实现协议路由控制分组的收发^[9]。路由查询请求 RLOC 在网络节点组成的 Chord 环上查询所需路由,当达到最大查询数 $\log(N)$ 时结束查询,并告知查询发起节点查询失败。查询发起节点收到查询失败消息后启动路由发现过程,在全网范围进行泛洪查找。若查找失败则删除临时缓存区内的数据分组,释放存储空间;若查找成功则通过 Netlink Socket 将新路由发至内核空间,完成对内核路由表的修改,同时调用 raw socket 将数据分组返回至内核空间进行发送。

3.3 Netlink Socket 实现内核/用户空间通信

在 Linux 中用户空间使用标准的 socket APIs, socket(), bind(), sendmsg(), recvmsg() 和 close() 就能很容易地使用 Netlink Socket,方便地实现用户空间对内核路由表项的维护操作。它提供了内核/用户空间的双向通道。内核接收到路由维护命令帧后,调用相应的路由维护模块执行命令帧指定的操作。用户空间创建 Netlink Socket 的方式为: socket(AF_NETLINK, SOCK_RAW, NETLINK_GENERIC)

SPDSR 为路由表中的每条路由选项设定一个定时器,在定时器规定的时限内若该条路由未被使用则删除该路由选项,以保证路由表的时效性和高效查询。数据分组是根据内核路由表进行发送的,用户空间无法获得路由使用情况,因此设计在 Netfilter 的 hook 点 NF_IP_POST_ROUTING 注册内核可加载模块 route_check,当数据分组经过该 hook 点时,route_check 查看数据分组的首部,更新相应的时间戳,并通过 Netlink Socket 将内核路由使用情况发送到用户空间,守护进程对路由表中的对应路由条目定时器进行归零处理。这样,用户空间就能实时监控内核中路由的使用,并对失效路由进行清除。不同于用户空间,内核空间使用 Netlink Socket 需要专门的 API,内核空间创建 Netlink Socket 的方式为:

```
struct sock *
netlink_kernel_create(int unit, void (*input)(struct sock * sk, int len));
```

并在将路由表使用情况发送完成后释放该 Netlink Socket:

```
void sock_release(struct socket * sock);
```

本地产生的数据分组经过 hook 点 NF_IP_LOCAL_OUT 处理后交由内核路由表进行路由表项查找,若存在可用路由则经过 hook 点 NF_IP_POST_ROUTING 由不同的网路接口发送到网络中;若不存在则将数据分组传至用户空间缓存区,继而发起路由过程。

4 网络测试

4.1 测试场景建立

SPDSR 协议测试网络中有 7 台安装 ubuntu6.06(Kernel Edition 2.6.15)操作系统的 PC 机模拟网络节点,

并开启系统数据转发功能。每个节点配备一块 H3C Aolynk 的 WUB320g 无线网卡,网络拓扑结构如图 2 所示,其中两条链路间由于距离及障碍物等因素,均处在对方通信范围之外,不会发生串路现象。

网络中各节点参数设置见表 2。

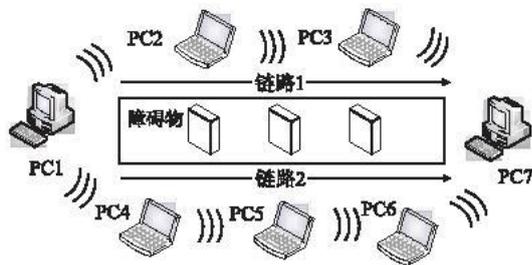


图 2 网络拓扑结构

Fig. 2 Network topology structure

表 2 节点参数设置

Tab. 2 Node parameter setting

节点	IP 地址	节点类型	运动速度/($\text{m} \cdot \text{s}^{-1}$)	位置
PC1	192.168.1.1	静止	0	室内
PC2 - PC6	192.168.1.2 - 192.168.1.6	移动(单兵)	0.7	室外
PC7	192.168.1.7	静止	0	室内

4.2 连通性及路由发现过程测试

PC1 和 PC2 之间定期发送 HELLO 消息建立到对方的路由:

hello - start: Starting to send HELLOs!

确认互为邻居节点,并在路由表中建立邻居节点路由。节点 PC1“ping”节点 PC2 并返回信息,说明两节点间保持联通。

以 PC1 为源节点,PC7 为目的节点,采用最常用的 ping 命令验证多跳路由建立情况。节点 PC1 在最初状态路由表中不包含到达 PC7 的路由,需要发送 RREQ 经中间节点转发至目的节点;PC7 收到 RREQ 后建立 RREP 消息并返回给 PC1。由于链路 1 经历跳数少,最早收到 PC7 发回的 RREP 消息,因此最先建立路由 PC1 - PC2 - PC3 - PC7。表 3 列出了以 PC1 为源节点,不同跳数对应的各项性能指标情况。

表 3 不同跳数条件下通信性能

Tab. 3 Communication performance with different hops

目的节点	跳数	路由发现时间/ms	端到端延迟/ms
PC2	1	0	9
PC3	2	14	22
PC7	3	21	38

关闭 PC3 电源,导致 PC2 无法收到来自 PC3 的 ACK 信息,故判断链路 1 断裂,同时建立 RERR 消息返回至源节点 PC1。PC1 收到出错消息需删除错误路由并重新启动路由发现过程,构建新的路由 PC1 - PC4 - PC5 - PC6 - PC7,将链路 1 上的通信切换到链路 2 上来,完成路由维护过程。PC1 对路由表的操作过程如表 4 所示,其中删除线表示 PC1 将该条路由删除,下划线表示新增路由。

