基于云计算 Live Mesh 的流媒体应用研究

曹 彬 程久军 闫春钢
(同济大学计算机科学与工程系 上海 201804)
(同济大学嵌入式系统与服务计算教育部重点实验室 上海 201804)

摘 要 流媒体技术和云计算的迅速发展,以及越来越多的流媒体应用平台的涌现,极大地促进了网络用户间媒体信息的传播和共享,但业务上同时存在很多问题,例如,媒体播放在设备间快速切换和流媒体断点续播等问题。研究了一种基于云计算平台 Live Mesh 的流媒体应用系统。该系统采用 URI 定位和 XML 实现了媒体资源的集中管理,并采用局域网 UDP 广播与多线程处理机制实现了媒体播放状态在局域网内不同终端之间的即时切换;采用 Live Mesh 的消息管理机制,实现了流媒体的断点续播功能。实验结果表明,在不同终端之间可以实现相同媒体资源的共享和媒体播放一键即时切换,同时可以智能地实现断点续播。

关键词 云计算,Live mesh,流媒体,无缝切换,断点续播
中图法分类号 TP393 **文献标识码** A

Study for the Streaming Media Application Based on Live Mesh of Cloud Computing

CAO Bin CHENG Jiu-jun YAN Chun-gang
(Department of Computer Science & Engineering, Tongji University, Shanghai 201804, China)
(Key Laboratory of Embedded System and Service Computing of Ministry of Education, Tongji University, Shanghai 201804, China)

Abstract With the development of streaming media technology and cloud computing, and the emerging on more and more streaming application platform, transmission and sharing on steaming information have been facilitated. But there exist a lot of problems, for example, playing state handover and broken-point continuingly-playing, and so on. A streaming media application system based on live mesh of cloud computing was proposed in this paper. The system adopted URI and XML to accomplish media source centralize management, and used LAN UDP broadcast and multi-thread mechanism to realize real-time handover on steaming media playing state between terminals. At the same time, the broken-point continuingly-playing was carried out with message management mechanism based on Live Mesh. The experiments show that the sharing on the same streaming media resource and the seamless handover on playing state have been well implemented between terminals, and the broken-point continuingly-playing has also been intelligently achieved.

Keywords Cloud computing, Live mesh, Streaming media, Seamless handover, Broken-point continuingly-playing

1 引言

随着科技的发展,越来越多的电子设备进入人们的日常生活,其中移动电子设备,如 PDA(Personal Digital Assistant)、智能手机和上网本等都给用户带来了随身的娱乐体验。而普通 PC 在存储能力和处理能力上有较大的优势。如果能够把移动设备移动性、便携性的优势与 PC 处理能力强的优势有效地结合在一起,就可以为用户提供不间断的流媒体服务。例如,用户在下班的路上使用手机收看节目,到家后可以瞬间切换到播放效果较好的 PC 上继续观看。但是媒体文件在这些设备上互相之间是独立的,不同设备之间很难共享或者根本无法共享。为了让相同的文件可以在不同设备之间共享,用户必须手动将文件复制到不同设备上,这对于用户来说是一个低效、繁琐的过程,而新兴的云计算(Cloud Computing)可以有效地解决这些问题。

2 云计算及 Live Mesh 介绍

云计算是一种新兴的共享基础架构的方法,可以将巨大的系统池连接在一起以提供各种 IT 服务[1]。云计算的核心是海量数据的存储和计算。由几十万台甚至几百万台计算机构成的计算机群,对信息进行聚合和分布处理,然后通过网络对客户提供服务。这样,用户只需使用电脑、手机、PDA 等终端设备接入互联网,便可获取需要的信息服务。由于信息存储和数据计算都发生在"云端",云计算突破了传统的以个人

到稿日期:2009-12-23 返修日期:2010-03-05 本文受 863 国家重点基金项目(2009AA01Z401),国家自然科学基金(90718012),上海市基础研究重点项目(08JC1419300),微软亚洲研究院与同济大学网格中心合作项目资助。

曹 彬(1985-),男,硕士生,主要研究方向为云计算、海量数据处理等,E-mail:caobin1234@126.com;程久军 男,讲师,主要研究方向为移动 P2P 网络、移动社交网络等;闫春钢 女,教授,博士生导师,主要研究方向为 Petri 网理论、工作流建模方法、Web 服务。

