

WLAN 环境下 MIPv6 快速切换协议的分析与实现^①

Analysis and Implementation of Fast Handover for
Mobile IPv6 under WLAN

熊继平 (浙江师范大学 数理与信息工程学院通信工程系 浙江金华 321004)

彭永超 (中兴通信网络事业部 南京 210000)

摘要: 本文针对 802.11 WLAN 环境给出了能够有效减少移动 IPv6 快速切换延迟的关键因素,描述了在实现该协议的过程中遇到的几个关键技术难点问题,提出并实现了相应的有效解决方案。本论文给出的设计以及实现对于 WLAN 环境下的快速移动 IPv6 协议标准的制定以及实现有着较好的参考价值。

关键词: 移动 IPv6 快速切换 无线局域网 切换延时

目前,IPv6 作为下一代因特网协议,已经开始进入大规模实用化阶段。与此同时,各种移动设备也在迅速普及,并成为 IPv6 的主要应用对象之一。为了保证在移动设备跨子网切换的过程中,会话不会因设备的网络地址改变而中断,IETF 制定了移动 IPv6 协议^[1]。但是该协议存在一个比较严重的缺点,就是切换延迟比较大,通常为几秒钟,不能满足实时业务(如 VoIP、网络视频)的需要。IETF 目前正在制定各种切换优化协议,用于减少移动 IPv6 的切换延迟^{[2][3]},快速切换就是其中一种受到学术界和产业界普遍关注的协议。此外 802.11a/b/g 因其无线宽带便捷的特点,已经得到越来越多的关注以及一定规模的部署。本文正是从这两个方面出发,探讨并实现了 WLAN 环境下的快速切换协议。

1 移动 IPv6 快速切换过程

1.1 基本流程介绍

移动 IPv6 快速切换是目前比较成熟的一种切换优化协议。它的基本思想可以概括为两条:

(1) 引入链路层触发器,减少移动检测时延 Tmd。

(2) 通过隧道转发和缓存机制,使移动节点在完成绑定更新过程之前,仍能使用旧转交地址收发报文。

所谓链路层触发器,是指当链路层的连接状况发生变化时,由链路层向上层协议发送的信号。在快速切换中,主要用到两种触发器:LinkPreDown 触发器,即 MN 将要断开同当前的无线接入点的连接时发送的信号;LinkUp 触发器,即 MN 建立了同新的无线接入点的连接后发送的信号。

根据是否可以预测链路层断开的时间,将快速切换分为预测式和反应式两种。另外,根据由哪一方发起快速切换,可以分为由网络发起的快速切换和由移动节点发起的快速切换。图 1 给出了由移动节点发起的预测式快速切换的完整过程。

由移动节点发起的反应式快速切换的过程如图 2 所示。它包括了没有启用 LinkPredown 触发器的情况,以及预测式快速切换过程中,移动节点没有在旧的链路收到 FBACK 消息的情况。

和预测式快速切换相比,反应式快速切换有几点不同:FBU 应该被封装在 FNA 消息内;NAR 收到 FNA 后,如果发现新的转交地址无效或重复,会丢弃 FNA 内

^① 基金项目:浙江师范大学人才引进课题(ZC304006123)

的 FBU, 否则将 FBU 转发到 PAR; NAR 不会缓存报文; PAR 将 FBACK 发送到新的链路, 而不在本地链路上发送。

1.2 切换延迟分析

有了链路层触发器之后, 网络层就可以在第一时间知道移动节点已经连接到新的子网, 移动检测时延

$$T_{pho} = T_{I2}$$

而反应式快速切换的延迟为:

$$T_{rno} = T_{I2} + T_{fbu}$$

其中 T_{I2} 是链路层切换时延, T_{fbu} 是从移动节点向先前的接入路由器发送 FBU, 到收到 FBACK 的时间间隔, 通常是几十毫秒。

由此可见, 快速切换的延迟一般不会超过 100ms, 能够满足实时业务的需要。

2 实现快速切换方案

我们以文献^[2]为基础, 分析和实现了移动 IPv6 快速切换, 并根据实际测试的结果提出并实现了本文提出的高效方案。我们的实现基于源代码开放的 Linux 内核^[4]和移动 IPv6 协议栈^[5]。限于文章篇幅, 我们仅对本文方案涉及到的关键方法做详细的方案论证和分析。

