

IMS 应用层连续性研究

吕新荣 廖建新 朱晓民 武家春

^①(北京邮电大学网络与交换技术国家重点实验室 北京 100876)

^②(东信北邮信息技术有限公司 北京 100083)

摘要: 应用层移动性对在移动环境下的 IP 应用很重要, 底层的移动性支持保证了切换中数据传输的平滑性, 而应用层移动性要保证上层应用在移动中正确收发数据。文中提出了移动扩展 SOCKET, 即 MESOCK(Mobile Extension SOCKET)来实现应用层的移动性。它允许应用数据传送中变换数据传输的源和目的 IP 地址, 使得数据可以直接经路由到达目的地而不是采用隧道的方式。能够很好地配合底层的 IP 地址切换, 即终端地址一旦变化, 应用层即启用新的 IP 地址来收发数据, 而不是通过数据隧道发送数据。并在地址切换过程中很好地保持连接的持续性, 从而保证服务的连续性和数据传输的完整性。仿真结果证明 MESOCK 可以很好地实现这一目标。

关键词: 移动通信; 应用移动性; 服务连续性; IMS; TCP 连接

中图分类号: TN929.5

文献标识码: A

文章编号: 1009-5896(2007)09-2217-05

Research on Application Continuity in IMS

Lü Xin-rong Liao Jian-xin Zhu Xiao-min Wu Jia-chun

^①(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

^②(EB Information Technology Co. Ltd., Beijing 100083, China)

Abstract: Application layer mobility is important for IP application under mobile environment. The support of under layer mobility guarantees the smooth of data transaction in handoff, while application layer mobility guarantees higher layer application can transfer and receive data correctly in handoff. MESOCK (Mobile Extension SOCKET) is introduced for application layer mobility, which is an extension of TCP/UDP and let running session using new IP address to transfer and receive data after getting it. So it can let data go to the destination not be forwarded by tunnel. It makes the data's integrity and continuity in the IP changing process. The result of simulation proves that the MESOCK can get the goal well.

Key words: Mobile communication; Application mobility; Application continuity; IMS; TCP connection

1 引言

UMTS引入IMS(IP Multimedia Subsystem)^[1]是移动网络向NGN演进的重大举措, IMS是一个与接入技术无关的基于IP体系结构的标准体系, 它可以与现存的语音和数据网络互通。IMS体系结构下各种类型的客户端之间都可以建立起具有特定QoS的P2P IP通信。IMS使互连网络与通信网, 固定网络与移动网络融合为一体, 成为一个IP通信网络, 使得通信网络可以提供更加丰富的多媒体业务, 如多媒体电话、会议、网络游戏、移动视频等。随着终端的智能化程度加强, 移动环境下的业务也会越来越丰富, 基于IP的应用目前大都在TCP/UDP上运行, 在移动中如何保证TCP连接的移动性成为移动环境下IMS/NGN应用的一个课题。移动中的服务质量和连接的连续性有很大关系, 终端的移动性要求服务在

移动环境中也要保持连续性, 即切换中平滑性, 特别是实时服务。

IMS中的控制协议使用SIP^[2], SIP可以支持IMS的移动性管理, 在文献[1]中对此有详尽的说明, 当移动终端的IP地址发生变换时, 终端中正在运行的会话进行切换, 以使得用户数据在网络中没有迂回, 从而减少IMS网络的流量负担。由于SIP切换涉及的节点不仅是本地网络的网络节点, 而是跨越了整个网络到达通信对端的节点, 切换时间很长^[3,4], 文献[4]认为SIP不适合用来支持对延迟要求严格的实时流媒体, 其原因是在SIP切换期间由于应用层对传输数据的同步和数据连接的维护, 用户数据在切换期间中断传输。虽然IMS的接入网可以保证切换的平缓性, 如UMTS中通过GPRS接入, GPRS系统可以采用软切换技术或移动IP技术保证切换的平滑性, 并使得切换中以新旧地址发往MS的数据包都能正确地到达MS, 但是应用层由于不能识别数据源和目的地址而不能正确接收应用数据, 因此保证应用在切换中的连续性, 必须在底层切换平滑保证的基础上, 保证TCP/UDP层在切换中的平滑性。对应用层来讲只要保持传