计算机为核心的硬件限制，集合了信息聚合和设备聚合的全新 Web 服务，将从根本上改变人们获取信息、沟通交流的方式[2]。在云计算的网络应用模式中，数据只有一份，保存在“云端”，所有电子设备只需要连接互联网，就可以同时访问和使用同一份数据。

工业界很多公司，如 Google，Amazon，Microsoft 等，分别提出了自己针对云计算的理解，并用不同的技术来实现各自的云计算平台。Google 的云计算平台环境是私有的环境，除了开放有限的应用程序接口，例如 GWT(Google Web toolkit)，Google App Engine 以及 Google Map API 等以外，Google 并没有将云计算的内部基础设施共享给外部的用户使用[3]。Amazon 通过提供弹性计算云，满足了小规模软件开发人员对集群系统的需求，减小了维护负担，但是用户要为使用的资源付费[3]。微软的云计算是想把软件和各项程序有机地集合起来，并且作用于互联网上，这样就可以缩小软硬件之间的差距，并且达到资源共享。Live Mesh 是微软云计算战略的产品之一，是一款基于 Live Services 的“软件＋服务”产品，以“软件＋服务”的方式将计算机、手机和其他数字设备通过互联网整合到一起，允许用户同步访问、共享和存储文件，并支持移动设备和网络云计算，使客户常用的终端成为一个巨大的、能及时更新的“移动硬盘”[4-6]。Live Mesh 具有“云端”负责数据存储、不同的设备相互兼容、软件将不同的设备和网络连接起来等 3 个特点，从而很好地解决了相同文件在不同设备之间的共享问题。

本文设计了基于 C/S 结构的流媒体应用系统，以微软 Live Mesh 服务器为云，以移动终端设备为端，以互联网为传输介质，实现媒体资源在云端的存储及在终端的播放和切换，从而为用户提供一个无缝的流媒体应用系统。

3 基于 Live Mesh 流媒体应用系统

本系统是一个在. NET 平台下开发的 C/S 框架系统，服务器采用微软 Live Mesh 云计算平台，提供媒体文件存储、资源定位、消息管理、权限认证等服务；客户端自行设计，主要包括媒体资源管理和流媒体服务等模块，既可以在普通 PC 上运行又可以在装有 Windows Mobile 6.0 操作系统的移动终端上运行。系统整体架构如图 1 所示。

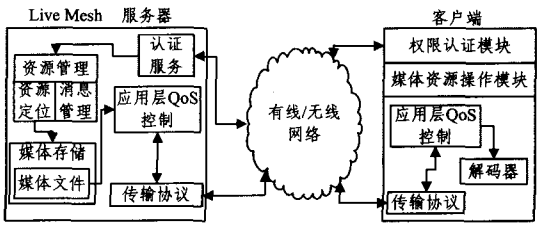


图 1 Live Mesh 流媒体应用系统结构图

3.1 资源管理模块

为了确保用户的信息安全，必须制定安全有效的认证机制。用户通过身份验证后，Live Mesh 根据不同的登录用户名，将他们定位到不同的空间，即每个用户在该模块中只可以看到并操作到属于自己的空间。在此模块中，目录以可视化的树形结构分层次显示，文件以表格形式分行列显示名称及属性，并为用户提供了对目录的创建、删除、改名等操作，以及对文件的上传、下载、删除、复制、改名等操作。Live Mesh 资源管理模块如图 2 所示，包括文件夹操作和文件操作两部分，具体过程如下。

