第32卷 第21期 计 算 机 工 程 2006年11月
Vol.32 No.21 Computer Engineering November 2006
·网络与通信· 文章编号：1000—3428(2006)21—0113—03 文献标识码：A 中图分类号：TP393

基于MIPv6的AAA系统接入控制的实现

廖 斌，郭 巧，蔺 源，沈 刚
(北京理工大学网络信息中心，北京 100081)

摘 要：随着移动 IPv6 技术的发展以及无线移动终端的普及，无线网络在人们的生活中扮演着越来越重要的角色，但应用于固定网络的AAA 系统已经无法适用于无线环境。该文针对移动 IPv6 环境下，由于用户在不同网络间漫游所导致的接入控制交接问题，提出了基于MIPv6 的 AAA 系统接入控制方案，给出了 Linux 平台上的实现，通过实验证明了方案的有效性和可行性。

关键词：移动 IPv6；AAA；接入控制；DIAMETER

Implementation of AAA Access Control System Based on MIPv6

LIAO Bin, GUO Qiao, LIN Yuan, SHEN Gang
(Network Information Center, Beijing Institute of Technology, Beijing 100081)

【Abstract】With the development of mobile IPv6 technology and the popular usage of mobile wireless terminal, wireless network plays a more and more important role in people life. However, the AAA systems used in the wired-networks are no longer appropriate to the wireless network. This paper focuses on the access control handover when user roams between networks, and proposes a resolution of access control in AAA system based on MIPv6. The paper implements it on the Linux platform and tests it on the test bed, which confirms its validation and feasibility.

【Key words】 Mobile IPv6; AAA; Access control; DIAMETER

随着网络应用的飞速发展，人们对Internet的依赖无处不在。然而当今出现的基于低速的手机通信网络Internet接入机制已经无法满足用户需要，于是基于WLAN的高速无线互联网进入人们的视野，它目前最新的通信协议 802.11G速度达到 54Mbps，能够很好地满足用户的需要。移动IPv6(MIPv6)[1]的出现，更为用户提供了一个速度更快，性能更高的IP网络平台。但是这种移动的、无处不在的互联网给资源的管理带来了许多问题，AAA正是为解决此类问题而提出的。

IETF 的 AAA 协议主要有 RADIUS 和 DIAMETER，后者借鉴了前者的优点，采用了新的体系结构，使得 AAA 协议更符合移动无线互联网的要求。但是当用户从一个网络漫游到另一个网络时，网络如何完成两个网络之间的接入控制交接，运营商如何实现对漫游用户进行实时的控制和收费，DIAMETER 并没有明确怎么解决这些问题，这正是本文提出的基于 MIPv6 的 AAA 系统接入控制的解决方案所针对解决的问题。

本方案应用了最新的 AAA 协议 DIAMETER，采用了主机无状态地址配置方式，构建于 MIPv6 无线移动网络基础之上，使得方案实现表现出良好的性能。本方案能很好地解决用户在网络间漫游的接入控制问题，实现了透明的接入控制交接，从而实现了实时地授权和计费，既保证了运营商的利益，又使得用户方便快捷透明地在不同的网络间漫游，使用不同网络的资源。

1 基于 DIAMETER 的 AAA 系统框架

根据文献[2]，在基于 Diameter 的 AAA 系统中，主要包含 Diameter 服务器、Diameter 客户端、Diameter 中继、Diameter 代理、Diameter 重定向器等节点。Diameter 服务器完成一个特定的管理域内的认证、授权和计费处理。Diameter 客户端为最终用户提供网络的访问，如 MIP 中的外地代理。Diameter 中继和 Diameter 代理都可根据系统需要部署，能够很好地扩展系统范围，Diameter 重定向器提供了统一处理 Diameter 消息的路由功能，减少了其它各个节点的路由配置负担。文献[3]给出了基于 MIPv6 的 DIAMETER AAA 系统的基本模型，如图 1 所示。

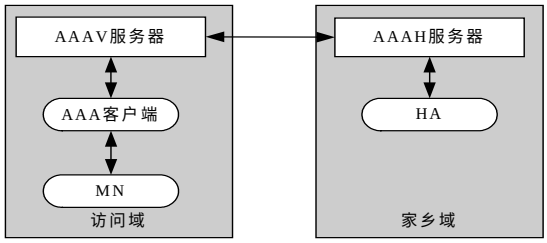

图 1 MIPv6 AAA 系统模型

2 移动 IPv6

移动 IPv6(MIPv6)定义了 3 个实体移动节点（MN）、家乡代理（HA）和通信对端（CN）。当 MN 漫游到外地网络时，分别发送绑定更新（BU）到 HA 和 CN 以保持通信的连续性。BU 是一个包含移动报头（MH）的 IPv6 数据包，它可以携带移动选项。移动选项包含类型、长度和取值 3 个数据段，即为 TLV 形式。MIPv6 规定移动选项的类型和长度字段均为 8b，所以用户可定义的选项有 256 种，选项取值最多为 256B。至今为止，IETF 定义了 5 种移动选项。