2006-02-10 收到, 2006-09-04 改回

国家杰出青年科学基金(60525110), 新世纪优秀人才支持计划(NCET-04-0111), 高等学校博士学科点专项科研基金(20030013006), 电子信息产业发展基金(基于3G的移动业务应用系统)和电子信息产业发展基金重点项目(下一代网络核心业务平台)资助课题

输层连接的持续性就可以保证应用的持续性。

2 相关工作

移动终端可能经常在不同的接入网之间移动, 保证业务在移动中服务质量是 UMTS 系统, 甚至是 NGN 的重要要求之一, 而应用层的移动性则是这一保证的一个重要方面。人们寻找了一些方法来支持应用层的移动性。

文献[5]中提出了一种RMS的方案来支持实时流媒体, 在通信双方的应用层和传输层间添加插件, 终端必须预测IP地址的改变, 一旦发现IP地址改变则将UDP接收和发送套接字切换到一个新的套接字上, 移动期间在多个备用套接字之间进行切换, 优点是不用改变现有的协议, 但是只是对UDP应用有效, 对TCP应用则无能为力。另外文献[6]提出了MSOCK(Mobile Socket)方案, 针对TCP进行改进, 在IP切换期间, 添加新的TCP连接交换流程, 同步更新通信两端的连接环境, 并告知对端地址的变换, 虽然可以实现应用层的移动性, 但是过程复杂, 也不适合SIP环境。SCTP(Stream Control Transaction Protocol)^[7]是一种全新的协议, 支持多宿主多连接, 可保证高效可靠的流媒体传输, 但是它本身并不支持移动性。多宿主、多连接在连接建立之后就是确定的, 只不过是增加了冗余备份, 传输数据时仍然是一个连接, 若活动连接无效, 则可以启用其它的连接。

本文所提出的 MESOCK 移动扩展多 IP 套接字, 可解决 TCP 连接的快速切换, 同时可解决文献[5]中解决的问题, UDP 在切换中的延时和丢包问题, 且不影响系统的控制信令流程。同样由应用程序来保证通信双方身份的可识别性。此方案可减少切换过程中的媒体传输中断时间, 并且可以保证 TCP 流的完整性, 并保证因切换而引起的数据重传很少。使其可以配合会话层的移动性以实现 SIP 的无缝切换。第3节中对移动扩展套接字进行了定义, 对其工作过程进行了描述, 在第4节对其进行了仿真实验, 最后给出了总结。

3 IMS 的会话应用切换过程

IMS 的会话管理是用 SIP 协议来实现的, 应用会话的切换包括两部分: SIP 会话的切换和媒体连接的切换。SIP 会话的切换过程如图 1 所示。用 SIP 协议中的 Re_invite 过程来实现, 在交换过程中携带新的通信参数。通信双方根据获得的通信参数, 重新维护通信环境。如果通信参数-通信终端的 IP 地址发生改变, 会话的方向必须改变, 即数据包的目的地址发生改变。对于 TCP 应用来讲是重新建立 TCP 连接, 对 UDP 应用来讲是改变通信双方的发送数据的目的地址。



图1 通信双方的参数协商过程

TCP/UDP 的套接字结构套接口由地址端口对组成, UDP 是无连接服务, 由端口来标识服务。只建立本地

SOCKET, 用来发送数据。接收数据时 UDP 协议层根据端口号来判断交给哪个服务, 由应用程序本身来处理数据, 区分数据源。TCP 是有连接的服务, 在通信之前要建立 TCP 连接, 每个连接由套接字对来标识: <本地地址, 本地端口; 远端地址, 远端端口>, 发送数据时使用此套接字对发送数据, 接收数据时, TCP 层根据 IP 交上的本地地址, 本地端口, 远端地址, 远端端口 4 项来匹配已有的 TCP 连接, 匹配成功则把数据交给响应的连接, 不成功则丢弃数据。UDP 应用在 SIP 切换中的连接, 处理过程如图 2 所示, 具体描述如下:

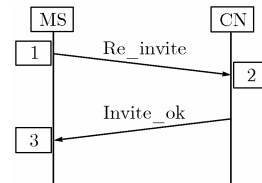


图2 UDP 应用的应用层移动性管理过程

(1)移动终端检测到地址变化, 向通信对端发送 Re_invite 来协商通信参数。

(2)通信对端收到 Re_invite 信令, 向应用层报告通信对端的地址变化, 应用层控制记录新的通信对端地址, 如果考虑到数据的重合, 要考虑与旧地址的结合使用。并向移动终端发送 OK 信令, 通信对端开始使用移动终端的新地址作为连接判定的主地址, 并开始使用新的移动终端的地址来向移动终端发送数据包。

(3)移动终端收到 OK 信令使用关闭旧本地套接字, 使用新套接字来发送数据。

(4)至此通信双方的连接同步已完成。

从过程中看通信终端的地址的改变对 UDP 应用的影响不大, 主要是应用层的数据源的跟踪问题, 但是对 TCP 应用就有很大的影响。在原始套接口下对于 TCP 应用, 为了保证数据的完整性并尽可能少地浪费已发送的数据, 必须使得通信双方对以传送数据的长度达成一致意见, 这就要在 Invite 过程中加入这些信息。由应用程序来控制连接的释放和建立(应用程序必须知道为其服务的套接字)。图 3 是 TCP 应用在 SIP 切换中的处理过程, 详细描述过程如下。

(1)MS 的应用层得到 IP 地址改变, 向通信对端发送 Invite 信令, 其中包含已经正确接受到的数据长度, 并开始关闭连接。

(2)对端向 MS 发送 OK 相应, 并将已正确收到的数据长度包含进信令。

(3)MS 收到 Invite_OK, 重新建立连接。

(4)连接已建立, 从断接点开始发送数据。

(5)连接已建立, 从断接点开始发送数据。

在原始套接口下 SIP 切换的整个过程到连接重新建立成功这段时间内, 对应用来讲数据传输是中断的。而且在关闭连接的过程中, 通信对端在收到 FIN 信令之前仍旧在发送数据, 在这段时间内的发送的数据是无效的。考虑到连接关闭

和 INVITE 过程的并行性，整个时间至少需要 5 倍的端到端时延。根据文献[4]中的分析 SIP 信令的端到端时延约为 6s，那么整个过程要中断 30s，而用户的底层连接是存在的，这种损失用户是不会满意的，而 MESOCK 则完全避免了这种情况。

4 移动扩展套接字(MESOCK)

底层链路为了减少数据的迂回路由，在终端接入新的网络时尽快使用新分配的 IP 地址来通信，通信对端使用旧地址发送的数据包，通过其它的方式正确到达终端(如 IP 的隧道技术，移动网络中在数据到达无线接入网络后，整个 IP 包是作为用户数据传输的，因此能正确到达无线接入网，就能正确到达终端)，并将数据交给 TCP/UDP 层。从第 3 节的介绍可知原来的 TCP/UDP 不能很好地支持移动环境中的应用。本节给出一种对 TCP/UDP 的移动性扩展 MESOCK(Mobility Extensive SOCKET)来支持移动环境中的应用。

MESOCK 扩展了 TCP/UDP 的套接字结构，使得其中的地址元素可以包含两个地址，一个地址是切换前地址，一个地址是切换中的新地址，这两个地址在切换过程中同时有效，使得通信双方在地址切换过程中使用新旧地址的数据都能正确地到达应用层。增加了套接字操作接口，提供给应用程序，在地址切换时对正在使用中的 MESOCK 进行修改，修改只影响套接口的地址元素，其它通信环境参数保持原状。原始 SOCKET 下通信地址的改变要重建通信环境，MESOCK 避免重新创建套接字，建立连接的开销。对 SIP 协议完全没有影响，并且可避免 TCP 连接关闭重建过程中的数据的无效传输。

