

CMMB 转播系统^①

刘同兴^{1,2} 蒲宝明² 盛 娴^{1,2} (1. 中国科学院研究生院 北京 100049;
2. 中国科学院沈阳计算技术研究所 辽宁 沈阳 110171)

摘 要: CMMB 是英文 China Mobile Multimedia Broadcasting (中国移动数字多媒体广播) 的简称。目前 CMMB 的主要业务为视频广播业务, 因为受终端硬件的约束, 目前只能在配有 CMMB 模块的设备上观看 CMMB 视频节目。提出了一种基于 CMMB 系统的数字电视转播系统, 阐述了系统架构以及各部分的设计与实现, 并对 H.264、AAC 的 RTP 传输、解码、同步表现等关键技术与算法进行了详细的说明。经实验证明, 该系统能够完成 CMMB 数字电视转播, 并且能够使多个用户在 Windows, Linux, Symbian 三类操作系统的终端下共享收看到数字电视节目。扩展了通用终端的功能, 节约了用户的经济投入, 并且解决了 CMMB 室内覆盖不足的缺点。

关键词: CMMB; 视频广播; H.264; RTP; Directshow; Symbian

CMMB-Based TV Broadcasting System

LIU Tong-Xing^{1,2}, PU Bao-Ming², SHENG Xian^{1,2}

(1. Graduate University of Chinese Academy of Sciences, Beijing 100049, China; 2. Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110171, China)

Abstract: CMMB is the abbreviation of China Mobile Multimedia Broadcasting. At present, CMMB system mainly provides video broadcasting service. With the restriction of hardware, only the terminal with the CMMB module can receive CMMB video program. A digital TV broadcasting system based on CMMB is presented. The framework of system and the design of all parts are expounded. The implementations of the key technologies and algorithms are explained in detail. Experiments show the system is able to accomplish broadcasting of the digital TV based on CMMB and share watching digital TV program under Windows, Linux, Symbian operating systems. It extends the function of general intelligent device and reduces the costs and resolves the problem of CMMB poor indoor signal coverage.

Keywords: CMMB; video broadcasting; H.264; RTP; directshow; symbian

1 引言

随着数字广播技术的迅速发展, 移动多媒体广播已成为炙手可热的发展热点^[1]。作为媒体传播的一种新媒体形式, 移动多媒体广播的最大优势是支持移动接收。国际现有的移动多媒体广播标准包括欧洲的 DVB-H、韩国的 T-DMB 以及美国高通公司的 MediaFLO 标准等, 然而我国复杂地貌产生的覆盖效率与性能问题, 庞大的用户群体带来的成本与安全问

题, 以及使用国外标准须缴纳的巨额专利费用问题使得现有国际标准不适用于中国^[2]。

中国移动多媒体广播(CMMB)系统是我国自主创新研发的移动多媒体广播系统, 是广电总局在全国推行的标准。它根据我国具体国情, 采用“天地一体, 星网结合, 统一标准, 全国漫游”技术体系, 解决了以上问题^[2]。目前我国现已完成了全国所有地市的 CMMB 网络覆盖, 形成完整的 CMMB 产业链条, 该

^① 收稿时间:2010-03-04;收到修改稿时间:2010-03-26

标准已进入商业运营阶段。

目前市场上的主流终端均为国际化大品牌厂商制造，他们的产品开发是着眼于全球市场的。这些着眼于全球市场的终端很少集成中国的 CMMB 接收播放功能。CMMB 转播系统弥补了终端的此项不足。另外 CMMB 转播服务端通过 Wi-Fi、蓝牙等无线接入方式与终端连接，使得用户可以共享一套 CMMB 接收与解密设备收看数字电视节目，节省了用户的经济投入。最后可以通过使用高灵敏度天线或安置在室内外信号好的地方来提高 CMMB 接收效果，解决了目前 CMMB 室内覆盖不足的缺点。

本文研究了 CMMB 技术标准、H.264 视频编码与 AAC 音频编码的码流结构与 RTP 打包传输技术、DirectShow 技术以及 Linux 与 Symbian 操作系统下相关技术，设计并实现了 CMMB 转播系统。

