

# 高速移动场景下的平稳宽带车载 视频点播系统设计

邓博轩<sup>1</sup>, 李学丰<sup>2</sup>, 闫尉深<sup>3</sup>

(1. 长安大学 信息工程学院, 西安 710000;

2. 华录易云科技有限公司, 南京 211800;

3. 河北省高速公路延崇管理中心, 河北 张家口 075000)

**摘要:** 针对列车乘客在接入路侧蜂窝移动网络时因多普勒效应而引起的带宽稳定性差的问题, 设计了一种高速移动场景下的平稳宽带车载视频点播系统, 用户可在手机客户端通过 WiFi 连接固定于车厢内的车载服务器以下载或上传本地视频资源; 此设计中, 通信两端均在车厢内部, 处于相对静止状态, 因此可有效避免多普勒效应; 同时, 使用了定向增益天线来增强信号在车厢窄长矩形区域的有效覆盖; 最终, 实验数据表明: 该系统在  $30 \times 10$  m 的室内场景下, 移动终端访问的实时传输带宽基本在 35 Mbps 到 56 Mbps 之间且波动较小, 可实现稳定流畅的视频服务。

**关键词:** 多普勒效应; 带宽稳定性; 车载服务器; WiFi; 视频服务

## Design of Stable Broadband In-vehicle Video-on-demand System in High-speed Mobile Scenarios

DENG Boxuan<sup>1</sup>, LI Xuefeng<sup>2</sup>, YAN Weishen<sup>3</sup>

(1. School of Information Engineering, Chang'an University, Xi'an 710000, China;

2. Hualu Yiyun Technology Co., Ltd., Nan'jing 211800, China;

3. Hebei Expressway Yanchong Management Center., Zhang'jiakou 075000, China)

**Abstract:** When the mobile terminals of train passengers are connected to the mobile network, the Doppler effect will make the bandwidth stability worse. To resolve this problem, a stable broadband in-vehicle video-on-demand system in high-speed mobile scenarios is designed. Users can download or upload the local video resources by connecting to the on-board server fixed in the car through the WiFi on the mobile client. Due to the relatively static on both ends of the communication, the Doppler effect can be alleviated effectively. Furthermore, the directional antenna is employed to meet the strip rectangular area of carriage. The experimental data shows that, in an indoor area of  $30 \times 10$  m<sup>2</sup>, the real-time transmission bandwidth accessed by the mobile terminal is basically between 35 Mbps and 56 Mbps, and the fluctuation of transmission bandwidth is small, which can achieve stable and smooth video service.

**Keywords:** doppler effect; bandwidth stability; on-board server; WiFi; video service

## 0 引言

在高速移动的列车车厢内, 用户终端能稳定地获取视频资源是重点研究内容。目前主流的接入网络方式包括 4G/5G 蜂窝移动通信网络和 WiFi 无线接入。5G 源于移动通信领域, 运用超密集组网和大规模天线阵列等关键技术<sup>[1]</sup>, 可大幅提高数据传输速率, 但是由于载波频率过高, 毫米波波段的绕射能力下降, 只有密集部署基站才能在室内等障碍物多的场合取得良好覆盖效果, 但这在用户较少的地方难以取得良好的经济效益。另外, 在高速移动的火车上, 由于 5G 的波长较短, 高速移动列车和基站之间的相对运动产生的多普勒频移也要大于 4G; 而且, 沿途变化的地形地貌也会加剧传输信道的时变性, 为信道估计提出了

更大的挑战。因此, 目前将 5G 应用于列车上还存在一些困难, 尚没有较好的解决方案。

WiFi 源于计算机网络, 属于成熟的移动端无线接入技术, 由于覆盖范围和工作载频的限制, 基本属于静态的非移动场景下的通信模式。WiFi 通常由连接有有线光纤网络的路由器提供, 不支持终端的快速移动。此外, 当前热门的无线 CPE, 采用大规模天线阵列、正交频分复用等关键技术, 在较好的高速条件下, 具有允许接入移动网络的能力, 但获取的网络带宽不太稳定。