MESOCK 的多地址套接字结构如图 4 所示，套接口仍旧是由<本地地址，本地端口；远端地址，远端端口>组成，但是地址变为两个元素，一个是当前地址，一个是切换地址。当前地址是发送数据包时使用的地址，两个地址共享同一个端口号。接受数据时，当 IP 层将 TCP 数据交上来时，将两个地址作为一个地址绑定使用，即检查时，连接的判定为套接字<本地当前地址||本地切换地址，本地端口，远端当前地址||远端切换地址，远端端口>=数据包<本地地址，本地端口，远端地址，远端端口>具体描述如下。

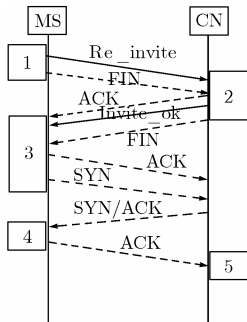


图3 TCP应用的应用层移动性管理过程



图4 MESOCK的原始套接口

数据发送过程:

- (1)套接字收到应用程序传来的发送数据;
- (2)TCP根据建立的套结字的收发端口组装TCP包;
- (3)从套结字中取出本地当前地址和远端当前地址;
- (4)将上面组装好的TCP包和取出的地址交给IP层处理;

数据接收过程:

- (1)IP层将<TCP包,接受地址,发送地址>交给TCP层;
- (2)TCP层解封TCP数据，得到应用层数据，发送端口和接受端口;
- (3)将<接受地址，接受端口，发送地址，发送端口>，

同存在的套结字进行匹配;

连接匹配过程:

```
for(所有系统中的已连接套结字)
    if(发送端口和接受端口同套结字一致)
        if((接受地址=本地当前地址||接受地址=本地切换地址)&&(发送地址=远端当前地址||接受地址=远端切换地址))
            {将数据交给此套结字 ;
             break;
            };
};
```

为套接字增加连接更新接口: (1)Conn_update(socketid, local_address, remote_address): 此操作保存原连接的一切环境，包括收发滑动窗、连接 ID 等，通信双方的 IP 地址改为所获得的新的地址，将替换调的地址仍然作为连接的一个元素存储。即将 local_address 写入本地当前地址，原地址写入本地切换地址; 将 remote_address 写入远端当前地址，原地址写入远端切换地址。(2)Conn_add_address(socketid, local_address, remote_address): 此操作保存原连接的一切环境，包括收发滑动窗、连接 ID 等，将 local_address 写入本地切换地址， remote_address 写入远端切换地址。参数说明: 其中 socketid 已连接套接字标识, local_address 是获得的新的本地 IP 地址, remote_address 是新获得的远端 IP 地址, 若没有新地址参数可以为空。

以 SIP 控制信令为例说明 MESOCKET 的使用过程，假设移动终端和通信对端使用<IP1, 端口 1>和<IP2, 端口 2>来建立连接，在移动终端处的套接口如图 4 所示。在通信过程中终端发生位置移动，网络切换完成后检测到终端地址发生变化，例如地址变为 IP1'，激发应用层移动性管理过程，如图 5 所示。因为 Re_invite 过程会引起会话状态的变化，而引起会话的中断，可使用 UPDATE 过程来代替，保持会话的激活状态。具体变化过程描述如下:

- (1) 移动终端检测到地址变化，向通信对端发送 Re_invite 来协商通信参数，并使用 Conn_add_address 命令原语来改变连接的参数，因为对方还不知道移动终端的地

