

基于移动 IPv6 的流切换协议的设计与实现

王剑白, 陈 峰, 楼亦华

(北京航空航天大学软件开发环境国家重点实验室, 北京 100083)

摘 要: 在移动 IPv6 的基础上设计并实现了一种面向流的切换协议。该协议增加了 2 个消息(流绑定更新、流绑定应答消息)和流绑定更新列表。系统通过修改 Linux 内核网络部分的数据处理流程和数据结构, 实现了移动切换过程中数据流灵活的重定向功能, 为上层的智能切换策略提供更多功能。通过实验仿真能够看出采用流切换的方式, 可以提高传输效率, 更有效地利用网络资源。

关键词: 无线异构网络; 移动 IPv6; 垂直切换; 流切换协议

Design and Implementation of Mobile IPv6 Based Per-flow Handover Protocol

WANG Jian-bai, CHEN Feng, LOU Yi-hua

(State key Lab of Software Development Environment, Beihang University, Beijing 100083)

【Abstract】 This paper designs and implements a Per-flow handover protocol based on mobile IPv6. It adds two messages(flow binding update, flow binding acknowledgement) and a flow binding list to mobile IPv6, and implements flexible data flow redirection by modifying Linux network kernel modules. The protocol provides more functions for intelligent handover policies of high layers. The experiments demonstrate that the Per-flow handover improves the performance of transmission and uses the network resource more efficiently.

【Key words】 wireless heterogeneous network; mobile IPv6; vertical handover; Per-flow handover protocol

近年来无线通信技术发展迅猛, 各种无线接入技术之间的覆盖范围和带宽都存在很大差异, 但是没有一种无线技术能够完全满足随时随地高带宽的接入需求^[1], 因此, UMTS, GPRS, CDMA, WLAN, WiMAX和卫星网络等多种无线技术共存和互通成为必然。许多国际标准化组织都在致力于解决异构网络的融合问题, IEEE已经成立了 802.21^[2]工作组来研究异构网络中介质无关性切换方法; 3GPP开始制定 3G与 WLAN的互通规范; 国际电信联盟(ITU)规划的第四代移动通信(4G)的目标也是对现存无线网的集成和融合并开发新的无线接口技术, 以满足面向多媒体的无线通信。

1 相关研究工作

移动切换是移动通信系统最重要的功能之一, 同构网络中把移动节点(Mobile Node, MN)在小区间的切换统称为水平切换, 而在异构网络中的切换称之为垂直切换^[1]。下一代移动通信网络融合了多种不同的无线通信技术, 因此, 在MN移动的过程中, 不仅要进行水平切换, 还需要根据不同网络的情况, 择优选择合适的接入网络进行垂直切换。垂直切换过程复杂, 要求低时延, 低功耗, 占用尽可能小带宽, 对无线接入技术、信号检测、信道分配和无线资源优化管理都提出了更高要求^[3]。

假设具有多制式网络接入能力的 MN 通过 N 个网络接口, 与 M 个通信对端(Correspond Node, CN)进行通信, 其中 MN 与单个 CN 之间数据流还不止一条, 则根据切换后不同的数据流的重定向方式, 垂直切换可以分为以下 3 种方式:

(1) 切换过程将所有的数据流从旧网络接口 I_1 全部重新定向到新网络接口 I_2 称为“ All-flow ”方式;

(2) 切换过程将某一个或者某一些 CN 的数据流重新定向

到新接口称为“ Per-CN ”方式;

(3) 切换过程将特定数据流的数据重新定向到新接口, 其他数据流保持不变, 称为“ Per-flow ”方式。

针对垂直切换的数据流重定向的管理, 法国电信和 Cisco 于 2005 年 7 月提出“ mobile IPv6 for multiple interfaces ”的草案, 论述了如何通过修改移动 IPv6 协议实现 All-flow 方式垂直切换。韩国 Ewha 大学通过修改 stcp 协议实现 Per-flow 方式的支持, 但该方案同一时刻只有一个接口处于使用状态, 另外的接口是作为备份, 无法利用多接口的优势。

移动 IPv6 技术支持移动终端使用不变的 IPv6 地址在不同的网络内进行持续通信, 对于上层应用程序透明, 被广泛认为是解决终端移动性问题的较好支撑技术。本文是在芬兰 HUT 大学实现的移动 IPv6 协议栈基础上设计并实现的支持 Per-flow 方式的切换管理系统, 同时保证了数据流在切换时灵活的重定向和对上层应用程序的透明。

2 Linux 网络内核分析

本文的实现基于 Linux, 为保证数据流能按要求重定向到相应的网络接口, 需对 Linux 内核的网络部分进行修改。

2.1 Linux 内核网络模块及关键数据结构

Linux 的网络系统可以分为硬件层/数据链路层、IP 层、INET Socket 层、BSD Socket 层和应用层 5 部分。其中, 在 Linux 包含了前面 4 部分, INET Socket 层的实现比 IP 协议层次高, 实现对 IP 分组排序、控制网络系统效率等功能, 而 IP