无线局域网下的嵌入式视频点播系统旨在解决网络带宽不稳定所导致的视频播放卡顿的问题, 文献 [2] 提出的网络视频点播系统支持视频媒体编解码, 客户端通过以太

收稿日期: 2022-03-15; 修回日期: 2022-05-07。

作者简介: 邓博轩(1995-), 男, 河北石家庄人, 硕士, 主要从事智能交通与信息系统方向的研究。

引用格式: 邓博轩, 李学丰, 闫尉深. 高速移动场景下的平稳宽带车载视频点播系统设计[J]. 计算机测量与控制, 2022, 30(11): 179-185.

网接口与视频服务器建立网络连接从而进行数字媒体传输,但该设计采用的通信方式不支持高速移动的车载设备终端且文献中没有对视频传输过程中的网络带宽稳定性进行测试。文献 [3] 提出的流媒体视频点播系统,重点对原始视频的预处理进行了改进。采用多编码技术协作的无损压缩方案,使得视频数据传输延迟减小从而保证视频播放的流畅性。但文中仅限于对视频点播器的功能效果进行测试,并未对视频传输速率、网络的稳定性等系统性能进行有效分析。文献 [4] 提出的视频点播技术采用 HTTP 流媒体传输协议与 FFmpeg 开源库相结合的方式实现视频信息在终端平台的展示,但对于文中提出的通过减小播放切片个数和切片时长来降低视频播放延迟的理论方案还缺乏系统性的验证与分析。

针对列车旅客在 4/5G 网络下使用手机浏览视频时网络稳定性差的问题,本文采用一种次优的解决思路,即在通信网络良好的时候,尽可能缓存一些流媒体数据到本地存储器中,以便在网络不稳定时通过释放缓存来延缓卡顿现象,同时可根据用户选择的热度,确定视频资料的保存时效。

本文提出了一种高速移动场景下的平稳宽带车载视频点播系统,用户无须下载客户端,采用通用的 HTTP 协议,增加了移动终端与服务器设备连接的灵活性,缩短了临时旅客的接入时间。在技术开发方面,使用开源的 Linux+Apache+Mysql+Php (LAMP) 架构进行软件开发,在高速移动和多用户的环境下充分利用 LAMP 具有的 Web 资源丰富、高性能、高并发等优势,支持多个用户同时点播视频。系统提供稳定的网络环境,可实现高清视频的流畅播放,使乘客拥有良好的观影体验。

### 1 系统设计与硬件平台

视频点播系统由客户端和服务端组成,客户端为手机或笔记本电脑等移动终端;车载服务器安装在列车车厢内,其搭载无线 WiFi 模块和以太网 MAC 控制器,用于构建车载无线通信网络,外接存储单元存储视频资源,本设计的物理拓扑结构如图 1 所示。

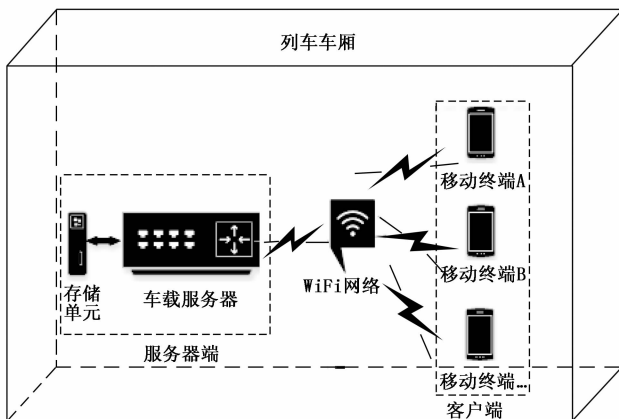


图 1 物理拓扑结构图

本系统分别从移动客户端和车载服务器两方面进行设计,如图 2 所示为系统设计框图。客户端程序是基于浏览器而设计,由于用户终端安装的浏览器不同,故需要对浏览器的类型进行自适应识别。利用 HTML5/Flash 自适应播放应用<sup>[5]</sup>,播放器可以自动侦测到浏览器类型。