2 系统方案设计

CMMB 转播系统由 PC 平台的服务端与 Windows, Windows CE/Mobile, Linux, ARM Linux, Symbian S60 3rd 几个平台下的客户端组成。如图 1 所示。

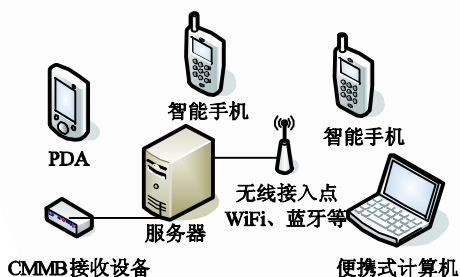


图 1 系统组成框图

服务端主要包括 CMMB 接收部分与转播两部分。CMMB 接收部分使用泰合志恒公司的 TP3001B 开发套件，该开发套件包括 TP3001B EVB 开发板、USB_

SPI 转换板、TP3001SDK_V2.7、相关驱动程序与参考文档。转播部分采用 RTP，视频遵循 RFC3984 打包方法，音频遵循 RFC3016 打包方法。使用相对时间戳技术保证媒体间同步[3]。

客户端主要包括音视频的接收、解码、同步播放三部分。需要在 Windows, Linux, Symbian 三类操作系统下实现，考虑代码复用，将 RTP 打包、解包，缓冲单向链表，缓冲队列等设计成标准 C++ 类库，在多平台共享使用。各操作系统使用的流媒体框架、开发平台、库、音视频输出方式如表 1 所示。

3 服务端的设计与实现

3.1 硬件组成

CMMB 转播服务端由 CMMB 信号处理模块、CAS(条件接收系统)终端系统与主机组成。使用的是泰合志恒公司的 TP3001B 大卡方案。如图 2 所示。

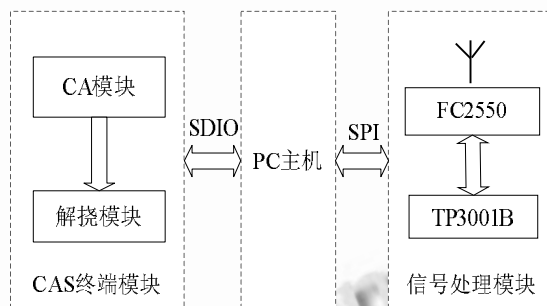


图 2 服务端的组成框图

CMMB 信号处理模块主要由 TP3001B 解调器与 FC2550 调谐器构成。TP3001B 解调器通过 I2C 接口控制 FC2550 调谐器，使用 SPI 接口输出 CMMB 复用子帧流给主机。主机对复用子帧解复用，并通过 SDIO 接口将 CMMB 音视频扰流数据以及授权控制信息 (ECM) 以及授权管理信息 (EMM) 发送给 CAS 终端模块，最终从 SDIO 接口返回清流数据。

表 1 客户端框架、开发平台、库、输出方式

操作系统	流媒体框架	开发平台	RTP 库	视频解码库	音频解码库	视频输出	音频输出
Windows	DirectShow	Visual Studio 2008	JRTPLIB	ffdshow	monoAAC	Video Render	Audio Render
Windows CE/Mobile	DirectShow	Visual Studio 2008	JRTPLIB	ffmpeg 0.5	ffmpeg 0.5, faad2-2.7	Video Render	Audio Render
Linux(X86, ARM)	无	Eclipse CDT	JRTPLIB	ffmpeg 0.5	ffmpeg 0.5, faad2-2.7	SDL	SDL
Symbian	MMF	Carbide.c++	自封装	ffmpeg 0.5	系统解码器	直接写屏	直接输出

3.2 软件设计与实现

服务端软件可以为驱动模块、解复用模块、条件接收解扰模块和转播模块^[4]。驱动模块实现对 TP3001B 硬件的控制以及数据的获取。解复用模块实现对复子用帧流的解析。条件接收解扰模块实现扰流数据的解扰。转播模块将 CMMB 清流中的音视频部分与电子信息服务指南(ESG)使用自定义的包格式广播到局域网中。