基金项目: 国家自然科学基金资助项目(60473084)

作者简介: 王剑白(1980 -), 男, 博士研究生, 主研方向: 计算机网络; 陈 峰, 博士研究生; 楼亦华, 硕士研究生

收稿日期: 2007-01-15 **E-mail:** wjb@nlsde.buaa.edu.cn

层是 TCP/IP 网络协议栈互联网层的实现。包括了硬件驱动和硬件发送组织工作的层次为硬件层。

应用层中的操作对象是 socket 文件描述符,通过文件系统定义的通用接口,使用系统调用从用户空间切换到内核空间。在 BSD Socket 层中,操作的对象是 socket 结构,每一个这样的结构对应的是一个网络连接。在 INET Socket 层中,操作对象是 sock 类型的数据,而数据存放在 sk_buff 结构中。sk_buff 数据结构是在 INET Socket 层以及以下层次中存放数据的结构^[4],不同层次之间的数据包的传递就是以 sk_buff 数据结构为载体完成操作的。在该结构中,包含 dst_entry 结构指针,存放了网络路径中下一台主机的地址。在数据包发送过程中,通过调用 dst_entry 结构中的 output 函数指针,向硬件层传递发送的数据包。

2.2 Netfilter 技术及数据包处理流程

Netfilter 是自 2.2 版本内核后 Linux 网络内核重要的组成部分,是 Linux 网络防火墙以及 IPtable 实现的基础。在 Netfilter 中,协议栈每种协议都定义了若干个钩子,如 IPv6 和 IPv4 均定义了 5 个钩子,而对应协议的数据包将按照一定的规则通过若干个钩子,每一个钩子都是处理函数挂载点。内核模块则可以在各个钩子上注册处理函数,以操作经过对应钩子的数据包。函数处理后,根据一定的策略返回内核进行下一步的处理。

Linux 内核中的数据包处理流程如图 1 所示,如 Netfilter 定义了位于 POST_ROUTING 位置的钩子函数,则先进行钩子函数的处理,之后数据包被传递给函数 ip6_output_finish。最后数据包依次通过 neigh_connected_output, dev_queue_xmit, hard_start_xmit 函数发送到网络。

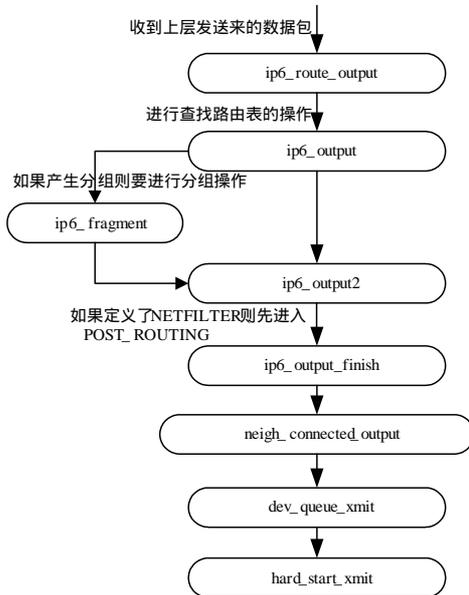


图 1 Linux 内核数据包处理流程

因此,重定向数据流时,系统需要查找路由表获得相应的 dst_entry 指针,通过挂载在 NF_IP_POST_ROUTING 位置的钩子函数截获数据包,使用新的 dst_entry 结构,就可以将数据包从新接口发送出去,实现将数据流重新定向到新的网络接口的功能。

3 系统框架及实现

为了实现对于 Per-flow 方式切换的支持,本文设计并实现了移动切换管理系统,分别对 MN 和 CN 进行了修改。如

图 2 所示, MN 包括用户空间进程和内核模块。其中,用户空间进程读入用户的配置并接受用户输入的指令;内核空间的 用户管理接口负责接受用户空间的命令。消息管理模块处理用户发来的消息或者网络消息,地址管理模块维护流绑定更新列表(FBUL),路由管理模块根据 FBUL 中的信息决定数据包的发送接口。IP 包接收模块过滤符合 FBUL 要求的数据包或者网络消息,将符合要求的数据包发送给 IP 包处理及路由控制模块。IP 包处理及路由控制模块完成 Per-flow 方式的路由,最后调用 IP 包发送模块将数据包发送出去。对于 CN 来说,与 MN 不同的地方在于 CN 没有路由管理模块,其他模块功能与 MN 相同。

