

移动 IPv6 中基于 MLD 的快速重复地址检测^{*})

刘 沙 杨寿保

(中国科学技术大学计算机科学与技术系 合肥 230026)

摘 要 移动 IPv6 尤其是层次化移动 IPv6 技术中,地址重复检测过程严重影响了移动节点的切换速度。这篇论文通过从搭建的层次化移动 IPv6 实验床收集的数据,分析了造成上述性能下降的原因,并据此提出了基于多播侦听者发现机制的快速重复地址检测方案,最后论证了此方案相对于传统方案的性能优势。

关键词 移动 IPv6,层次化移动 IPv6,重复地址检测,绑定更新,多播侦听者发现

MLD-Based Fast DAD for Mobile IPv6

LIU Sha YANG Shou-Bao

(Dept. of Computer Science and Technology, University of Science and Technology of China, Hefei 230026)

Abstract In Mobile IPv6 especially in Hierarchical Mobile IPv6, Mobile Node's handoff performance is greatly impaired by the Duplicate Address Detection procedure. Based on the data gathered from our HMIPv6 test bed, this paper analyzes the origin of such performance degradation, proposes a MLD-Based Fast DAD scheme based on the analysis, and demonstrate that the MLD-FastDAD can perform much better than traditional DAD scheme.

Keywords Mobile IPv6, HMIPv6, DAD, BU, MLD

1 引言

随着网络技术的不断发展,移动通信技术逐步走进了人们的生活。为了在移动技术上承载多媒体业务数据,必须要求移动技术中的切换延时尽可能地小。对于一般的交互应用和多媒体数据流如视频点播而言,所能容忍的通信中断的时间为 10~100ms 级,而对于移动 IP(Mobile IP)而言,通信中断主要是由于移动节点(Mobile Node, MN)进行切换(handoff)所引起的。因此,对于移动 IP 的性能改进关键就在于减少切换延时。

本文主要改进的是移动 IPv6(Mobile IPv6, MIPv6)特别是层次化移动 IPv6(Hierarchical Mobile IPv6, HMIPv6)中 MN 的切换速度,对 MN 检测到移动后形成新的相关转交地址(Care-of Address, CoA)的过程做了很大的优化,从而极大地减轻了此过程对绑定更新的负面影响。全文接下来的组织结构如下:第 2 部分简要介绍了 MIPv6^[4]和 HMIPv6^[5]中 MN 进行绑定更新、形成 CoA 以及进行重复地址检测(Duplicate Address Detection, DAD)的过程;第 3 部分则通过实验,分析了 MN 对新 CoA 进行 DAD 的过程对绑定更新的影响;第 4 部分提出了基于多播侦听者发现(MLD)协议^[3]的快速重复地址检测(MLD-FastDAD)方案,通过分析,论证了该方案远优于传统方案;最后进行了总结。

2 MIPv6 的基本协议过程

2.1 MIPv6^[4]中移动节点的绑定更新

在 MIPv6 中,当 MN 检测到移动接入新的外地网络后,一般是通过接收其接入路由器(Access Router, AR)发送的路由公告(Router Advertisement, RA)来进行无状态自动地址配置^[3],从而生成 CoA。CoA 的配置方法是 MIPv6 与 MIPv4

的主要不同点之一,而正是这种机制带来了新的问题。MN 以此 CoA 向家乡代理(Home Agent, HA)发送绑定更新消息(Binding Update, BU),待接收到相应的绑定确认(Binding Acknowledgement, BA)后通信方可继续。