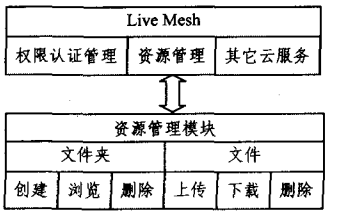


图 2 Live Mesh 资源管理模块图

3.1.1 文件夹操作

文件夹的创建、浏览和删除操作都是构造一个 xml 请求，通过 HTTP 把这个操作请求发送到服务器来完成的。例如一个新建文件夹的操作可以用以下 xml 文件描述：

```
〈entry xml:base="https://user-ctp.windows.net/V0.1/Mesh/"
      xmlns="http://www.w3.org/2005/Atom"〉
  〈title type="text"〉Folder Name〈/title〉
  〈summary type="text"〉Test Data〈/summary〉
  〈category term="LiveMeshFolder" label="LiveMeshFolder"
          scheme="http://user.windows.net/MeshObject"/〉
  〈category term="MeshObject" label="MeshObject" scheme="
          http://user.windows.net/Resource"/〉
  〈category term="LiveDesktop" label="Creator"/〉
〈/entry〉
```

其中 title 标签里的 Folder Name 表示新建文件夹的名称，〈category term="LiveMeshFolder" label="LiveMeshFolder" /〉中的 term 属性值“LiveMeshFolder”表示文件夹。

3.1.2 文件操作

文件操作主要包括文件显示和上传/下载。文件显示是指随着当前目录的切换，实现选中目录下文件信息的动态显示。其显示包括对文件的动态排序、分页、选择等，并可以按照用户的要求自定义其外观和行为。文件上传使用了. NET 平台提供的 FileUpload 控件[7]，该控件显示一个文本框和一个浏览按钮。当用户点击浏览按钮时，可以打开一个文件选择对话框，将客户端待上传的文件名显示在文本框中，以便用户上传。用户可以在本地设备上选择单个文件上传，也可以选择多个文件进行批量上传。只有当文件的类型是系统允许上传的类型，而且文件大小在系统规定之内，上传的目标路径存在，才能成功上传。文件的下载利用服务器的 Response 响应将服务器端的文件以文件流的方式发送到客户端。

由于文件上传/下载过程时间并不确定，本系统采用异步编程设计模式来实现文件的上传和下载。异步操作在主应用程序线程以外的线程中执行。应用程序调用方法异步执行某个操作时，应用程序可在异步方法执行其任务时继续执行[7]。我们使用异步操作，为上传或下载开辟一个线程，在后台执行；而在主应用程序中继续执行其他操作。

3.2 流媒体服务模块

3.2.1 流媒体播放功能

由于受网络带宽、计算机处理能力和协议规范等方面的限制，要想从 Internet 上下载大量的音频和视频数据，无论从下载时间和存储空间上来讲都是不太现实的。但流媒体技术的出现则很好地解决了这一难题。本部分使用流式传输技术，为用户提供实时、交互、按需点播的服务。

当用户要观看系统中的媒体文件时，首先要在终端上登

录，并通过 HTTP 请求发生器将请求发送到 Live Mesh。Live Mesh 接收到此请求后，向客户端发送当前存储的媒体文件的列表，并在客户端资源管理器中显示出来，用户可以根据此列表，来选择自己要观看的媒体文件。若用户选择了某一媒体文件进行观看，则由 Live Mesh 向客户端发送用于浏览该媒体文件的一系列图片，用户根据接收到的图片序列，选择自己感兴趣的片段进行在线观看，并可将下载播放的媒体数据保存到终端的存储器中。当 Live Mesh 接收到客户端停止观看的请求时，停止其对客户端的数据传输，并向客户端发送一条确认信息。

媒体播放器是使用 AxImp 工具生成的 wmp. dll 代码和微软公司提供的 Hosting Library 库[8]实现的，可支持 WMV 和 ASF 流媒体文件，并可以方便地进行播放控制。为了适应流媒体应用对实时性的要求，有效地缓解网络带宽瓶颈和保证流媒体播放质量，本部分在设计中采用了流媒体缓存技术[9-11]，在客户端设置了缓存。当播放器发现本地缓存中的视频播放完成，将等待缓冲；缓冲完成后将继续播放。由于文件下载速度难以估计，因此也就无法估计缓冲等待时间。在这里采用固定延时等待的方式，如果时间到了仍然不能得到有效的本地缓存，则继续等待。具体过程如图 3 所示。