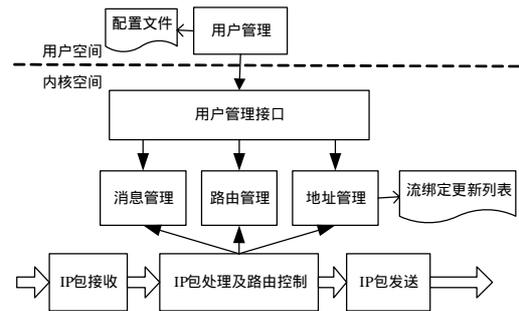


图 2 MN 侧移动切换管理系统结构

系统的工作流程如下:

(1) MN 以新接口的地址作为 MN 的转交地址(MNCoA)发送流绑定更新(FBU)消息到 CN, FBU 消息中包含了 MN 的家乡地址(MNHoA)以及即将发生切换的流的信息。

(2) CN 将 FBU 消息中的流信息加入到本机的 FBUL 中,并回复流绑定应答消息(FBA)到 MN。

(3) MN 收到 FBA 消息后将流信息加入到本机的 FBUL 中。

(4) CN 对于收到的每个数据包,如果数据包的源地址为 MNCoA,则将源地址替换为 MNHoA,并交给上层处理,否则直接交给上层处理。

(5) CN 对于每个发往 MNHoA 的数据包,查询 FBUL 中的信息,如果数据包的流信息与 FBUL 中的某项信息匹配,则把目的地址替换为 MNCoA,并交给下层处理,否则直接交给下层处理。

(6) MN 对于每个发往 CN 地址(CNAddr)的数据包,查询 FBUL 中的信息,如果数据包的流信息与 FBUL 中的某项信息匹配,则把源地址替换为 MNCoA,并交给下层处理,否则直接交给下层处理,最后由路由管理模块选择相应的接口将数据包发送出去。

(7) MN 对于每个从 CNAddr 发来的数据包,查询 FBUL 中的信息,如果数据包的流信息与 FBUL 中的某项信息匹配,则把目的地址替换为 MNHoA,并交给上层处理,否则直接交给上层处理。

FBUL 的结构中包含了以下几部分内容: MNHoA, MNCoA, CNAddr, MNPort, CNPort 传输层协议类型。其中, MNHoA 项作为 MN 的标识,同一个 MN 可以有多个 FBUL 项,每项代表了一个离开 MN 家乡网络的流。

FBU 消息的格式如图 3 所示。MN 以自己的转交地址为源地址,以 CNAddr 为目的地址发送 FBU 消息给相应的 CN,并将自己的家乡地址放入消息的家乡地址选项中。消息中的生存时间是以 4 s 为单位的时间计数,用于告知 CN 本地

FBUL 中该流信息的生存时间。消息序号是用来唯一标识一条 FBU 消息及其应答的整数,对于每一个 FBU/FBA 消息对,具有相同的消息序号。

扩展报头 协议	扩展报头 长度	信令类型	保留
校验和		消息序号	
生存时间		MN 端口号	
CN 端口号	传输层协议类型	保留	

图 3 FBU 消息定义

FBA 消息的格式如图 4 所示。CN 在收到 MN 发来的 FBU 消息后,更新本地的 FBUL,并以相同的消息序号返回 FBA 消息给 MN。消息中的生存时间是以 4 s 为单位的时间计数,用于告知 MN 本地 FBUL 中该流信息的生存时间。状态字段为 0 表示该次流绑定更新成功,否则表示该次流绑定更新中发生的错误号。

扩展报头 协议	扩展报头 长度	信令类型	保留
校验和		消息序号	
生存时间		状态	保留

图 4 FBA 消息定义

路由管理模块在实现的时候,本系统采用 Linux 内核的 Netfilter 机制获取符合 FBUL 中某项信息的数据包的 sk_buff 结构,通过修改 sk_buff 结构指向 dst_entry 结构的 dst 指针实现。其中关键在于寻找合适的 dst_entry。目前 Linux 内核在查找 dst_entry 时,根据目的地址在路由缓存中进行查找。因此为了获得新的 dst_entry,需要用新的目的地址进行查找。将每个接口的网关地址配置在配置文件中,通过使用每个接口对应的网关地址作为 dst_entry 的查询条件,可以获得每个接口对应的 dst_entry。系统查找路由表获得相应的 dst_entry 指针之后的数据包,NF_IP_POST_ROUTING 位置的钩子函数截获该数据包,然后使用新的 dst_entry 结构,可以将数据包从新接口发送出去。

4 实验仿真与分析

为了验证该 Per-flow 移动切换协议的性能,本文使用 NS2 进行了实验仿真。假设 MN 同时具备 WLAN 和 UMTS 2 种无线接入能力,WLAN(802.11b)网络带宽 11 Mb/s,延时 10 ms;UMTS 网络带宽 384 Kb/s,延时 70 ms。本实验使用 CBR(UDP)流模拟实时数据传输流 Flow1;使用 FTP(TCP)流模拟非实时数据传输流 Flow2。