2.2 HMIPv6^[5]中移动节点的绑定更新

HMIPv6 引入了一个新的功能实体——移动停靠点(Mobile Anchor Point, MAP),其功能可以概括为“本地家乡代理”。MN 在 MAP 的管理域内获得两个相关转交地址——区域相关转交地址(Regional Care-of Address, RCoA)和链路相关转交地址(On-Link Care-of Address, LCoA)。支持 HMIPv6 的 MN 进入此 MAP 管理域内并接入到某 AR 后,接收到此 AR 的 RA,算出计划使用的 LCoA 并立即以此 LCoA 为 CoA 向 HA 进行绑定更新;然后 MN 按照接收到的 RA 中的 MAP 信息生成 RCoA,并向 MAP 注册绑定此 RCoA 与 LCoA,MAP 对其注册成功后返回相应的 BA;紧接着 MN 以 RCoA 为 CoA 向 HA 和 CN 发送绑定更新,通告其新的 CoA 为 RCoA。此后 MAP 将截获发往 RCoA 的数据包,并通过隧道传给 MN 的 LCoA。MN 在此 MAP 管理域内移动并发生链路切换时,仅需向 MAP 注册其新的 LCoA 而 RCoA 不变,从而实现在 MAP 管理域内的移动对该域以外透明,因此大大减少了绑定更新的信令数量和来回的延迟。

2.3 IPv6 无状态自动地址配置^[2]

MN 移动并接入到新的外地网络后,必须形成 CoA(HMIPv6 中为 LCoA 和 RCoA)。一般 CoA 的形成采用的是 IPv6 的无状态自动地址配置协议。MN 在收到相应 AR 第一个 RA 后,将根据其中的路由器地址和前缀长度计算出相应的 CoA 或 LCoA,但 MN 无法确定该链路上是否还有其它主机正在或将要使用此地址,因此必须执行 DAD。为了避免同链路多节点同时启动时立即执行 DAD 可能造成的网络拥

^{*})本课题研究受国家 863 项目“移动 IPv6 的关键技术”支持(项目号:2001AA121041)。刘 沙 硕士研究生,主要研究方向为移动计算。杨寿保 教授,博士生导师,主要研究方向为网络与移动计算、网络安全、网络计算。

塞,MN 将等待一段随机时间(1~1000ms)后才开始 DAD。MN 通过发送一个特殊的邻居请求(Neighbor Solicitation, NS)消息来执行 DAD,其中源地址为零地址(::),目的地址为请求节点组播地址、请求地址为该 CoA。这里请求节点组播地址高 104 位为 ff02::1:ff,低 24 位为 CoA 中网络接口地址部分的低 24 位,MN 同时加入此组播组。若在 DAD 开始后一段时间内(RFC2462 推荐值为 1s)未收到已占用此地址的主机发出对此地址的邻居公告(Neighbor Advertisement, NA)或其它主机对此地址的 DAD,则 DAD 成功,CoA 可以正式开始使用(DAD 多次发送 NS 可以降低某 NS 或其应答意外丢失的风险,RFC2462 推荐发送 NS 次数为 1)。

3 MIPv6 中由 DAD 导致的性能问题

为了研究 MIPv6(包括 HMIPv6)的性能,我们搭建了一个 HMIPv6 实验床,软件环境为 Red Hat Linux 7.3、2.4.18 版本内核、radvd 7.1、MIPL 0.9.3、Monash HMIPv6 0.2 for MIPL 0.9.3。整个实验床网络拓扑如图 1 所示。其中 AR1 和 AR2 同属于 MAP 管理域,分别作为不同子网的网关路由器;NIST Net^[8]模拟实现 MAP 到 HA 之间的线路延时;各节点均配备 PII 以上级别 CPU 和 128MB 内存,并通过 100M LAN 相连。