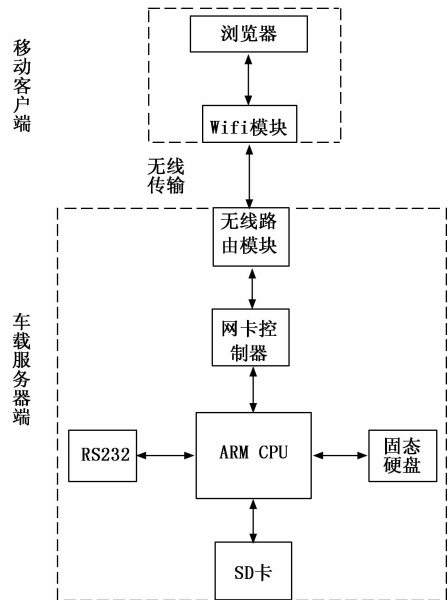


图 2 系统设计框图

车载服务器的硬件平台采用模块化的设计方法,由核心板和外围模块接口构成,核心板选用基于 s3c6410 处理器的 ARM 最小系统,外围模块接口主要包括 RS232 接口,固态硬盘接口, JTAG 接口和无线路由模块接口<sup>[6]</sup>。s3c6410 是一款高性能的 32 位处理器,工作主频高达 667 MHz,内部集成 USB、UART、SDRAM 和 FLASH 控制器,支持丰富的外部设备<sup>[7]</sup>,且在交互式操作较为频繁的状态下具有较快的响应速度。无线路由模块使客户端和服务器处于同一个无线局域网下,在服务器程序启动之后,由无线路由模块提供 WiFi 信号,客户端连接 WiFi 并通过浏览器向服务器发出点播请求。无线路由模块选用 USR-WiFi232-D2 模组,其发射的 WiFi 信号传输速率高达 150 Mbps,为近距离传输提供了较高的数据吞吐量,减少了信道的占用时间,同时可容纳 32 个 WiFi 客户端接入,空旷无遮挡物的环境下, WiFi 信号可覆盖 100 m 直径的圆形区域<sup>[8]</sup>,符合本系统的要求。

本设计运行在嵌入式硬件平台上的 Linux+Apache+Mysql+Php (LAMP) 环境中,客户端与服务器之间采用 B/S 模式<sup>[9]</sup>。系统的工作原理如下:移动客户端在良好的无线网络环境下与车载服务器端建立连接,采用 HTTP 协议通过无线 WiFi 进行数据交互,移动终端向服务器发送视频点播请求,Apache 服务器响应后,交给 Php 引擎解析文件,处理请求,并将视频点播结果返回给客户端。考虑到服务器端视频资源的实时更新性,将视频资源存放在固态硬盘中的特定文件下,方便用户更新资源。

## 2 软件系统设计

### 2.1 操作系统软件的制作

Linux 系统除了有开源、稳定和有良好的移植性等优点, 还支持所有的 TCP/IP 协议, 具备稳定可靠的网络功能。因此车载服务器的操作系统采用 Linux 作为软件的开发和调试平台。从软件的角度分析, 嵌入式 Linux 系统通常包括引导加载程序、Linux 内核、文件系统和用户应用程序这 4 个层次<sup>[10]</sup>。

Bootloader 是指系统的引导加载程序, 可以完成硬件设备的初始化并建立内存空间的映射图, 本系统 Bootloader 采用的是支持 ARM 体系结构的 u-boot。移植过程的关键点是在下载并解压 u-boot 源码后, 需要将解压后的文件导入到本地目录下, 并重新配置目录中的 Makefile 文件, 使用配置好的交叉编译工具对系统源码进行编译, 生成 u-boot.bin 文件说明编译成功。