实验场景 1 的初始状态为 2 条数据流均通过 WLAN 网络传输,而此时 Flow1 数据量猛增产生拥塞的情况,对比 Flow2 流从 WLAN 网络切换到 UMTS 网络后的效果。其中,CBR 流速率为 10.96 Mb/s,包大小为 210 B;FTP 流包大小为 210 B。

由图 5、图 6 可以看出,Flow2 不切换到 UMTS 网络,Flow1 传输速率的抖动比较大,而 Flow2 由于无法得到带宽资源,传输速率一直维持在 60 Kb/s 附近波动;而 Flow2 如果切换到 UMTS 网络,Flow1 传输速率明显提高且抖动比较小,Flow2 由于切换到新的网络,传输速率也显著提高。

由图 7、图 8 可以看出:(1)2 条数据流都通过 UMTS 网络传输,不做任何切换,则 Flow1 的抖动非常明显,传输很不稳定,而 Flow2 只能保持在非常低的速率进行传输;(2)2 条数据流都切换到 WLAN 中,Flow1 和 Flow2 的传输速率都得到了很大的提高,但仍然有明显的抖动;(3)仅将非实时的 Flow2 切换到 WLAN,而实时流 Flow1 维持在 UMTS 网络中

传输,则 Flow1 和 Flow2 都可以获得较高的传输速率,且抖动很小。

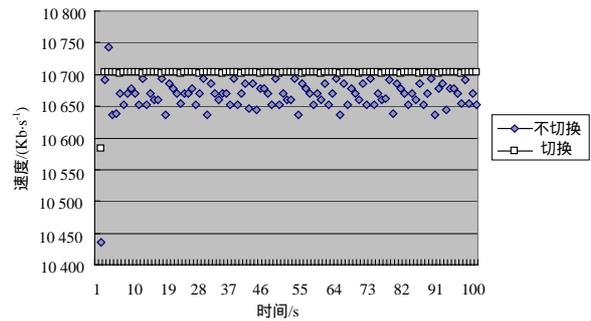


图 5 Flow2 切换与否对 Flow1 的影响对比

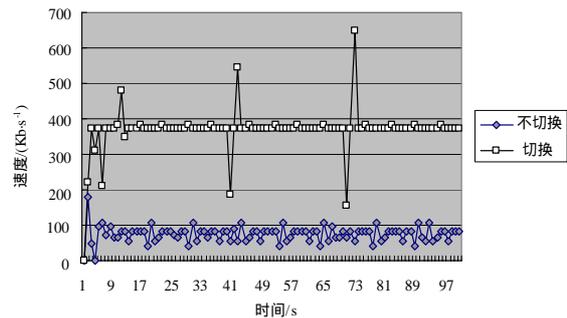


图 6 Flow2 切换与否对自身的影响对比

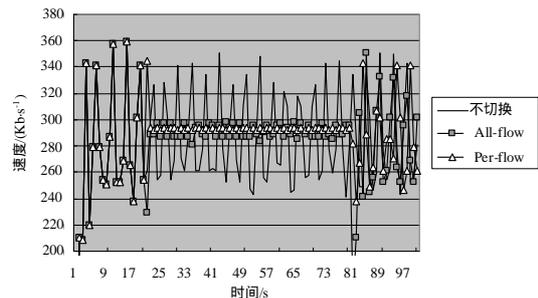


图 7 Flow1 在 3 种切换策略下的速率比较

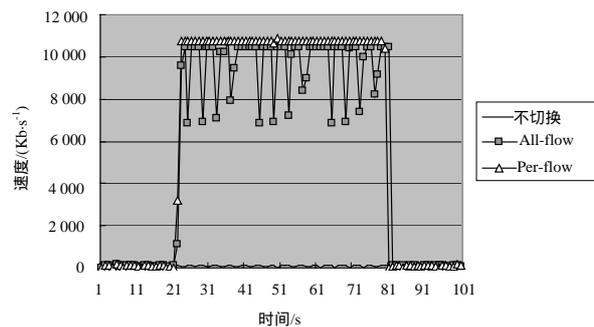


图 8 Flow2 在 3 种切换策略下的速率比较

通过上面的比较,可以看出流切换的方式为数据流提供了灵活的传输支持,能够充分应用网络的有效资源,提高传输效率。

5 结束语

本文设计并实现了一种基于移动 IPv6 的流切换协议。该协议在移动 IPv6 的基础上增加了流绑定更新消息和流绑定应答消息,在 MN 和 CN 的协议实现实体中增加了流绑定更新列表 FBUL,从而实现了数据流级别的移动切换绑定。系

(下转第 113 页)