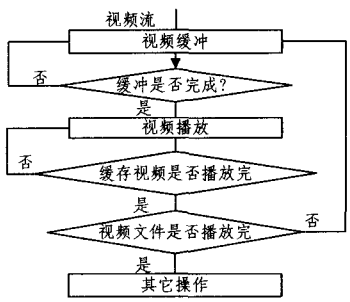


图 3　流媒体播放过程

3.2.2　流媒体播放在局域网设备间的切换过程

为了让用户享受到不间断的流媒体服务，本系统支持播放在不同终端设备的切换，即用户可以将当前终端设备的视频播放状态切换到局域网内的其它终端设备，并从中断点开始播放，例如从 PDA 切换到普通 PC。Live Mesh 支持多点登录，即同一个 Live ID 可以在不同的终端设备上同时登录。这一功能的出现方便了用户在不同终端之间切换。

由于多线程的并行处理可以提高程序的吞吐量，多任务的相互独立性也使程序在运行时间效率方面得到提高[12]，因此本部分采用多线程并行处理的思想进行设计。图 4 描述了流媒体播放在局域网设备间切换的实现过程，该部分主要通过 3 个线程完成，分别为：

(1) 负责收发广播的线程：该线程通过收发广播消息的形式，获得局域网内使用该系统的设备信息。将局域网内使用该系统的设备信息添加进本地链表；将局域网内退出该系统的计算机信息从本地链表中删除。广播消息线程使用 UDP 套接口，它与 TCP 套接口的主要区别在于通信双方不需要事先建立连接即可收发数据[13]。

(2) 负责监听端口建立连接的线程：该线程的功能类似于客户机/服务器模型中的服务器端，对局域网中的连接请求进行监听，对于新的连接请求建立新的 Socket 用于连接，并且对于不同的连接请求创建新的数据接收线程，用于处理连接后的消息传递以及文件传输等功能。

(3) 负责处理外部终端控制命令的线程：该线程用于该系统与用户的交互，处理外部终端的控制命令，并对于不同的命令实现相应的功能。

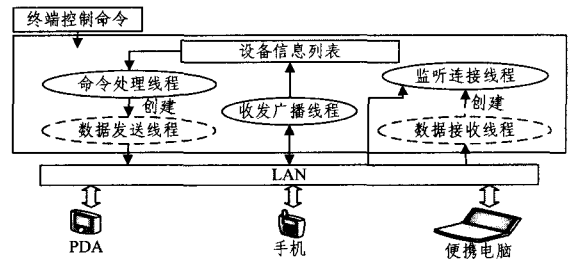


图 4　流媒体播放在局域网设备间切换的实现过程

播放切换需要解决的一个问题是媒体内容的同步。理论上，不同设备之间必须维护相同的媒体文件集合，以保证当前媒体文件的播放可以从一个设备自由地切换到另外一个设备。Live Mesh 是微软的云计算平台，具有“云端”负责数据存储、不同的设备相互兼容、软件将不同的设备和网络连接起来等 3 个特点，从而很好地解决了相同文件在不同设备之间共享问题。

图 5 描述了终端设备 1 和终端设备 2 之间的媒体播放切换的过程，前提是这两个终端设备都安装了本系统。播放终端 1 首先点击“切换设备”按钮，从弹出的本地设备信息链表中选择播放终端 2。此时负责监听端口建立连接的线程会建立两个设备之间的连接，并把播放终端 1 的同步消息传递给播放终端 2。播放终端 2 接收到消息后，启动自己的播放线程，并从播放终端 1 的中断点继续播放。输入参数包括当前用户、断点所处的时间位置、文件断点位置、文件 ID 等。例如用户 user 在播放视频 1. wmv 时，第 30 秒继续播放动作将被表示成：

SyncPlaying(user,30,2393256,urn:uuid:955f1ee7-4339-403c-b738-ab78b8d35d52)

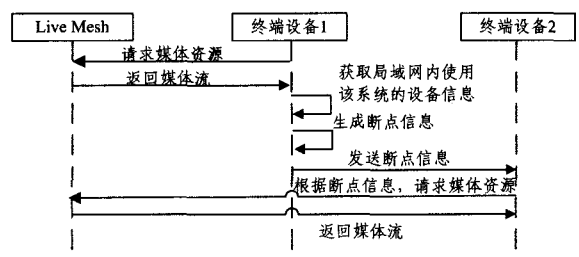


图 5　流媒体播放在设备间切换过程

3.2.3　流媒体断点续播功能

用户在欣赏视频时经常要用到的两个功能，即断点续播和定点播放。前者可以帮用户记住上次停止的位置，以便下次接着观看；后者则可以让用户直接从某个位置开始观看，如跳过电视连续剧的开头部分。很多流行的播放软件都提供了这两项功能，如暴风影音、KMPlayer 等。但是，如果用户观看在线视频，那么播放软件的断点续播功能就会受到限制，因为用户观看的不是本地视频文件，而是从网络上一边下载一边观看。