基金项目：IPv6-CJ 基金资助项目
作者简介：廖 斌(1980－)，男，硕士生，主研方向：下一代互联网；郭 巧，博导；蔺 源、沈 刚，硕士生
收稿日期：2006-02-03 **E-mail：**qguo@bit.edu.cn

3 无状态地址自动配置

无状态地址自动配置[4]为IPv6 网络主机自动配置的一种方式，它使得网络主机通过路由器公告就能方便而快捷地完成地址以及其它信息的配置。移动IPv6 采用此种主机配置方式能够减少切换延迟，更好地实现无缝切换，提高网络性能。

4 技术方案

我们提出的解决方案包括两部分：MN 端部分和接入网关（AG）端部分。MN 端包括 2 个功能模块 :接入客户端（AC）以及验证信息管理模块（AIM）。AG 端包括 5 个功能模块：验证信息收集模块(AIC)，验证请求模块(AR)，MN 信息管理模块(MNIM)，数据包过滤模块（DF）以及接入服务器（AS）。大体框架如图 2 所示。

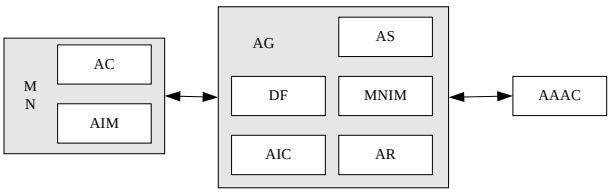

图 2 方案基本框架

此方案的主要思想是网络采用无状态地址自动配置方式对接入的 MN 配置地址信息，用户使用便携设备 MN 通过两种方式接入网络：静态接入和漫游接入。它们是 MN 不同时机的接入方式，静态接入是指 MN 首次接入网络时的通过使用客户端连接接入服务器接入网络的方式，而漫游接入是指 MN 通过静态接入方式接入网络后，从一个子网漫游到另外一个子网后的接入方式，此过程完全无需用户干预，它是实现接入控制透明交接的关键。

4.1 静态接入

MN 必须首先通过静态接入方式接入到网络。MN 配置完地址信息后，通过 AC 连接 AS，向 AS 发送接入请求；然后由 AS 向 AAAC（AAA Client）发出 AAA 验证请求；验证应答信息沿着相反的路径返回给 MN。同时，AS 根据收到的应答信息的状态向 MNIM 模块发送处理请求 如果验证成功，则请求 MNIM 模块添加此 MN 相关的信息，以使得此 MN 发出的数据报能通过 DF 模块，然后把验证应答转发给 MN；否则，仅仅把验证应答转发 MN。MN 收到验证应答之后，根据验证成功与否来决定是否把验证信息传递给 AIM 模块，为将来发生的漫游接入做准备。

4.2 漫游接入

当 MN 成功地接入网络后从一个子网漫游切换到另一个子网时，它请求 AIM 模块获取用户的验证信息，然后通过 BU 中的移动选项携带用户验证信息 NAI 发送出接入验证请求，而接入网关(AG)的验证信息收集模块截获此 BU 并从中收集信息，然后发送给 AR 模块，由 AR 模块向 AAAC 发起请求，此请求同时完成绑定更新和接入验证过程。AR 收到应答后，如果验证成功，则通知 MNIM 模块更新 MN 的信息表，以及从应答中提取绑定更新应答返回给 MN；否则通知 MNIM 模块删除 MN 的信息，然后返回包含出错码绑定更新应答给 MN。MN 根据收到的绑定更新应答成功与否判定是否成功地接入所漫游到的网络。

4.3 数据包过滤

数据包过滤由 AG 上的 DF 模块实现。对于 AG 收到的每个数据包，DF 向 MNIM 发出获取此数据包对应 MN 状态请求，如果返回已经通过验证，则放行此数据包，否则丢弃此数据包。而每个接入的 MN 的信息由 MNIM 模块管理，并且为其它模块提供 MN 信息操作接口。

5 具体实现

5.1 实现平台

目前，对 MIPv6 的实现中性能最好的就是由芬兰赫尔辛基大学的开发的开源工程 MIPL（Mobile IPv6 for Linux），因此我们采用 Red Hat Linux 9.0（内核为 2.4.22）作为操作系统平台，MIPL 1.0 版本作为 MIPv6 实现。