2.1 链路触发器的实现

链路触发器的具体实现方式和无线网卡的型号、驱动程序相关。我们最后选择的是基于 Prism II 芯片组的 802.11b 无线网卡和 linux-wlan-ng 驱动程序^[6]。

无线网卡有两种主要工作模式: 手动模式和漫游模式。在手动模式下, 需要用户手动设置无线链路的参数 (SSID、BSSID、信道), 手动控制切换。在漫游模式下, 无线网卡能自动发现周围的无线接入点 (AP), 自动切换到新的 AP。当用户在大范围内移动时, 应该让无线网卡在漫游模式。下面主要介绍漫游模式下的切换过程。

漫游模式下, 网卡驱动会周期性检查当前信道的质量, 当发现当前信道的质量低于某个阈值 (cell_search_threshold) 时, 开始扫描信道。扫描的结果显示在移动节点周围有哪些 AP 存在, 以及有哪些可用 threshold 的信道。如果新的 AP 的信道质量比当前信道的质量高出某个阈值 (delta_snr), 则准备切换到新的 AP。这时启动 LinkPredown 触发器, 并等待一段时间 (joinreq_delay), 让快速切换的消息交换过程能够完成。然后断开同当前 AP 的连接, 建立同新的无线接入点的连接, 完成后, 启动 LinkUp 触发器。图 3 给出了

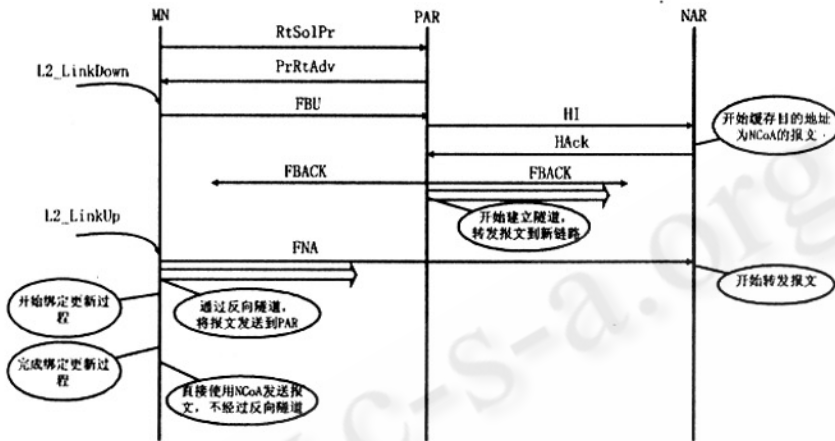


图 1 移动节点发起的预测式快速切换
Fig1. Fast handover of proactive mode

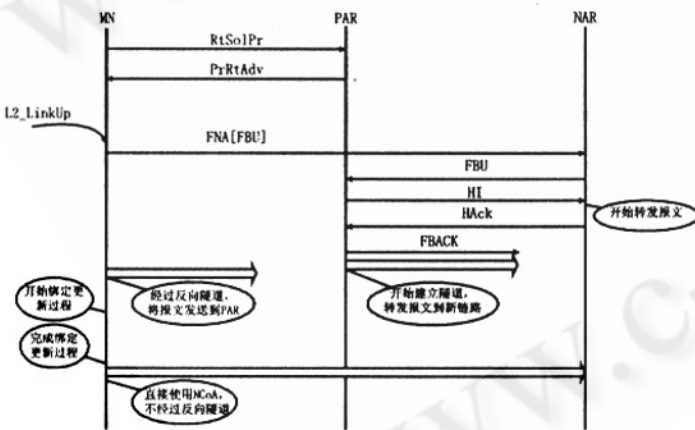


图 2 移动节点发起的反应式快速切换
Fig2. Fast handover of reactive mode

Tmd 就被去掉了。另外, 在快速切换中, 由新的接入路由器进行对新转交地址进行重复检测, 这个工作通常靠直接查询邻居列表完成, 所以地址配置及重复检测时延 Tdad 也可以去掉。由于绑定更新过程完成之前, 移动节点可以通过它同 PAR 间的双向隧道接收和发送报文, 所以绑定更新时延 Tbu 也可以去掉。