本部分可以在流媒体中断时记录下断点信息，并发送到 Live Mesh 服务器。一般情况下，一条断点信息包括用户名、视频名称、视频断点时间、文件断点位置、文件 ID 及其所在文件夹 ID。例如用户 username@hotmail. com 在播放视频

1. wmv时，第 79.088 秒的断点信息被表示成：

〈Text〉username@hotmail.com|1.wmv|79.088|4383678|urn:uuid:955f1ee7-4339-403c-b738-ab78b8d35d52|urn:uuid:42abb4d1-9ead-d2e9-e1db-cd45a4606e05〈/Text〉

Live Mesh 接收到断点信息后，以一条记录的方式进行存储。等到用户下次再看时，客户端先从 Live Mesh 服务器上读取流媒体文件的断点信息；然后使用正则表达式解析断点信息，根据断点信息重新定位视频播放位置；然后再等待缓冲；缓冲完毕，就可以从上次的结束位置开始观看。具体过程如图 6 所示。

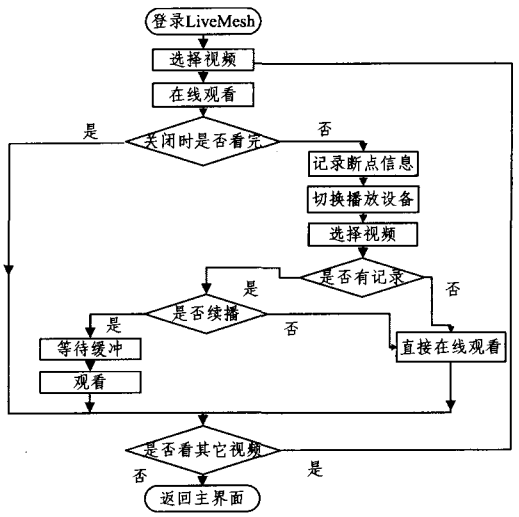


图 6　断点续播的流程图

4　实验结果与分析

本系统的实验环境如下：安装 Windows Mobile6.0 系统的 PDA 以及 Windows XP 系统的 PC，两设备之间通过 AP 访问 Live Mesh 服务器，进行流媒体通信。本系统所设计的 Live Mesh 资源管理器如图 7 所示。“资源管理器”可以显示 Live Mesh 上的所有目录和资源，方便用户清楚、直观地认识 Live Mesh 上的文件和文件夹。在“资源管理器”中还可以对文件进行各种操作，如打开、复制、删除、上传、下载等。若文件处于选中状态，用户将查看到选中文件的概要信息，并可播放音、视频文件以及预览图片文件。若双击目录中的文件夹，会自动获取其内容并显示在浏览区。

流媒体播放从 PDA 向 PC 切换的效果图如图 8、图 9 所示。当从 PDA 上点击“切换设备”按钮，即 11:09:44 时刻，在 11:09:46 时刻，PC 机上出现如图 9 所示的结果。切换的过程在以秒级计算的范围内，是用户可以接受的。

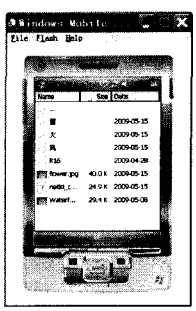


图 7　资源管理器界面

图 8　PDA 上流媒体播放及切换瞬间效果图

图 9　已切换到 PC 上的流媒体播放时刻

此外，从图 8 与图 9 的对比中不难发现，两个不同设备的画面质量明显不同。这是因为适合在手持设备上播放的媒体的码率普遍比较低，而同样的媒体在 PC 等分辨率较高的设备上播放时，类似于把小图片扩大。解决这个问题的一个有效方法是采用可伸缩性的视频编码，文献[11,14,15]在这方面做了研究。另外，还有其他方法，如将媒体文件中的视频码流转换为一个特定码率和图像尺寸的码流；或者把同一段视频内容编码生成多个具有不同码率和图像尺寸的码流，然后自适应选择一个最合适的码流传输给用户。