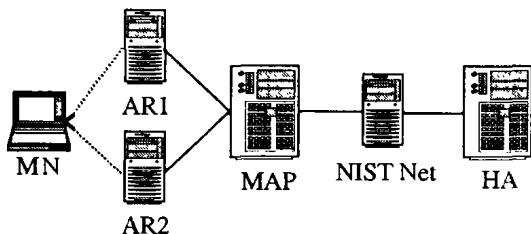


图 1 HMIPv6 试验床拓扑结构

表 1 HMIPv6 实验结果分析:MN 初次接入 AR1

时间	源	目的地	数据包分析
04:03:04.097904	AR1	ff02::1	RA
04:03:04.098415	LCoA-1	HA	BU(0)
04:03:04.200681	AR1		NS(AR1 试图把 BA(0)发送给 LCoA)
04:03:04.266479	::		NS(开始对 LCoA-1 的 DAD)
04:03:05.146773	LCoA-1	MAP	BU(1)
04:03:05.197054	AR1		NS(AR1 试图把 BA(0)发送给 LCoA)
04:03:06.196963	AR1		NS(AR1 试图把 BA(0)和 BA(1)发送给 LCoA)
04:03:06.197022	LCoA-1	AR1	NA
04:03:06.197277	HA	LCoA-1	BA(0)
04:03:06.197368	MAP	LCoA-1	BA(1)(源路由到 RCoA)
04:03:06.656565	LCoA-1	MAP	BU(2)
04:03:06.657149	MAP	LCoA-1	BA(2)(源路由到 RCoA)
04:03:06.657462	RCoA	HA	BU(3) (将 HA 中 CoA 更新为 RCoA)
04:03:06.759031	HA	RCoA	BA(3)(由 MAP 通过隧道传给 LCoA-1)

我们采用的实验方案和参数如下:MAP 到 HA 的来回延迟 RTT=100ms;MN 离开家乡网络,移动并接入到 AR1;经过一段时间后 MN 离开 AR1 所在子网链路,接入到 AR2;再过一段时间后,MN 离开 AR2 所在子网链路,接入到 AR1。

我们利用 tcpdump 对各个节点的网络事件进行了记录,并在表 1~3 中给出了 MN 在某 MAP 域中移动时所经历的最重要的一些事件(经过多次实验我们均得到类似的结果)。表中的绑定更新过程分为两类,到 HA 的和到 MAP 的,预期延时分别为 100ms 和 1ms 左右。

从表 1 可以看出当 MN 初次接入 AR1 时,从发送 BU(0)到接收 BA(0)用了约 2.1s,发送 BU(1)到接收 BA(1)耗时约 1.05s,都远远超过了预期值。而此后 MN 从发送 BU(2)到接收 BA(2)仅用时不到 1ms,从发送 BU(3)到接收 BA(3)耗时约 102ms,这都符合预期。

同样,由表 2 可见,MN 离开 AR1 链路并初次接入 AR2 时也有类似现象——从发送 BU(4)到接收 BA(4)耗时为 2.0s,接收 BU(5)的确认也耗费了不正常的约 491ms。

表 2 HMIPv6 实验结果分析:MN 离开 AR1 初次接入 AR2

时间	源	目的地	数据包分析
04:03:44.058099	AR2		RA
04:03:44.058695	LCoA-2	MAP	BU(4)
04:03:44.060788	AR2		NS(AR2 试图把 BA(4)发送给 LCoA)
04:03:44.666492	::		NS(开始对 LCoA-2 的 DAD)
04:03:45.057049	AR2		NS(AR2 试图把 BA(4)发送给 LCoA)
04:03:45.566577	LCoA-2	MAP	BU(5)
04:03:46.056979	AR2		NS(AR2 试图把 BA(4)发送给 LCoA)
04:03:46.057068	LCoA-2	AR2	NA
04:03:46.057353	MAP	LCoA-2	BA(4)(源路由到 RCoA)
04:03:46.057450	MAP	LCoA-2	BA(5)(源路由到 RCoA)

但是当 MN 离开 AR2 链路接回 AR1 时(表 3),从发出 BU 到接收 BA 的延时则完全符合预期:向 MAP 进行的绑定更新用了 1.2ms 左右(BU/BA(12)),而向 HA 进行的绑定更新则耗时 101ms(BU/BA(13))。

表 3 HMIPv6 实验结果分析:MN 在其 LCoA-1 生命期结束前离开 AR2 链路并再次接入 AR1

时间	源	目的地	数据包分析
04:04:46.170029	AR1		RA
04:04:46.170480	LCoA-1	MAP	BU(12)
04:04:46.171409	AR1		NS(AR1 试图把 BA(12)发送给 LCoA)
04:04:46.171472	LCoA-1	AR1	NA
04:04:46.171727	MAP	LCoA-1	BA(12)(源路由到 RCoA)
04:04:46.171830	RCoA	HA	BU(13)
04:04:46.272981	HA	RCoA	BA(13)(由 MAP 通过隧道传给 LCoA-1)

对于上述异常现象的解释可以通过对 MN 第一次接入 AR1 时的网络事件时序分析得到:

1. 首先,MN 接收到 AR1 发出的 RA,从而算出计划采用的 LCoA-1,并标记为临时地址(Tentative)^[2];