Linux 内核位于操作系统的最底层, 负责管理外围硬件设备的驱动, 而且为系统的应用程序提供合适的运行环境。内核移植需要注意的地方是在配置内核选项时, 要根据硬件资源添加所需的驱动程序, 主要包括网卡、USB、大容量存储设备等, 使最后编译的内核能够支持 s3c6410 硬件平台<sup>[11]</sup>。

虽然内核是 Linux 系统的核心但用户与操作系统交互的主要方式是系统文件。Linux 系统开启后, 首先加载根文件系统, 根文件系统包含系统启动时所必须的目录和关键性的文件, 以及挂载其他文件系统所必备的文件。根文件系统的制作, 需要利用编译安装好的 Busybox 工具包, 在 Busybox 生成的 install 文件下添加 bin 目录、root 目录、etc 目录、home 目录等, 并为每个目录导入对应的设备文件, 最后使用 mkcramfs 工具完成对根文件系统的打包。

### 2.2 基于 Linux+Apache+Mysql+Php 的设计方案

面对列车旅客人数较多, 常出现同一时刻服务器用户访问量激增导致的视频流传输不稳定的情况。需设计大流量、高并发车载服务器系统架构, 在软件架构上选用 Linux+Apache+Mysql+Php 为组合的 LAMP 架构, 其具备高性能的操作系统、高稳定的 Web 服务器、高存储量的数据库、高效率的编程语言。

Apache 服务器通过 HTTP 协议响应客户端请求并提供 Web 服务, 拥有快速处理 Php 程序的能力, 不容易出现连接超时的情况, 适合应用于访问量较高的站点。Php 服务程序以 Apache 服务器为依托, 侧重于对客户端请求的处理并高效率的执行动态网页。Mysql 数据库对视频资源进行动态管理, 数据存储量大, 支持视频流的存储, 可与 Apache 服务器建立稳定连接, 同时 Mysql 数据库没有用户数量限制, 多个客户端可同时使用一个数据库, 适用于响应多用户请求的系统设计。基于 LAMP 框架的模块交互流程如图 3 所示。

视频点播系统的设计基于 HTTP 协议, HTTP 协议属于一种请求/响应模式的协议, 是实现客户端与服务端通信

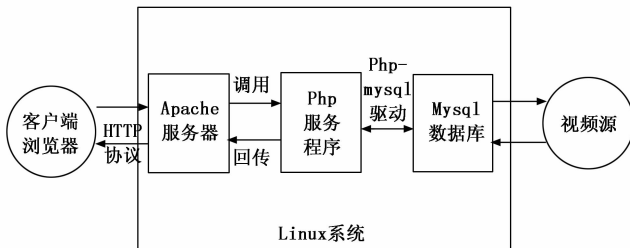


图 3 LAMP 框架模块交互流程图

的常用协议<sup>[12]</sup>。此外, 在多用户访问服务器的情况下, 使用 HTTP 协议有利于减轻服务器的负担, 能够将更多的 CPU 资源提供对外服务。在视频点播系统具备良好通信协议的基础上, 用户可在浏览器中输入服务器 IP 地址, 回车触发页面文件名请求, Apache 服务器接收到响应后, 调用 Php 服务程序并启动 Php 引擎对数据库进行访问<sup>[13]</sup>。Php 引擎将会对 Apache 传送过来的文件从头到尾进行扫描, 根据命令从后台读取, 处理数据, 并动态地生成相应的 HTML 页面, 最终服务器通过 HTTP 协议将页面返回给浏览器。视频点播流程如图 4 所示。