这样, 预测式快速切换的延迟为:

整个完整的过程。

需要根据实际的网络状况设置参数 `cell_search_threshold` 和 `delta_snr` 的值,以便切换能够进行。`joinreq_delay` 的值对快速切换的性能有影响:太大则由于 PAR 将报文转发到新链路,移动节点在切换到新链路前不能收到报文,增加了切换延迟;太小则快速切换的消息交换过程不能在旧链路上完成,同样会增加切换延迟。一个可能的改进方法是当快速切换的消息交换过程完成时,由网络层指示链路层切换到新的无线接入点。

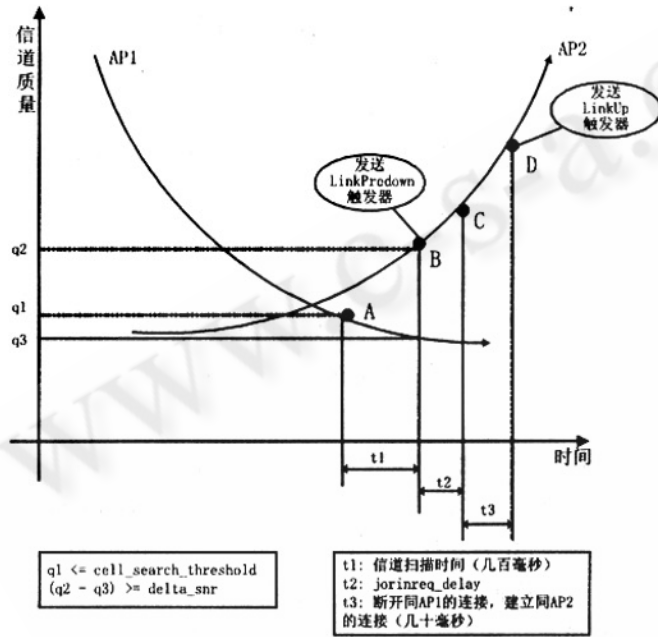


图 3 链路层切换
Fig3. Link layer handover

另外,根据我们的测试结果,完全扫描 14 个信道,需要的时间为 500 ~ 700ms。而且在扫描期间,网卡的发送和接收中断都被禁止,收到的数据帧被放在网卡的接收缓冲区内,等扫描结束后,才会被读取到协议栈的接收缓冲区中。这通常不会造成报文丢失,但会使单个报文的接收延时变大。如何减少信道扫描的时间,是一个值得进一步研究的问题。

手动模式下的切换过程比较简单。用户使用配置工具(如 iwconfig)指定新的无线接入点的 SSID、BSSID、信道号后,网卡驱动就会立刻开始切换到新的无线接入点的过程,完成后,启动 LinkUp 触发器。这种情况下没有 LinkPredown 触发器,只能进行反应式快速

切换。

2.2 报文缓存的实现

如前所述,在预测式快速切换过程中,在移动节点连接到新的链路前,旧的接入路由器可能已经开始通过隧道向新的链路转发报文。为了避免报文在新的链路上被丢弃,新的接入路由器应该为移动节点缓存报文,直到它收到 FNA 消息。

为了实现报文缓存功能,首先创建一个虚拟的网络接口设备 `fmip6buf0`。在 linux 内核中,每个网络接口设备,不管是物理的还是虚拟的,都对应一个 `net_device{}` 结构。这个结构很复杂,但我们只需要关心其中的 `hard_start_xmit` 函数:

```
struct net_device {
    .....
    int (*hard_start_xmit) (struct sk_buff *skb,
        struct net_device *dev);
    .....
};
```

创建缓存时,首先添加一条主机路由,让所有目的地址为移动节点的转交地址的报文都经过设备 `fmip6buf0`,然后为该地址创建一个缓存队列。

缓存过程中,发往移动节点的报文到达新的接入路由器时,根据路由规则,将通过设备 `fmip6buf0` 发送报文。设备 `fmip6buf0` 的 `hard_start_xmit` 函数被调用。该函数的操作过程是:检查是否有和报文的地址匹配的缓存队列,如果有,则将报文加到该队列尾部,否则,丢弃报文。这样,发往移动节点的新转交地址的报文就被缓存了。