2. MN 以此临时地址向 HA 发送 BU(0),约 100ms 后 BA(1)到达了 AR1(通过对 AR1 的网络事件分析得到),因此 AR1 进行 NS,请求 LCoA-1 的链路层地址。由于此时对 LCoA-1 的 DAD 尚未开始,因此这次 NS 在超时时限内(1s)

得不到响应;

3. MN 在发送 BU(0)后等待了约 168ms(随机产生)才开始对 LCoA-1 的 DAD;

4. AR1 在等待第一次 NS 的应答约 1s 后超时,再次发出对 LCoA-1 的 NS,而此时相应的 DAD 尚未完成,因此此次 NS 仍然得不到响应,BA(0)仍然滞留在 AR1 中;

5. 约过 68ms 后 MN 对 LCoA-1 的 DAD 完成,但并不立即发送 NA;

6. 再次等待 NS 应答 1s 后超时,AR1 发出对 LCoA-1 的第三次 NS;

7. 此时对 LCoA-1 执行的 DAD 已经完成,所以 MN 发出对 LCoA-1 的 NA 到 AR1;

8. AR1 此时方能将 BA(0)传给 MN。

究其原因,MN 接入新的 AR 后产生异常 BU 延时的原因可归结如下:DAD 之前等待了随机一段时间,而 DAD 的延时又恰好和 AR 重试 NS 的间隔时间相同,因此导致 AR 进行多次 NS 才能得到相应的 NA 响应。因此总的延时为:

$$Delay = \begin{cases} RTT_{BU/BA} + 2s, & \text{若 } RTT_{BU/BA} < T_{PreDAD} \\ RTT_{BU/BA} + 1s, & \text{若 } RTT_{BU/BA} > T_{PreDAD} \end{cases}$$

其中 $RTT_{BU/BA}$ 表示从 MN 到 HA 或 MAP 的来回线路延迟, T_{PreDAD} 表示 MN 在开始 DAD 之前随机等待的时间。

在这种情况下,对于进入一个新的外地网络的 MN,其切换开销主要是由于进行 DAD 造成。在 HMIPv6 中,如果 MN 移动并接入同一个 MAP 管理域里的新子网,DAD 仍然必须执行,可以看到这种开销对于在同一 MAP 管理域内切换时的性能损害尤其明显。假设 MAP 到 HA 的来回线路延迟为 RTT_{HA} ,而 MN 在某 MAP 管理域内到 MAP 的来回线路延迟为 RTT_{MAP} ,HMIPv6 设计的目标是使 MN 在此 MAP 管理域内移动时进行绑定更新的延时与通常 MIPv6 进行绑定更新的延时的比例尽量接近 RTT_{MAP}/RTT_{HA} 。而由于 DAD 的存在,两者的比例为:

$$\frac{D_{MAP}}{D_{HA}} = \begin{cases} \frac{2+RTT_{MAP}}{2+RTT_{HA}}, & \text{若 } RTT_{HA} < T_{PreDAD} \\ \frac{2+RTT_{MAP}}{1+RTT_{HA}}, & \text{若 } RTT_{MAP} < T_{PreDAD} < RTT_{HA} \\ \frac{1+RTT_{MAP}}{1+RTT_{HA}}, & \text{若 } RTT_{MAP} > T_{PreDAD} \end{cases}$$

其中, D_{MAP} 和 D_{HA} 分别表示向 MAP 和 HA 进行绑定更新时,从发送 BU 到接收相应 BA 的延时。可以看出,最好的情况为 $RTT_{MAP} > T_{PreDAD}$ 时,向 MAP 的绑定更新延时减少为向 HA 进行绑定更新延时的 $(1+RTT_{MAP})/(1+RTT_{HA})$ 。在我们的实验中, $RTT_{HA} \approx 100ms$, $RTT_{MAP} \approx 1ms$,因此所得仅有 $1.001/1.100=0.91$ 的延时减少。事实上,由于 T_{PreDAD} 的期望为 $500ms$,因此现实中极少出现 $RTT_{MAP} > T_{PreDAD}$ 的情形,在绝大多数情况下,我们处于 $RTT_{HA} < T_{PreDAD}$ 的情形。