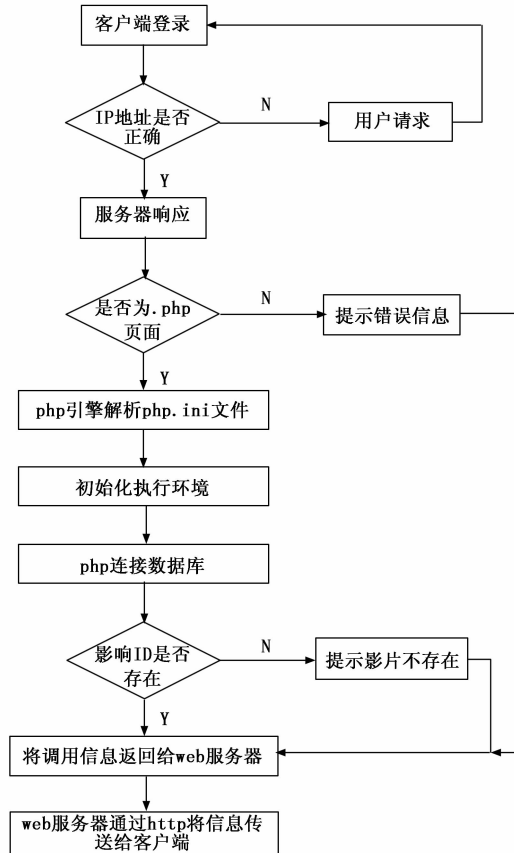


图 4 视频点播流程图

### 2.3 服务端播放器程序的设计

Apache 服务器采用 php 程序调用的方式实现与数据库的动态交互。php 的程序结构主要主要有四部分组成, 分别为服务端应用编程接口、主程序、Php 引擎和代码扩展层。

服务端应用编程接口位于 php 框架的最外层，负责与外围模块进行数据交互；主程序主要完成协议的解析、Web 服务器的通信、文件流的处理等工作；Php 引擎不能单独存在，需要依托 Apache 服务器，它是 Apache 功能模块的一种扩展，主要用于解析 php 程序<sup>[14]</sup>。代码扩展层主要由类库和函数组成，是扩充 php 功能的一种方式。采用 Php 语言编写的服务器端网站主页主要包括视频，图片，书籍，上传几个选项。部分代码如下：

```

</div>
<div class="navbar-collapse collapse">
<ul class="navbar-nav pull-rightmainNav">
<li class="active"><a href="index.html">首页</a>
</li>
<li><a href="Videoindex.Php">视频 &#x2013;</a></li>
<li><a href="images.Php">图片</a></li>
<li><a href="books.Php">书籍</a></li>
<li><a href="load.html">上传</a></li>
</ul>

```

视频点播系统采用 Video 标签在页面中嵌入视频播放器，通过 Video.js 播放器库所提供的 HTML/Flash 自适应播放应用<sup>[15]</sup>，客户端可以自动侦测当前浏览器类型。根据是否支持 HTML5 标签和 Mp4/H.264 编码格式，决定使用 Video 标签嵌入视频播放器还是调用 flash 程序来实现视频播放。客户端播放模式设计如图 5 所示。

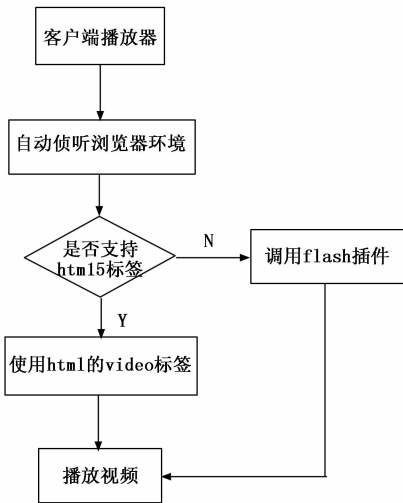


图 5 客户端播放模式设计图

下面是引入 Video 播放的部分核心代码：

1) 在页面中 head 区域引用必要的 js 文件

```

<link href="assets/css/Video-js.min.css" rel="stylesheet">
<script src="assets/js/Video.min.js">
</script>

```

2) 设置 flash 路径，Video.js 会在不支持 HTML5 的浏览器中使用 Flash 播放视频文件：

```

<script>

```

```

Videojs.option.flash.swf="Video-js.swf"