在此实现中，MN中的AIM模块以及AG中的除了AS所有模块工作在内核空间，而AC和AS模块则工作在用户空间。内核和用户空间之间的通信采用netlink套接字[5]来实现。netlink 套接字的最大特点是对中断过程的支持，它在内核空间接收用户空间数据时不再需要用户自行启动一个内核线程，而是通过另一个软件中断调用用户事先指定的接收函数，而且它还支持全双工通信方式。当 netlink 套接字用于内核空间与用户空间的通信时，在用户空间的创建方法和一般套接字使用类似，而在内核空间则是调用相关的接口创建即可。

5.2 接入客户端与服务器

AC 和 AS 设计为 C/S 模式，AC 和 AS 之间通过 socket 交互验证请求和应答，实现比较简单，对此不作详细介绍。

5.3 MN 端实现

在 MN 端，我们对 MIPL 绑定更新例程作些更改，定义新的移动选项和新的绑定更新标记，用于发送绑定更新时携带用户验证信息 NAI。在用户通过静态接入方式成功地接入网络后，NAI 由接入客户端通过 netlink 套接字传递到 AIM 模块中缓存起来，MN 发送绑定更新时向 AIM 发送请求获取 NAI，然后填写到 BU 移动选项和 BU 一起发送出去。

(1)新的绑定更新标记

绑定更新标记用于标记此更新的类型或者处理方式，定义一新的标记来标记绑定更新携带了 AAA 验证信息，这样接入网关收到携带此标记的数据报后交由 AIC 模块处理。此标记定义为：MIPV6_BU_F_AAA，取值为 0x08 的常量，称其为 AAA 标记。

(2)新的移动选项

此类选项用于携带 AAA 验证信息，类型定义标记为 MIPV6_OPT_AAA_DATA，取值为 0x06 的常量。类型定义为

```
struct mipv6_mo_aaa_data
{
    __u8        type;
    __u8        length;
    __u8        *data;
};
```

它的定义遵守选项定义为 TLV 形式的规约，各字段具体用途如下：

type：类型，取值为 MIPV6_OPT_AAA_DATA；

length：长度；

data：无符号 8 位指针，指向 NAI 验证信息。

(3)主要处理流程

1)启动家乡注册（init_home_registration），它在完成相关的其它准备工作之后发送 BU 之前，置 AAA 标记位，然后读取 NAI 信息并填充到 BU 选项，最后发送 BU。

2)MN 到 HA 切换（mn_ha_handoff），它在发送 BU 之前，判断 AAA 标记是否置位，如果置位则读取 NAI 信息并填充到 BU 选项，然后发送 BU；否则直接发送 BU。

5.4 AG 端实现

在AG端，最主要的是DF和AIC两个模块的实现，对于其它模块，由于实现比较简单，不作详细介绍。DF和AIC实现主要基于Linux网络的Netfilter[6]框架，它们根据各自功能需要，分别注册了不同的等级钩子（HOOK）函数。

(1)DF 模块

DF 模块用于过滤数据包，控制接入网络的用户对网络资源的使用。它在 NF_IP6_FORWARD 钩子注册了级别为 NF_IP6_PRI_FILTER+2 的钩子函数。钩子函数操作流程如下：

1)对于需要转发的每个数据包，首先用它的源地址为参数请求 MNIM 模块获取此地址的用户验证状态，如果已经通过验证，则返回 NF_ACCEPT；

2)用它的目的地址为参数请求 MNIM 模块获取此地址的用户验证状态，如果已通过验证，则返回 NF_ACCEPT，否则返回 NF_DROP，通知系统丢弃此数据包。

(2)AIC 模块

AIC 模块负责从 BU 选项中提取用户的验证信息，然后发送请求给 AR 模块，由 AR 完成验证发起以及应答接收的过程。AIC 在 NF_IP6_FORWARD 钩子注册了级别为 NF_IP6_PRI_FILTER+1 的钩子函数。目的是防止 DF 模块丢弃携带验证信息的正在发起验证的用户所发送的 BU。钩子函数操作流程如下：

1)首先尝试从数据包中提取家乡地址，如果提取失败则可以肯定此数据包不是 BU，返回 NF_ACCEPT；

2)分析扩展头，如果不存在移动 IPv6 扩展头，则返回 NF_ACCEPT；

3)解析出移动 IPv6 扩展头，判断 AAA 标记是否置位，如果没有置位，则返回 NF_ACCEPT；

4)以此 BU 的转交地址为参数向 NMIM 模块请求此地址对应的用户地验证状态，如果已通过验证，则从 BU 中去掉 AAA 验证信息选项，然后返回 NF_ACCEPT，如果正在验证，则返回 NF_DROP；