表 4 PC1 对路由的维护操作

Tab. 4 Operation in maintaining route for PC1

目的节点	目的节点序号	下一跳地址	跳数
192.168.1.2	91	192.168.1.2	1
192.168.1.3	92	192.168.1.2	2
192.168.1.7	96	192.168.1.2	3
192.168.1.4	93	192.168.1.4	1
192.168.1.7	96	192.168.1.4	4

5 结束语

本文详细阐述了在 Linux 操作系统中实现按需路由协议的方法,通过设计可加载内核模块 kspdsr.ko 和用户空间 SPDSR 路由守护程序,分别实现了对进入内核数据分组的控制和路由协议控制分组的各项处理操作,最终在 Linux 系统中实现了大规模 MANET 路由协议 SPDSR。本文将一种基于 P2P 的 MANET 路由协议

移植到装有 Linux 系统的 PC 机上,使其能够脱离仿真工具独立运行,最后组建小型网络对实现结果进行了路由功能测试,为协议推广应用奠定了基础。

在实验中发现,测试过程存在网络节点分布范围广、移动场景无法重复、外界条件干扰大、实验过程成本较高等限制。文章下一步研究方向是设计实现一种新型的测试方法,其目标是为 MANET 协议测试提供一种节点易管理、过程可重复、低成本、支持大规模网络且可在实验台上进行的实现技术方案,为之后对协议性能进行更合理准确的测试提供支持。

参考文献(References):

- [1] 齐卫宁,于宏毅,栾玉洁. 自组网路由协议综述[J]. 计算机应用,2005,25(3):511-514.
QI Weining, YU Hongyi, LUAN Yujie. Overview of Ad hoc routing protocols[J]. Computer applications, 2005, 25(3):511-514. (in Chinese)
- [2] 陈靖,罗樵,黄聪会,等. 基于 DSR 的位置预测分簇算法研究[J]. 空军工程大学学报:自然科学版,2011,12(1):55-58.
CHEN Jing, LUO Qiao, HUANG Conghui, et al. The research on clustering algorithm of position forecast based on DSR[J]. Journal of air force engineering university: natural science edition, 2011, 12(1):55-58. (in Chinese)
- [3] 李祖鹏. 基于 P2P 计算模式的 MANET 路由协议研究[D]. 北京:中国科学院计算技术研究所,2007.
LI Zupeng. MANET routing protocol research base on P2P computing[D]. Beijing: Chinese academy of sciences, institute of computing technology, 2007. (in Chinese)
- [4] 郭一辰,陈靖,罗樵,等. 大规模 MANET 路由协议 SPDSR 的仿真研究[J]. 计算机工程与设计,2012,32(6):2118-2122.
GUO Yichen, CHEN Jing, LUO Qiao, et al. Simulation research of the large scale MANET routing protocol SPDSR[J]. Computer engineering and design, 2012. (in Chinese)
- [5] Daniel P bovet, Marco Cesati. 深入理解 Linux 内核[M]. 北京:中国电力出版社,2007:557-591.
Daniel P bovet, Marco Cesati. Understanding the Linux kernel[M]. Beijing: China electric power press, 2007:557-591. (in Chinese)
- [6] 刘焕敏. Linux 平台 Ad hoc 网络按需路由协议实现技术研究[D]. 长沙:国防科技大学,2005.
LIU Huanmin. Research on Implementation of Ad hoc on-demand routing protocols on Linux[D]. Changsha: National university of defense technology, 2005. (in Chinese)
- [7] Tun/Tap universal driver [EB/OL]. (2005-09)[2012-04] <http://vtun.sourceforge.net/tun>.
- [8] Netfilter/Iptables homepage[EB/OL]. (2005-09)[2012-04] <http://www.netfilter.org>.
- [9] 殷肖川,姬伟锋,陈靖,等. 网络编程与开发技术[M]. 西安:西安交通大学出版社,2009:57-64.
YIN Xiaochuan, JI Weifeng, CHEN Jing, et al. Network program and develop technology[M]. Xi'an: Xi'an jiaotong university press, 2009:57-64. (in Chinese)

(编辑:徐楠楠)

The Design and Implement of the Large Scale MANET Routing Protocol SPDSR under Linux

GUO Yi - chen , CHEN Jing , ZHANG Li , HUANG Cong - hui

(School of Information and Navigation, Air Force Engineering University, Xi'an 710077, China)

Abstract: Aimed at the problem that MANET routing protocol could be only simulated in software which lead to a inaccurate result, and taking a large scale MANET routing protocol SPDSR for example, a method of implementing MANET routing protocols in Linux OS is put forward. This method lays a foundation for building a practical MANET test. Based on the study in Linux original routing architecture, by using net-filter framework and TUN/TAP technology the functions of packets intercepting and user space buffering are achieved, then the SPDSR routing algorithm is started. With establishing a route check module, which delivers route status to user space through and updates route table timer, a faulting delete operating for past routes will be avoided. Finally, the SPDSR protocol has been successfully added in Linux. Practical network test result shows that the implementation of SPDSR based on Linux is able to build and maintain route in a wireless mobile ad hoc net-work, as well as meet the needs of engineering application.

Key words: Ad Hoc; Linux; net-filter; SPDSR