```

```

</script>

```

3) 设置视频内容文件信息，包括路径信息，播放器支持的视频格式，以及播放器的窗口大小。其中 WebM 文件格式主要支持 Vorbis 音频编码和 VP8 视频编码；Ogg 文件格式支持 Vorbis 音频编码和 Theora 视频编码。

```

<Video id="example_Video_1"
class="Video-js vjs-default-skin"
controlspreload="none"width="640"height="320"
poster="http://Video-js.zencoder.com/oceans-clip.png"
data-setup="{}">
$filename=$_GET[filename];
$string=substr($filename, strrpos($filename, ".")+1);
echo"<source
src='Videos/'. $filename. ".type='Video/'. $string. "
<source src='http://视频地址格式.Webm
.type='Video/webm' />
<source src='http://视频地址格式.ogv'
type='Video/ogv' />
<trackkind="captions" src="demo.captions.vtt" srclang="en"label="English"></track><!--Tracks need an ending tag thanks to IE9 -->
<trackkind="subtitles"src="demo.captions.vtt" srclang="en" label="English"></track>
<!-- Tracks need an ending tag thanks to IE9 -->
</Video>

```

### 2.4 数据库程序的设计

为了更好的对数据进行处理操作，使用了 Mysql 数据库管理系统，在数据库中专门为系统中的几类对象建立了数据表，包括影片信息表、用户信息表、资源分类表、点播统计表等。影片信息表主要包括如下属性：存储影片 ID、影片文件名、影片所属分类、影片描述、上传时间、上传影片名称、上传影片大小、文件存储地址等<sup>[16]</sup>，其中影片 ID 是影片在数据库中的唯一标记，通过该属性可以检索到对应的视频。影片信息表中的关键字段如表 1 所示。

表 1 影片信息表

字段名称	数据类型	描述
ID	Int(12)	影片 ID
filename	Char(256)	影片文件名
class	Char(25)	影片所属分类
up_time	Char(25)	上传时间
up_file_name	char(256)	上传影片名称
up_file_size	char(25)	上传影片大小
file_addr	char(256)	文件存储地址

系统中的每个页面都动态生成, 页面显示的内容从数据库实时提取会导致操作量激增, 为了减少连接服务器、选择数据库等重复性操作, 本文定义了一个数据库访问类 DB\_Mysql, 将重复执行的程序隐藏在类的内部。该类的定义如下:

```
class DB_Mysql
{
var $Host = "localhost"; //服务器地址
var $Database = "media"; //数据库名称
var $User = "root"; //用户名
var $Password = ""; //用户密码
var $Link_ID = 0; //数据库连接
var $Query_ID = 0; //查询结果
var $Row_Result = array(); //结果组成的数组
var $Field_Result = array(); //结果集字段名组成的数组
var $Affected_Rows; //影响的行数
var $Rows; //结果集中记录行数
var $Fields; //结果集中字段个数
var $Row_Position = 0; //记录指针位置索引
function construct(); //建立服务器连接
function destruct(); //取消服务器连接
Function connect($ Database = "", $ Host = "", $ User = "", $ Password = ""); //数据库连接
function query($ Query_String); //字符串查询
function seek($ pos); //数据存储位置搜索
function get_rows_array(); //行数结果的集中记录
function get_fields_array(); //字段个数结果的集中记录
function get_affected_rows(); //函数返回前一次 Mysql 操作所影响的记录行数。
function get_rows(); //行数获取
function get_fields(); //字段个数获取
function fetch_one_array($ sql); //数据以数字索引方式存储在数组中
}
```

DB\_Mysql 类是基于 Mysql 扩展函数库封装得到的<sup>[17]</sup>, 该类中, 属性 \$Host、\$Database、\$User、\$Password 定义了连接 Mysql 使用的默认参数; \$Row\_Result 为查询返回的结果集组成的数组; \$Field\_Result 为查询结果集中字段名称组成的数组; Rows 为查询结果集的记录行数。DB\_Mysql 类的使用, 有利于数据库与页面交互程序的快速执行。

## 2.5 LAMP 软件的移植

嵌入式软件开发采用交叉编译, 源文件在宿主机上编译, 运行在目标机上。LAMP 软件的移植, 是指在嵌入式平台上运行交叉编译后的二进制文件的过程<sup>[18]</sup>。这里重点说明 Apache 和 Php 的移植过程。