3.2.1 驱动模块

驱动模块向控制模块提供硬件控制的接口，它的主要功能是实现 PC 端与 TP3001B 解调器数据通信，数据通信采用三层协议栈模型。最底层的为物理层，采用 SPI 接口协议，其中 PC 为 SPI 主设备，TP3001B 为 SPI 从设备。第二层为传输层，采用 SDIO 协议，通过向 TP3001B 发送 CMD0, CMD5, CMD52, CMD53 四种指令以及接收返回数据实现主从设备的通信。第三层为应用层，使用的是 TP3001B 芯片自定义的协议^[4]。

TP3001B 开发套件中提供了 TP3001B 芯片在 Windows 下的驱动程序 SdioSpiBus, SdioSpiBus 包括 F321 与 usbiox 两种硬件下的实现。本系统采用了 F321 硬件方案。

3.2.2 解复用模块

在移动多媒体广播系统中，复用的功能是完成音频、视频、数据、电子业务指南等信息的封装和排列，解复用是复用的逆过程，它的功能是将复用帧中的音频、视频、数据、电子业务等信息解封装，分离出来^[5]。文献 5 定义了 CMMB 复子用帧封装格式，如图 3 所示。

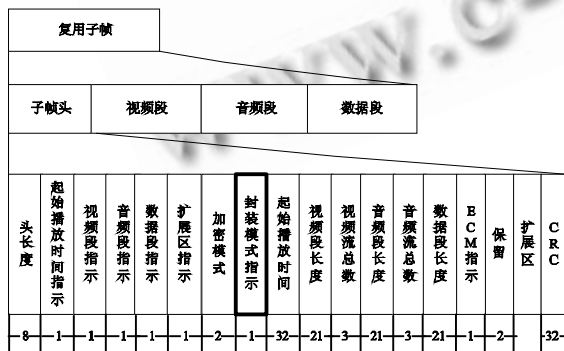


图 3 CMMB 复子用帧封装格式^[5]

TP3001SDK_V2.7 的 frameparser 模块实现了解复用的相关函数。其中 ParseMuxUnit 为解析封装

模式 2 的复用块函数, ParseH264Nal 为解析 H.264 视频的 NAL 单元函数, ParseLatmAudioUnit 为解析 AAC 音频的 LATM 单元函数, ParseEsgData 为解析电子信息服务指南(ESG)数据函数。

3.2.3 条件接收解扰模块

条件接收(CA)是目前 CMMB 加密授权系统所进行授权控制管理的主要方式。条件接收系统(CAS)使用加扰器对内容进行加扰,并通过条件接收系统的授权管理信息(EMM)、授权控制信息(ECM)对节目密钥进行加密传输。终端设备解扰过程包括以下两个步骤:(1)通过 EMM 与 ECM 解扰出节目密钥;(2)使用节目密钥将加扰的节目码流解密为清流。

目前 CMMB 终端的 CA 解扰有三种方案:大卡方案、小卡方案和 SMD 方案。大卡方案与 SMD 方案在解密卡中实现了解扰过程的两个步骤,SMD 方案直接将 SMD 芯片焊接在 PCB 板上。小卡方案在解密卡中只实现了解扰过程的第一步,第二步由主机上的软件实现。本系统大卡方案 CA 驱动部分由武汉天喻信息产业股份有限公司提供。

3.2.4 转播模块

转播模块按照 RFC3984 对 H.264 视频流打包,聚合包采用不包含 DON(解码顺序号)的单时间聚合包 STAP-A 方式,分片采用 FU-A 方式^[6]。按照 RFC3016