结束语　本文着眼于不同设备间流媒体播放快速切换以及断点续播等存在的问题，研究了一种基于云计算平台 Live Mesh 的流媒体应用系统。系统利用局域网 UDP 广播与多线程处理机制实现了流媒体在局域网内不同移动终端之间的即时切换；利用 Live Mesh 的消息管理机制，实现了流媒体的断点续播功能。实验结果表明，系统实现了媒体资源的集中管理和共享，在不同终端之间媒体播放可以一键即时切换，而且在用户可以接受的秒级范围内，这与传统切换方法分钟级的延时相比具有重要的使用价值；流媒体断点续播功能与传统方法相比不仅更智能，而且节约了用户时间。本系统实现了用户在不同时间、不同地点、不同设备间的媒体信息的传播和共享，体现了人性化、智能化的优点。

参 考 文 献

[1] Buyya, Yeo C S, Venugopal S. Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities[C]// Proceedings of the 2008 10th IEEE International Conference on High Performance Computing and Communications. Vol. 00, 2008: 5-13

[2] Armbrust M, Fox A, Griffith R, et al. Above the Clouds: A Berkeley View of Cloud Computing [DB/OL]. http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf, 2009-06-12

[3] 陈康，郑纬民. 云计算：系统实例与研究现状[J]. 软件学报，2009(5)

[4] Microsoft Corporation. What's inside Live Mesh[EB/OL]. http://www.mesh.com/Welcome/features/features.aspx

[5] Microsoft Corporation. Live Framework Query Model[EB/OL]. http://msdn.microsoft.com/en-us/library/dd136521.aspx

[6] Microsoft Corporation. Walkthrough Resource Model and Feeds [EB/OL]. http://blogs.technet.com/james/archive/2009/01/02/liveframework-walkthrough-guides.aspx

[7] Lowey J. Programming. NET components [M]. Sebastopol: O'Reilly, 2005: 155-182

[8] Microsoft Corporation. Hosting ActiveX Controls in the. NET Compact Framework 2. 0[EB/OL]. http://msdn. microsoft. com/en-us/library/aa446515. aspx

[9] Bommaiah E, Guo K, et al. Design and Implementation of a Caching System for Streaming Media over the Internet[C]// IEEE Real Time Technology and Applications Symposium, 2000: 111-121

[10] Hofmann M, Guo K, et al. Caching techniques for streaming multimedia over the internet[R]. BL011345-990409-04TM. 1999

[11] 尹浩，林闯，文浩，等. 大规模流媒体应用中关键技术的研究[J]. 计算机学报，2008(5)

[12] Beveridge J, Wiener R. Multithreading Applications in Win32 [M]. Boston: Addison-Wesley Professional, 1996: 397-413

[13] Stevens W R. TCP/IP 详解，卷 1：协议[M]. 范建华，等译. 北京：机械工业出版社，2000: 107-132

[14] de Cuetos P, Ross K W. Adaptive Rate Control for Streaming Stored Fine-Grained Scalable Video[C]// Proc. of NOSSDAV. Miami, Florida, 2002: 3-12

[15] 吴枫，李世鹏，张亚勤. 渐进、精细的可伸缩性视频码[J]. 计算机学报，2000, 23(12): 1276-1282

(上接第 85 页)

争时隙到来才能转发。

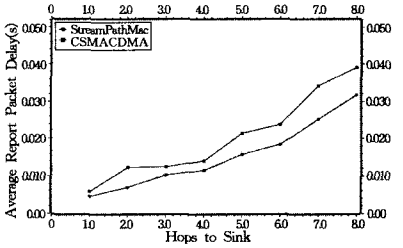
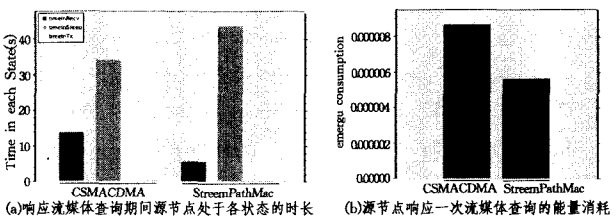