停止缓存时,首先删除主机路由。如果已经收到了移动节点的 FNA 消息,则调用 `netif_receive_skb`,重新接收处理并转发缓存队列中的报文。否则,调用 `kfree_skb` 释放缓存队列中的报文。最后,删除缓存队列。

为了避免缓存的报文占用过多的内存空间,我们对缓存的工作时间和缓存队列中报文的个数进行了限制。

2.3 链路稳定性问题

移动节点同无线链路间的连接可能是不稳定的,即移动节点建立了同一条无线链路的连接后,可能会在极短时间内,又切换到另一条无线链路。造成这种

现象的原因可能是移动节点所在区域的信号不稳定,或者它移动得太快。从切换的角度来说,主要有两种表现形式:

(1) 三方切换,即移动节点从无线链路 1 切换到无线接入点 2,又在极短时间内切换到无线接入点 3;

(2) 乒乓现象,即移动节点以极短的时间间隔在两条无线链路间来回切换。

为此,我们引入过渡期的概念:当 MN 建立了同新的链路的连接,且收到 FBACK 之后,不立即开始绑定更新过程,而是等待一段时间(几秒钟),我们称这段等待时间为过渡期。

(1) 如果过渡期结束后,移动节点还在新的链路上,则可以认为移动节点同新链路的连接已经稳定,开始绑定更新过程。

(2) 如果在过渡期内,MN 又发生切换,它会向 PAR 发送 FBU。PAR 将拆除原来的隧道,建立新的隧道。这种情况对应于三方切换。

(3) 如果在过渡期内,MN 又发生切换,且回到了 PAR 的链路,它会向 PAR 发送生存期为 0 的 FBU。PAR 将拆除原来建立的隧道。这种情况对应于错误移动。

通过引入过渡期,我们在链路不稳定的情况下减少了不必要的绑定更新过程,提高了系统工作的稳定性。

3 测试数据

我们组建了一个小规模移动 IPv6 示范网,对移动 IPv6 的切换性能进行了测试,采用的测试方法是:从通信对端向移动节点周期性发送 UDP 数据包,时间间隔为 20ms,通过统计在切换过程中丢失数据包的个数来计算切换的延时。测试中接入路由器发送路由器公告报文的时间间隔为 2~6 秒,绑定更新过程的时间延迟为几十毫秒,可忽略。表 1 所示为连续 10 次切换的数据包丢失情况,其中第 1 行对应基本切换,第 2 行对应预测式快速切换,第 3 行对应反应式快速切换。结果显示,基本切换的延迟为 1~6 秒,预测式快速切换的延迟为 0~40ms,而反应式快速切换的延迟为 20~60ms。

表 1 测试数据(丢包计数)

基本切换	197	258	243	131	233	118	171	270	76	41
预测式	2	2	2	2	1	2	1	1	2	2
反应式	0	0	1	0	1	0	0	1	0	1

4 结束语

本文对移动 IPv6 的基本切换在 802.11 环境下的延迟进行了深入分析,提出了一种快速切换方案,并对快速切换的延迟进行了分析。重点介绍了我们在实现该协议的过程中遇到的几个难点问题和相应的解决方法,通过实际的测试数据证明我们实现的快速切换方案达到了预期的效果。本文的结论对于快速切换协议以及实现有着较大的参考价值。

下一步的研究工作是考虑如何将快速切换协议和其它切换优化协议(例如层次化移动 IPv6)结合,以及如何将快速切换和其它功能模块(QoS、AAA、头标压缩)结合。

参考文献

- 1 D. Johnson, C. Perkins, J. Arkko. Mobility Support in IPv6. RFC3775. June 2004. <http://www.ietf.org/rfc/rfc3775.txt>.
- 2 R. Koodli, Ed. Fast Handovers for Mobile IPv6. RFC4068. July 2005. <http://www.ietf.org/rfc/rfc4068.txt>.
- 3 P. McCann. Mobile IPv6 Fast Handovers for 802.11 Networks. RFC4260. November 2005. <http://www.ietf.org/rfc/rfc4260.txt>.
- 4 <http://www.kernel.org>
- 5 <http://www.mobile-ipv6.org>
- 6 <http://www.linux-wlan.com>