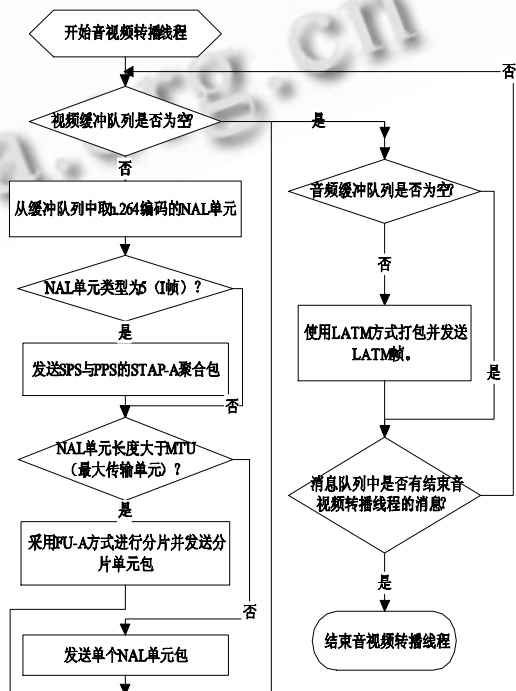


图 4 音视频打包发送流程图

中 LATM 封装格式对 AAC 音频流打包, 其中 Audio SpecificConfig 信息采用带外传输方式^[7]。打包流程如图 4 所示。最后使用 RTP 协议在局域网中组播音视频码流, RTP 的实现采用 JRTPLIB 开源库。发送端同步采用相对时间戳技术, 相同时间表现的音视频单元打上相同的 RTP 时间戳^[3]。

4 客户端的设计与实现

4.1 总体设计

CMMB 播放客户端的主要功能是流式播放服务端发送的 CMMB 清流节目。音视频单元在客户端经过接收、解码、同步处理, 最终表现给用户。客户端按功能可划分为: 音视频 RTP 接收模块、H.264 视频解码模块、AAC 音频解码模块、PCM 音频输出模块、YUV420 视频显示模块、音视频同步控制模块以及 UI 模块。其核心部分框图如图 5 所示。框图表示了音视频单元传递的流程, 音视频同步原理以及各模块之间的关系。

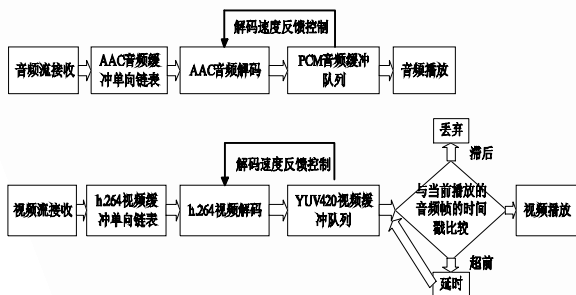


图 5 CMMB 播放客户端核心部分框图

以视频部分为例, 框图中使用了两个缓冲队列, 相对文献 3 中的系统多了一个解码前的缓冲单向链表, 这样做的目的有两个: 1. 本系统的客户端大多在嵌入式平台下, 内存通常不大, 增加解码前缓冲可减少内存的占用率; 2. 在多路由情况下, UDP 报文可能会乱序, 增加解码前的缓冲单向链表可以方便实现数据包在解码前的排序, 以实现媒体内同步。另外可以根据解码后的缓冲队列中的视频单元数, 来调节解码器的速度。当缓冲队列中的视频单元数目比较多时, 在解码空闲时调用 Sleep 函数减慢解码速度, 当缓冲队列中视频单元数目比较少时, 不调用 Sleep 函数加快解码速度。

图 6 为视频部分的类图设计。工厂类 CH264

Factory 实例化了 RTP 接收类 CH264Receiver、H.264 解码类 CH264Decoder、YUV420 显示类 CYV12Display, 并通过传递参数建立了各类之间的依赖关系。