图 7 报告数据包的端到端时延随跳数的变化

基于簇的无线多媒体传感器网络中，簇头是一直处于工作状态的，只有成员节点会在无数据需要发送或接收时转入休眠。图 8(a)显示了本文协议与传统基于簇的 CDMA 与 CSMA 混合协议在响应一次流媒体查询期间某个满足条件的源节点处于各状态的时间长度。不难看出，在本文协议中源节点的休眠时间比前者长得多，而处于发送和接收模式的时间均明显较短。在此基础上，图 8(b)显示了本实验两种协议源节点的能耗对比，显然较长的休眠时间使本协议在源节点响应流媒体查询时能节约更多的能量。



(a)响应流媒体查询期间源节点处于各状态的时长 (b)源节点响应一次流媒体查询的能量消耗

图 8 响应一次流媒体查询源节点能耗对比

结束语 本文针对无线多媒体传感器网络提出了一种基于时隙预留的多信道实时 MAC 协议 StreamPathMac。该协议充分利用了流媒体数据的周期性特点，在源节点响应流媒体查询之前，事先建立一条从源节点到汇聚节点的流路径，路径上的簇头在每个时间帧均为该流预留特定的时隙，使得源节点的流媒体数据在每一跳都能被及时地接收并转发，最小化数据包在每一次转发时的信道接入时延。同时，由于时间表调度与簇间多信道机制的引入，延长了被查询源节点的睡眠时间，并且避免了流媒体数据包的碰撞冲突。仿真实验结果表明，本文协议明显减小了流媒体数据的端到端时延与时延抖动，并且达到了较好的能量有效性。虽然本协议对报告数据包的实时性稍有影响，并且流媒体查询时需要较长的响应时间，但这并不影响协议的整体性能。最后本协议在设计时并未考虑流媒体数据包的传输差错，传统基于请求重传的机制显然无法满足流媒体数据的实时性要求，因此如何针对流媒体数据包较长且可以容忍一定差错率的特点，结合本协议设计适合流媒体数据的可靠性机制，是我们将来进一步研究的课题。

参 考 文 献

[1] 罗武胜，翟永平，鲁琴. 无线多媒体传感器网络研究[J]. 电子与信息学报，2008, 30(6): 1511-1516

[2] 孙岩，马华东. 无线多媒体传感器网络 QoS 保障问题[J]. 电子学报，2008, 36(7): 1412-1420

[3] 李瑞芳，李仁发，罗娟. 无线多媒体传感器网络 MAC 协议研究综述[J]. 通信学报，2008, 29(8): 111-123

[4] Tavli B, Heinzelman W B. MH-TRACE: multihop time reservation using adaptive control for energy efficiency[J]. IEEE Journal on Selected Areas in Communications, 2004, 22(5): 942-953

[5] Guo C L, Zhong L Z C, Rabaer J M. Low power distributed MAC for ad hoc sensor radio networks[A]// Global Telecommunications Conference[C]. 2001: 2944-2948

[6] Li Cheng, Wang Pu, Chen Hsiao-Hwa, et al. A Cluster Based On-demand Multi-Channel MAC Protocol for Wireless Multimedia Sensor Networks[C]// Proc. IEEE ICC 2008. Beijing, China, May 2008: 2371-2376

[7] Akayya K, Younis M. An Energy-aware QoS Routing Protocol for Wireless Sensor Networks[A]// the Proceedings of IEEE Workshop on Mobile and Wireless Networks (MWN2003)[C]. Providence, RI(2003)

[8] Kulkarni P, Ganesan D, Shenoy P, et al. SensEye: A multi-tier camera sensor network[A]// Proc. of the13th Annual ACM International Conference on Multimedia'05[C]. ACM Press, 2005: 229-23

[9] Yonis O, et al. HEED: A hybrid, energy-efficient, distributed clustering approach for ad-hoc sensor networks[J]. IEEE Trans on Mobile Computing, 2004, 3(4): 366-379