当 MN 接回 AR1 时,由于从 LCoA-1 的生命期还没有结束,因此 MN 无需进行 DAD 就能继续以 LCoA-1 与 AR、MAP 以及 Internet 通信,因此各种绑定更新的延时符合预期。

4 基于 MLD 的快速 DAD 协议(MLD-Fast DAD)

IPv6 节点进行无状态自动地址配置时生成地址的网络接口 ID 是由其网卡的链路硬件地址(如 MAC 地址)决定的,而由于生产厂商生产的网卡链路地址绝大多数是全球唯一的,因此地址前缀和网络接口 ID 同时相同的可能性是微乎其微的,据文献报道此种可能仅为 $1/10G^{[7]}$ 。由此可见,对于绝大多数情况会成功的 DAD,花费 1s 的时间进行超时等待代价过大。因此,要想提高 MIPv6 尤其是 HMIPv6 的性能,一个

关键就是尽可能减小 DAD 所带来的影响。

Greg Daley 首先研究了基于 MLD 的减少 MN 进行 DAD 的较长延时的方案(MLD-DAD)^[6],但是该方案存在一定的缺陷。下面我们将首先简要介绍 MLD 以及 MLD-DAD 协议,分析 MLD-DAD 方案的缺点,然后介绍我们设计的基于 MLD 的快速 DAD 方案(MLD-Fast DAD),并论证其性能上的巨大优势。

4.1 MLD 协议工作原理

MLD 协议^[3]要求 IPv6 节点在加入某个特定组播组之前先发送一个 MLD 消息给管理 MLD 的路由器(MLD-AR),MLD-AR 并不返回回答,但记录加入各个组播组的节点地址。此 MLD 消息格式(ICMPv6 类型)如图 2 所示。

0	7 8	15 16	31
Type	Code=0	Checksum	
Maximum Response Delay		Reserved	
Multicast Address(32bits)			

图 2 MLD 消息格式

4.2 MLD-DAD 工作原理

为了利用 MLD-AR 的 MLD 机制,要求 MN 开始 DAD 时,首先向本链路所有路由器组播组地址(ff::2)发送一条特殊 MLD 消息声明将加入相应节点请求组播组。与一般 MLD 消息不同的是,此消息要求 MLD-AR 给予响应(MLD Response Request)。这种新型 MLD 消息格式如图 3 所示。其中 Code 域置为 1 表示要求 MLD-AR 给与相应的应答,为 0 则表示不需要应答。

0	7 8	15 16	31
Type	Code = 1	Checksum	
Request ID		Reserved	
Multicast Address(32bits)			

图 3 MLD 请求应答消息格式

若 MLD-AR 收到 Code 域置 1 的 MLD 消息,而消息要求加入的组播组此前并未创立,MLD-AR 则发送给 MN 一个应答消息(MLD Response),格式如图 4 所示。其中 Code 域指明该 MLD 消息是否是第一个要求加入该组播组。若是,则 MN 接收到此应答后可以立即成功结束 DAD,否则进行传统的 DAD 过程。

0	7 8	15 16	31
Type	Code	Checksum	
Request ID		Reserved	
Multicast Address(32bits)			

图 4 MLD 应答消息格式

4.3 MLD-Fast DAD 工作原理

由于节点 IPv6 地址的网络接口部分低 24 位相同的可能性很小,因此同一链路上两个节点加入同一节点请求组播组的可能性也很小,上述 MLD-DAD 协议可以基本消除为完成 DAD 所需等待的 1s 时间,从而减少了进入新的外地网络后第一次绑定更新的延时。如果 $RTT_{BU/BA} < T_{PreDAD}$,则在 MN 开始进行 DAD 之前 BA 已经到达 AR。此时 AR 发出对 CoA 的 NS,显然是得不到相应的 NA 应答的。所以 AR 不得不等待 1s 超时后再次试图发送 BA 给 CoA;由于 $T_{PreDAD} < 1s$,如