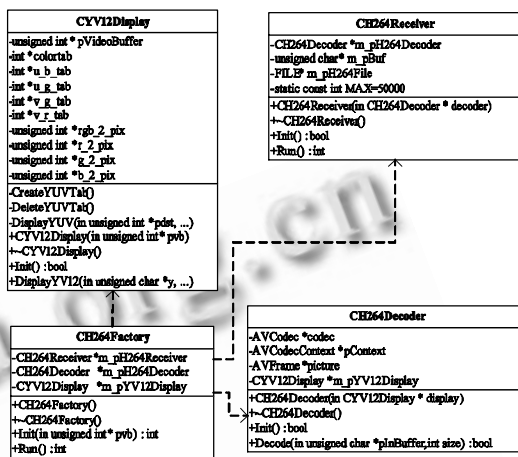


图 6 H.264 视频播放部分类图

Windows 平台下的实现因为使用的框架的特殊性, 与设计部分有所不同。

4.2 音视频媒体间同步控制算法

如 3.2.4 小节所述, 音视频在组播发送时将相同时间表现的音视频单元打上了时间戳, 因此在客户端需要将此时间戳传送给音视频播放部分, 完成媒体间的同步。由于听觉的不连续比视觉对图像不连续的敏感度高, 因此选择音频流为主流, 视频流为从流, 音频流连续播放, 通过调整视频播放过程, 实现媒体间同步控制^[3]。

从图 5 可以看出音视频的控制放在了解码之后, 这样就将网络抖动时延与解码时延作为一个整体进行了处理。本文的音视频媒体音同步控制算法^[3]如下

1. 按照音频的比特率从音频缓冲队列中取出音频帧, 送入音频输出设备进行回放, 并记录当前播放音频帧的时间戳 TA;

2. 从视频缓冲队列中取出视频帧, 该帧的时间戳为 TV。规定音视频不同步的容忍度为 TC=120ms。如果 $|TV-TA| \leq TC$, 则正确播放此视频帧。如果 $TV-TA > TC$, 表示该视频帧超前, 仍保持该视频帧在队列头。如果 $TA-TV > TC$, 表示该视频帧滞后, 则从队列中删除该视频帧, 丢弃该视频帧。

4.3 音视频 RTP 解包的设计与实现

3.2.4 小节中按照 RFC3984 与 RFC3016 实现了

音视频打包, RTP 解包是打包的逆过程。解包的流程如图 7 所示。

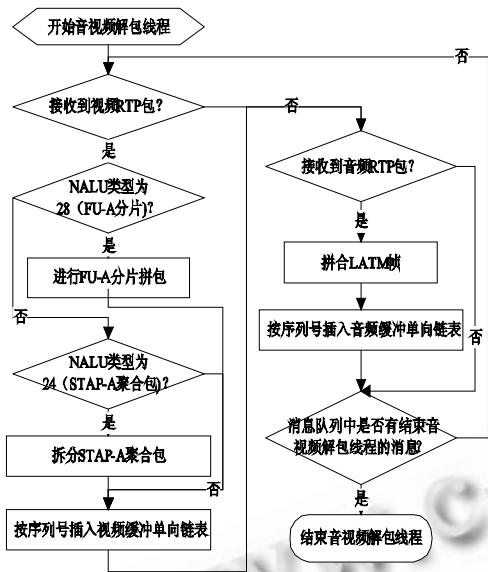


图 7 音视频解包流程图

4.4 各操作系统下 CMMB 客户端实现的关键技术

4.4.1 Windows 与 Windows CE/Mobile 操作系统

Windows 与 Windows CE/Mobile 操作系统使用的流媒体开发框架为 DirectShow, DirectShow 的基本单元为 Filter(过滤器), 核心部分为 Filter Graph Manager。本课题在 Visual Studio 2008 下实现了 CMMB 接收源过滤器、H.264 解码过滤器、AAC 解码过滤器, 并完成了软件界面设计。Windows 系列 CMMB 播放客户端的过滤器图如图 8 所示。

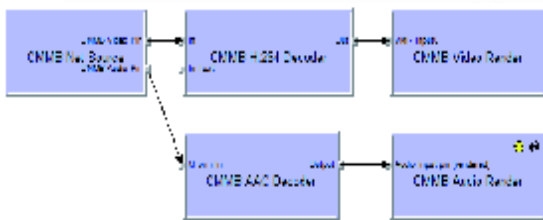


图 8 Windows 系列 CMMB 播放客户端 DirectShow 过滤器图

CMMB Net Source 包含 RTP 接收模块、RTP 解包模块, 并通过 Sample 的方式将 H.264 的 Nal 单元与 AAC 的音频单元传递给 H.264 解码器与 AAC 解码器。RTP 接收模块使用 JRTPLIB 开源库实现 RTP 包的接收, 实现部分与 4.4.2 节 Linux 平台类似。