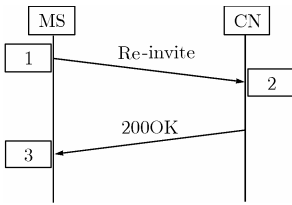


图5 MESOCK下应用层移动性管理过程

| 扩展 socket | | | |
|------------|------|------------|------|
| 本地当前地址:IP1 | 端口 1 | 远端当前地址:IP2 | 端口 2 |
| 本地切换地址:IP1 | | 远端切换地址:空 | |

图6 MS执行 Conn_add_address 后的 MESOCK 套接口

址变化, 如果使用 Conn_update 命令原语不能保证对端的正确接收, 执行过的套接口如图 6 所示, 此后移动终端仍然使用旧的 IP 地址来发送数据, 但是, 开始使用两个地址来进行连接判定, 但是旧地址仍然是主地址。移动终端的连接可以接收通信对端使用旧地址转发往本地的数据, 并将数据发往通信对端。此后的一段时间中存在三角路由。

(2)通信对端收到 Re_invite 信令, 使用 Conn_update() 改变连接的 SOCK 参数, 并向移动终端发送 OK 信令, 套接字的变化如图 7 中通信对端中所示, 此后, 通信对端开始使用移动终端的新地址作为连接判定的主地址, 并开始使用新的移动终端的地址来向移动终端发送数据包。通信直接将数据发往移动终端, 数据转发的三角路由消失。

(3)移动终端收到 OK 信令使用 Conn_update() 改变连接的套接字参数使用, 套接字的变化如图 7 中移动终端中所示, 移动终端开始使用自己的新地址来发送数据, 并且用新的地址作为连接判定的主地址。

(4)至此在扩展的 SOKET 下通信双方的连接同步已完成。

扩展套接口比原始套接口多出两个地址空间的代价, 套接口选择判定中地址次数在切换完成后没有影响, 因为当前地址总是正在使用的地址。只有在切换过程中, 由于通信双方地址切换的时间不同步或因数据传输路径不同导致到达数据失序的情况下出现比较两次地址的情况。

| 通信对端 | | | | 移动终端 | | | |
|------------|------|------------|------|------------|------|------------|------|
| 本地当前地址:IP2 | 端口 2 | 远端当前地址:IP1 | 端口 1 | 本地当前地址:IP1 | 端口 1 | 远端当前地址:IP2 | 端口 2 |
| 本地切换地址:空 | | 远端切换地址:IP1 | | 本地切换地址:IP1 | | 远端切换地址:空 | |

图7 执行 Conn_update 后的套接口

5 仿真结果分析

仿真网络模型如图 8 所示, 移动终端 MS 与网络中的服务器在通信期间从一个路由区域移动到另一个路由区域, IP 地址发生变化, 这期间路由器 1 可以将从网络端发送给 MS 原地址的数据通过隧道转发给新的网络接入点路由器 2, 路由器 2 再将数据发送给移动终端, 直到服务器 S 得到 MS 新的 IP 地址。使用 OPNET 进行仿真, 链路的传输速率是 9kbps, 链路误码率采用 dpt_error 模式(OPNET 自带的模式), 发包间隔服从均值为 0.05s。仿真主要是验证使用 MESOCK, IP 地址的切换对 TCP 应用的数据传输抖动和数据重传的影响。

图 9 是在此网络结构下 SIP 切换前后, 使用 MESOCK 情况下的 TCP 数据传输的延时抖动, 图 10 是使用 MESOCK 情况下的 TCP 数据传输的端到端延时, 可以看出延时和抖动相对于数据包长度在切换点附近稍微加大, 主要是因为 IP 地址变换引起数据传输路径的变化引起的, 切换过程中部分数据包有路径迂回造成, 与新旧接入点之间的传输速度也有关。但是从整个仿真过程来看, 切换期间的抖动和延迟并不比其它的时间段更显得突兀。图 11 是取不同的种子情况下的 12 次仿真中出现的重传的情况。可以看出 12 次仿真中有 4 次出现了重传, 但重传出现的位置比较分散, 离切换时间点很远, 因此同切换没有什么必然关系, 这说明 MESOCK 方案下切换没有影响 TCP 数据的重传。因此将此扩展套接字应用于 UDP 也不会影响数据发送接收。