果 DAD 按照 MLD-DAD 顺利提前完成,则 AR 得到 NA 应答,此时就能将 BA 发送给 CoA。这种情况下 MLD-DAD 将 MN 进入新的外地网络后的第 1 个绑定更新的延时减少为 $(RTT_{BU/BA} + 1)s$ 。类似分析可得,当 $RTT_{BU/BA} > T_{PreDAD}$ 时,MLD-DAD 将此延时降为 $RTT_{BU/BA}$ 。

但是,正如我们之前指出,大多数情形是 T_{PreDAD} 远大于 $RTT_{BU/BA}$,在表 1、2 中列出的实验数据中看到两次 T_{PreDAD} 分别为 168ms、609ms(经过多次实验均有类似结果)。而从 MN 发送 BU 到 BA 返回到相应 AR 仅需约 100ms;向 MAP 发送 BU 到 BA 返回 AR 更是只需要 1ms 左右的时间,几乎不可能大于 T_{PreDAD} 。在这种情形下,MLD-DAD 协议有两个缺陷:

1. 大多数情况下,MN 仍然需要等待多于 $RTT_{BU/BA}$ 的时间 T_{PreDAD} 才能开始进行 DAD;

2. DAD 完成后不能立即发送对之前收到 NS 的 NA 应答。

为此,我们设计了 MLD-Fast DAD 方案,目标就是消除 MLD-DAD 上述两个缺陷。MLD-Fast DAD 协议将按照如下描述工作:

• 当 $RTT_{BU/BA} < T_{PreDAD}$ 时,

1. MN 收到对其某个 CoA 的 NS 后,如果判断出此 NS 并非别的节点在进行 DAD,并且此 CoA 处于临时状态(Tentative),则触发 DAD 过程;

2. 整个 DAD 过程同 MLD-DAD;

3. MN 收到 MLD 应答成功完成 DAD 后,立即发送一个对该 CoA 的 NA,目的地址为本链路所有节点组播组地址 ff02::1;

4. AR 收到对此 CoA 的 NA 后,立即将目的地址为该 CoA 的数据包发送给 CoA。

• $RTT_{BU/BA} > T_{PreDAD}$ 时,过程同 MLD-DAD。

4.4 MLD-FastDAD 协议性能分析

同一条链路上的来回传输延时一般小于 1ms,从 AR 发出对 CoA 的 NS 到 MN 收到 BA 仅需 1ms 左右,记为 ϵ 。因此按照 MLD-DAD 的设计,可以将 MN 进入新的外地网络后第一次绑定更新的延时减少到大约为 $RTT_{BU/BA}$ 。对 MIPv6 而言,相应的绑定更新延时与传统方案以及 MLD-DAD 方案的比较如表 4 所示。

表 4 DAD、MLD-DAD、MLD-Fast DAD 性能对比

条件	发生概率	$T_{DAD}(s)$	$T_{MLD-DAD}(s)$	$T_{MLD-FastDAD}(s)$
$RTT_{BU/BA} < T_{PreDAD}$	很大	$RTT_{BU/BA} + 2$	$RTT_{BU/BA} + 1$	$RTT_{BU/BA}$
$RTT_{BU/BA} > T_{PreDAD}$	比较小	$RTT_{BU/BA} + 1$	$RTT_{BU/BA}$	$RTT_{BU/BA} + 1$
实验数据		2.1	1.1	0.1

采用 MLD-FastDAD, HMIPv6 中在 MAP 管理域内移动时向 MAP 进行本地化绑定更新延时对传统 MIPv6 的延时效比为:

$$D_{MAP}/D_{HA} = (RTT_{MAP} + \epsilon) / (RTT_{HA} + \epsilon)$$

同样对已有实验数据,可以预期能够得到如下改进:

$$D_{MAP}/D_{HA} = (1+1)/(100+1) \approx 1/50$$

表 5 比较了本文提及的三种方案在 HMIPv6 中得到的切换速度提升。从表中可以看出,当 MN 进入新的外地网络时,在 $RTT_{MAP} < T_{PreDAD} < RTT_{HA}$ 的情况下,DAD 和 MLD-DAD 进行 HMIPv6 本地绑定更新的延时甚至大于直接向 HA 进行

绑定更新的延时。在每种条件下,MLD-FastDAD 的切换速度相比其它两种方案都有相当可观的提高,尤其是在 $RTT_{HA} < T_{PreDAD}$ 这种最可能发生的情况下。