CMMB H.264 Decoder 为 H.264 解码过滤器。Windows 平台下有很多商业或开源 H.264 解码过滤器, 如: CoreAVC、CLAVC、ffdshow、XVID H.264 Decoder 等。但在 Windows CE/Mobile 平台, 却少有开源的 H.264 解码过滤器, 因此本课题封装 ffmpeg 0.5 实现了 H.264 解码过滤器, 解码实现部分与 4.4.2 节 Linux 平台类似。

CMMB AAC Decoder 为 AAC 解码过滤器。如同 H.264 解码器, Windows 平台下有很多商业或开源的 AAC 解码器, 如: CoreAAC, monoAAC 等。但在 Windows CE/Mobile 平台, 却少有开源的 AAC 解码过滤器, 因此本课题封装 faad2-2.7 实现了 AAC 解码过滤器, 解码实现部分与 4.4.2 节 Linux 平台类似。

CMMB Video Render 与 CMMB Audio Render 为系统的视频输出与音频输出过滤器。只要对相同时间表现的音频视频帧以同样的时间戳传送给这两个过滤器, 就可以很容易的实现音视频的同步。

4.4.2 Linux 与 ARM Linux 操作系统

Linux 的 CMMB 播放客户端在 Opensuse 11.1 + Eclipse 环境下开发完成, Eclipse 安装使用了 cdt-master-6.0.1.zip, eclipse-cpp-galileo-SR1-linux-gtk.tar.gz 两个安装包。客户端 ARM 开发平台选用了天嵌公司的 SKY2440 开发板与交叉编译工具链 EABI-4.3.3_EmbedSky_20091210.tar.bz2。CMMB 播放客户端实现时使用了 jrtpplib-3.7.1, jthread-1.2.1, ffmpeg-0.5, faad2-2.7, SDL-1.2.14, pthread1.0 等开源库。开发步骤为: 1. 软件模块划分与 UML 设计; 2. 在本机上实现 PC 平台的软件并完成软件的测试; 3. 移植依赖的开源库到 ARM9 平台; 4. 将 PC 平台的软件源码交叉编译移植到 ARM9 平台; 5. 完成 ARM9 平台的软件测试。

音视频 RTP 接收模块中从 RTPSession 类派生出 MyRTPSession 类, 重写 MyRTPSession 类中的 OnPollThreadStep 函数实现 RTP 包的接收。程序中首先对 MyRTPSession 类进行实例化并进行初始化设置。当有 RTP 包到来时, JRTPLIB 库会自动调用 OnPollThreadStep 函数, 在此函数中根据 RTP 的载荷类型判断 RTP 包是音频包还是视频包, 并将包插入到音频解码缓冲单向链表或视频解码单向链表。注意在插入缓冲单向链表之前需要判断链表中节点个数是

否达到预定义的最大个数，如果达到则需要执行一次清缓冲单向链表操作。

H.264 视频解码模块在 CH264Decoder 类的构造函数中调用 avcodec_init, avcodec_register, avcodec_find_decoder, avcodec_alloc_context, avcodec_open, avcodec_alloc_frame 实现 ffmpeg 的初始化，在解码函数 Decode 中调用 avcodec_decode_video 实现解码，通过判断返回值是否大于 0 来判断是否正确解码出一帧图像。然后将解码出的 YUV420 格式的图像帧传递给 YUV420 视频显示模块。最后在 CH264Decoder 类的析构函数中调用 avcodec_close 与 av_free 完成 ffmpeg 的反初始化。

AAC 音频解码模块与 H.264 视频解码模块结构相似，都是调用 ffmpeg0.5 完成解码，再将解码完的帧传到缓冲单向链表中，在此不再赘述。与 H.264 解码不同的是 ffmpeg0.5 中默认不包含 AAC 的解码库，所以需要先编译好 faad2-2.7，并在配置 ffmpeg0.5 时加入 --enable-libfaad 参数，再编译 ffmpeg0.5，使得 ffmpeg 支持 AAC 解码。