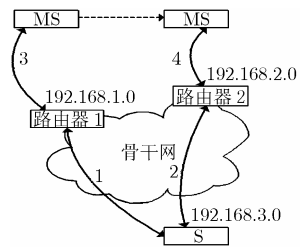


图8 仿真网络模型

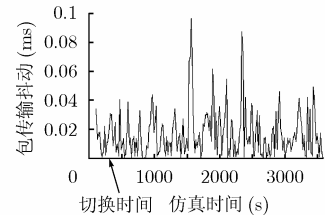


图9 MESOCK 下的 TCP 传输数据抖动

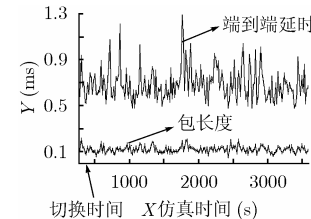


图10 MESOCK 下的 TCP 传输数据延时

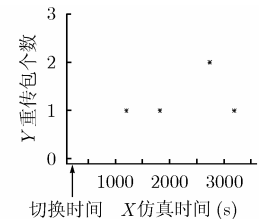


图11 MESOCK 下的 TCP 数据重传统计

6 结束语

NGN 中的移动性管理倾向于移动终端处于不同的路由区域获得不同的 IP 地址, IMS 作为其中的一部分也存在这个问题, 而 IP 地址的变换为应用的移动性和 TCP 连接的连续性带来了问题, MESOCK 方案则使得在不改变会话移动性管理信令流程的情况下, 通过对 SOCKET 的扩展, 可保证 TCP 连接的连续性, 而且不影响 SOCKET 的其它命令的使用, 对协议的改变很小, 比较容易实现, 程序员很容易接受, 同时对应用层以下的移动性连接没有要求。仿真结果也证明了 MESOCK 可以很好地支持应用的移动性。

参考文献

- [1] 3GPP, TS 23.228 V6.9.0. IP multimedia subsystem (IMS), 2005-03.
- [2] IETF, RFC 3261. SIP: Session Initiation Protocol, June 2002.
- [3] Nakajima N, Dutta A, Das S, and Schulzrinne H. Handoff delay analysis and measurement for SIP based mobility in

- IPv6. IEEE International Conference on Communications 2003(ICC'03), Anchorage, Alaska, USA, 11-15 May 2003, Vol 2: 1085-1089.
- [4] Banerjee N, Basu K, and Das S K. Hand-off delay analysis in SIP-based mobility management in wireless networks. Proceedings. for International Parallel and Distributed Processing Symposium, 2003.(IPDPS'03), Nice, France, 22-26 April, 2003: 224-231.
- [5] Kristiansson J and Parnes P. Application-layer mobility support for streaming real-time media. IEEE Wireless Communications and Networking Conference 2004 (WCNC 2004), Atlanta, GA USA, 21-25 March, 2004, Vol.1: 268-273.
- [6] Maltz D A and Bhagwat P. MSOCKS: An architecture for transport layer mobility. 17th IEEE International Conference on Computer Communications 1998 (INFOCOM'98), San Francisco, CA USA, 29 March-2 April 1998, Vol.3: 1037-1045.
- [7] IETF, RFC 2960. SCTPR. Stewart, Stream control transmission protocol, October 2000.
- 吕新荣: 女, 1976年生, 博士生, 研究领域为通信软件.
- 廖建新: 男, 1965年生, 教授, 博士生导师, 主要研究领域为通信软件、增值业务提供技术.
- 朱晓民: 男, 1974年生, 博士, 副研究员, 中国电子学会高级会员, 主要研究领域为智能网、下一代业务网络、协议工程.
- 武家春: 男, 1974年生, 博士生, 研究领域为智能网、通信软件.