表 5 三种方案在 HMIPv6 中切换速度比较

条件	发生概率	DAD	MLD-DAD	MLD-FastDAD
$RTT_{HA} < T_{PreDAD}$	很大	$\frac{RTT_{MAP} + 2}{RTT_{HA} + 2}$	$\frac{RTT_{MAP} + 1}{RTT_{HA} + 1}$	$\frac{RTT_{MAP} + \epsilon}{RTT_{HA} + \epsilon}$
$RTT_{MAP} < T_{PreDAD} < RTT_{HA}$	比较小	$\frac{RTT_{MAP} + 2}{RTT_{HA} + 1}$	$\frac{RTT_{MAP} + 1}{RTT_{HA}}$	$\frac{RTT_{MAP} + \epsilon}{RTT_{HA} + \epsilon}$
$RTT_{MAP} > T_{PreDAD}$	几乎不可能	$\frac{RTT_{MAP} + \epsilon}{RTT_{HA} + \epsilon}$	$\frac{RTT_{MAP} + \epsilon}{RTT_{HA} + \epsilon}$	$\frac{RTT_{MAP} + \epsilon}{RTT_{HA} + \epsilon}$
实验数据		2001/2100	1001/1100	1/50

(注:分别采用 DAD、MLD-DAD 和 MLD-Fast DAD 时 HMIPv6 本地绑定更新与 MIPv6 的绑定更新性能比较(D_{MAP}/D_{HA}))

4.5 MLD-Fast DAD 协议的健壮性

MLD-FastDAD 同 MLD-DAD 一样,仅仅改动了当前 MLD 协议中未使用的域。对于不理解其协议的 MLD-AR 或主机而言,仅需忽略消息报文中的不识别信息。

MLD-FastDAD 相比 MLD-DAD 的主要改进在于:

1. 将 T_{PreDAD} 缩短为 $RTT_{BU/BA}$,相对于接入的外地网络中其它主机而言,仍可看作是随机的;

2. DAD 成功后立即发送 NA。

DAD 的过程 MLD-Fast DAD 与 MLD-DAD 基本相同,因此协议的健壮性与 MLD-DAD 基本相同^[6]。

MLD-Fast DAD 并没有将 T_{PreDAD} 减少为 0,而是将其减少为 $RTT_{BU/BA}$ 。 T_{PreDAD} 减少为 0 可以消除表中的 ϵ 项。但是我们不愿意破坏 RFC2462 设立 T_{PreDAD} 的宗旨——避免同一链路上多台主机同时启动并立即进行 DAD 所造成的网络拥塞。MLD-Fast DAD 仅由 MN 执行,而某外地子网内多台来自同一个家乡网络($RTT_{BA/BU}$ 相同)的 MN 同时启动并引起网络拥塞的概率我们认为可以忽略不计。

总结 传统 IPv6 网络中,MN 移动到新的外地网络后进行 CoA 配置时的 DAD 过程严重损害了 MIPv6 尤其是 HMIPv6 的切换速度。本文结合对从 HMIPv6 实验床所搜集的数据以及 MLD-DAD 协议的分析,提出了性能更好的 MLD-Fast DAD 协议,通过对网络事件的时序分析,论证了此方案能极大地提高 MIPv6 尤其是 HMIPv6 的切换速度,更能体现 HMIPv6 的优势。

参考文献

- Neighbor Discovery in IP Version 6 (IPv6). RFC2461, Dec. 1998
- IPv6 Stateless Address Autoconfiguration. RFC2462, Dec. 1998
- Multicast Listener Discovery (MLD) for IPv6, RFC2710, Oct. 1999
- Mobility Support in IPv6, Internet Draft, Feb. 2003
- Hierarchical Mobile IPv6 Mobility Management, Internet Draft, Oct. 2002
- Duplicate Address Detection Optimization using IPv6 Multicast Listener Discovery, Internet Draft, Feb. 2003
- Bagnulo M, Soto I, Garcia-Martinez A, Azcorra A. Random Generation of Interface Identifiers, Revision 00, Internet Draft, Jan. 2002
- NIST Net. <http://snad.ncsl.nist.gov/itg/nistnet/>