5)从 BU 选项中提取出用户验证信息 NAI，以及绑定更新数据，向 AR 模块发送请求。

6 实验结果

根据实现方案，搭建如下实验平台，如图 3 所示。实验平台包括子网 A 和子网 B，使用笔记本 MN 通过 AP-A 与 AP-B 连接网络，HA-A 以及 HA-B 分别为两个子网的家乡代理，而 AG-A 和 AG-B 为我们的接入网关。MN 的家乡网络为 Subnet-A。

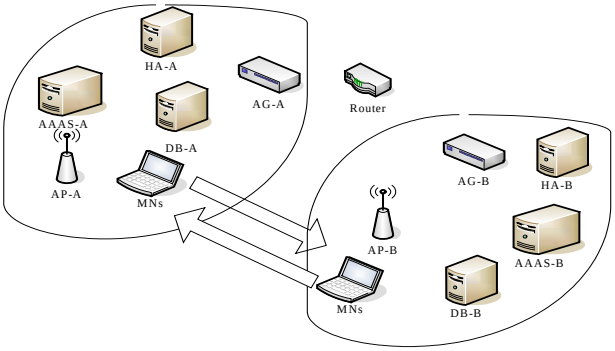

图 3 实验平台

首先以静态接入的方式以 banliao 为用户名接入家乡网络，然后从家乡网络 Subnet-A 漫游到 Subnet-B，通过系统的日记消息，如图 4~图 7，可以清晰地看到接入控制交接所引发的整个验证过程。MN 在发送 BU 的时候携带的 AAA 验证信息被接入网关解释出来，然后发起验证过程，而验证的同时完成了绑定更新，从而缓解了验证时延所带来的切换延迟，同时 AG-A 上将释放 MN 所分配的资源及停止对 MN 的计费。而通过 ping 包所产生的结果如图 8 以及图 9，可以看到整个交接过程所产生的时延较短，系统并没有明显地降低 MIPv6 的性能，相信通过采用更好的切换算法将会使得系统的性能更优异。

```
Aug 29 14:57:47 ipv6 kernel: mip6[mipv6_send_bu]: Sending BU to CN 2001:251:1a05:1:0:0:0:1 for home address 2001:251:1a05:1:0:0:0:3
Aug 29 14:57:47 ipv6 kernel: mip6[mipv6_send_bu]: AAA authentication information for: banliao
Aug 29 14:57:47 ipv6 kernel: mip6[mipv6_send_bu]: Setting bul callback to bul_resend_exp
```

图 4 MN 发送携带 AAA 验证信息的 BU

```
(4552|147466) Client: Starting client for user banliao
(4552|147466) Client: Session Started, session id=.;1125298845; 1; 4312b29d
(4552|147466) Client: Sending auth message # 1
lifetime in bu is: 2415984640
sequence in bu is: 0
flag in bu is: 200
coa is: 20001:251:1a05:2:208:74ff:fe23:3897
```

图 5 AAA 客户端发起验证

```
we have received an auth answer message!!
        Result-Code =2001
errormsg is: login success
we have get ba from answer!
BA = [
                status = 0
                Reserved = 0
                Sequence = 0
                Lifetime = 400
]
lifetime in ba from auth answer is: 400
ready to send auth ack to kernel!
```

图 6 AAA 客户端收到验证应答

```
Aug 29 14:57:47 ipv6 kernel: mip6[mipv6_bul_add]: adding entry: c6e530a0
Aug 29 14:57:47 ipv6 kernel: mip6[mipv6_ba_rcvd]: BA received with sequence number 0x0, status: 0
Aug 29 14:57:47 ipv6 kernel: mip6[mipv6_ba_rcvd]: setting callback for expiration of a Home Registration: lifetime:400, refresh:320
```

图 7 MN 收到 BA

```
64 bytes from 2001:251:1a05::111: icmp_seq=10 ttl=63 time=0.426 ms
64 bytes from 2001:251:1a05::111: icmp_seq=11 ttl=63 time=0.436 ms
64 bytes from 2001:251:1a05::111: icmp_seq=12 ttl=63 time=0.439 ms
64 bytes from 2001:251:1a05::111: icmp_seq=16 ttl=63 time=0.622 ms
64 bytes from 2001:251:1a05::111: icmp_seq=17 ttl=63 time=0.994 ms
64 bytes from 2001:251:1a05::111: icmp_seq=18 ttl=62 time=0.482 ms
```

图 8 ping6 在 MIPv6 MN 切换时的结果

(下转第 118 页)