PCM 音频输出模块与 YUV420 视频显示模块都由 SDL 库实现。在完成必要的初始化后，最后通过 SDL_MixAudio 与 SDL_DisplayYUVOverlay 将音频与视频表示出来。

4.4.3 Symbian 操作系统

Symbian 操作系统的 CMMB 播放客户端的集成开发环境为 Carbide.c++ v2.0，它是基于 Eclipse 的 Symbian OS C++ 开发工具。另外还需要安装 ActivePerl, JRE, S60 SDK, Plugin for S60 以及 Nokia PC 套件。

Symbian 平台下的 CMMB 播放客户端整体框架与 Linux 下相同。RTP 库部分在 UDP 基础上实现，RTP 的报文是 UDP 的载荷，它的基本头部有 12 个字节^[8]。程序中按照 RTP 的格式对 UDP 载荷解析即可^[8]。Symbian 平台不推荐使用 C 语言编程，但仍然提供 libc 库。因为 ffmpeg0.5 库遵循 C99 标准，所以在 libc 的支持下，ffmpeg 容易被移植到 Symbian 平台。Symbian 平台为了提高系统的效率不推荐使用多线程编程，而改由活动对象(Active Object)编程。因此需要将多线程部分改由活动对象实现。



图 9 Symbian S60 3rd 平台下的客户端截图

音视频 RTP 接收模块中除 RTP 实现使用 RSocket 外，其它部分与 Linux 下几乎相同。H.264 视频解码模块的实现也与 Linux 几乎相同。AAC 音频解码模块改由系统解码器实现，通过配置一系列参数，再调用解码对象的 ProcessL 方法实现 AAC 的解码。PCM 音频输出模块与 YUV420 视频显示模块通过调用 CMdaAudioOutputStream 与 CDirectScreenAccess 两个类实现。需要注意的是 Symbian 中每个像素中颜色的排列顺序为 BGR565 与其它平台的 RGB565 不同。图 9 为 Symbian 操作系统下的 CMMB 播放客户端在 Nokia 5320XM 手机上的运行界面。

5 结论与展望

为了解决在通用终端设备上不能收看 CMMB 电视节目的问题，提出了一种转播方案。在此方案的基础上详细论述了系统的架构、关键技术。最后在 Windows 平台实现了转播服务端，并在 Windows, Windows CE/Mobile, Linux, ARM Linux, Symbian 几个操作系统下实现了播放客户端。通过实验测试，此系统运行稳定正确。

下一步，将在嵌入式平台实现转播服务端，节约服务端能耗，提高系统使用效率。同时，将进一步研究多路视频转播时，如何提高网络传输的效率。

参考文献

- 1 解伟.移动多媒体广播系统与标准.现代电信科技, 2008,6(6):22-29.
- 2 解伟.移动多媒体广播 CMMB 技术与发展.电视技术,

(下转第 51 页)

(上接第 37 页)

- 2008,32(4):4—7.
- 3 董春兵.音视频同步的设计与实现[博士学位论文].长春:吉林大学, 2005.
 - 4 陈智.基于 CMMB 标准的移动多媒体广播接收与播放系统开发[硕士学位论文].厦门:厦门大学, 2008.
 - 5 国家广播电影电视总局.GY/T 220.2-2006.移动多媒体广播第 2 部分:复用.北京: 国家广播电影电视总局, 2006.
 - 6 Wenger S, Hannuksela MM, Stockhammer T, Westerland M, Singer D. RTP Payload Format for H.264 Video, RFC 3984, February 2005.
 - 7 Kikuchi Y, Toshiba, Nomura T, NEC, Fukunaga S, Oki, Matsui Y, Matsushita, Kimata H. NTT, RTP Payload Format for MPEG-4 Audio/Visual Streams. RFC 3016, November 2000.
 - 8 Schulzrinne H, Casner S, Frederick R, Jacobson V. RTP: A Transport Protocol for Real-Time Applications, STD 64, RFC 3550, July 2003.