### 1) Apache 的移植。

Apache 的交叉编译, 需要 Apache 库文件的支持, 通过 gcc 编译器生成 pc 端的 Apache 可执行文件, 使用 arm-linux-gcc 交叉工具链对 Apache 源文件进行编译。移植流程是在官网下载 Apache 源码, 进入解压缩的 Apache 源文件根目录进行配置:

```
[heyun@heyun Apache-arm11]. /configure CC=arm-linux-gcc--host = arm-linux--prefix =/usr/local/Apache-arm--enable-so--enable-modules = so--with-mpm = prefork ac_cv_file__dev_zero=yes
ac_cv_func_setpgrp_void=yes apr_cv_process_shared_works=yes
apr_cv_mutex_robust_shared=yes apr_cv_tcp_nodelay_with_cork=yes ap_void_ptr_lt_long=no
```

配置选项中, 指明了交叉编译工具链 arm-linux-gcc, 生成二进制可执行代码的安装目录/usr/local/Apache-arm, 编译成动态模块。配置完成后, 会自动将当前目录下 configure 文件中的 CC、LD、AR、NM、RANLIB 以及 STRIP 选项修改为:

```
CC=arm-linux-gcc
AR=arm-linux-ar
LD=arm-linux-ld
NM=arm-linux-nm
STRIP=arm-linux-strip
RANLIB=arm-linux-ranlib。
```

配置完成后, 执行 make 进行编译。

### 2) Php 的移植。

Php 的移植需要 Php、libconv、libxml2、zlib 等库文件的支持<sup>[19]</sup>。对于 Php 编译, 在官网下载 Php-5.2.0 源码后, 进入解压缩的 Php 源文件根目录进行配置:

```
[heyun@heyun Php-5.2.0]:. /configureCC = arm-linux-gcc--host = arm-linux--prefix =/usr/local/Apache-arm11/Php-5.2.0/sapi/Apache/--with-apxs =/usr/local/Apache-arm11/Apache/bin/apxs--enable-pdo = shared--with-zlib --with-iconv
```

配置选项与 Apache 类似, 不同之处在于 Php 的编译指明了依赖库文件 zlib, libconv。配置完成后, 执行 make 进行编译即可。

### 3) 配置 Apache 支持 Php。

由于 Apache 默认支持静态网页, 不支持 Php 页面, 所以 Apache 与 Php 各自独立编译完成后, 需配置二者的协同工作<sup>[20]</sup>。进入编译好的 Apache 文件中, 找到 configure 文件, 主要对该文件的下述选项进行配置:

```
# 登陆服务器的端口号与用户名的设置
Port 8080
User nobody
Group nobody
# 网站根目录的设置
```

```
DocumentRoot "/sdcard/ShipinDianBo/"
# 配置 Apache 支持 Php 解析
DirectoryIndex index.html
DirectoryIndex index.Php
DirectoryIndex index.Php3
DirectoryIndex index.phtml
# 指定 Php 解析器的路径及类型
LoadModule.Php5_module libexec/libPhp5.so
ScriptAlias/cgi-bin/" /usr/local/Apache-arm11/cgi-bin/"
ScriptAlias/Php5/" /usr/local/Apache-arm11/Php/bin/"
AddType application/x-tar.tgz
AddType application/x-httpd-Php.Php3
AddType application/x-httpd-Php.Php
AddType application/x-httpd-Php.phtml
```

Apache 与 Php 的移植和配置工作全部完成后，启动 Apache 服务器，自动加载 Php 解析器，可实现客户端与服务器端页面的动态交互。

### 3 系统测试

#### 3.1 系统测试平台

系统的嵌入式平台搭建了 s3c6410 为处理器的 ARM 最小系统，USR-WiFi232-D2 模组，固态硬盘，RS232 串口模块，电源模块。嵌入式系统采用 linux 内核，使用 yaffs 文件管理系统<sup>[21]</sup>，服务器的运行环境选用 LAMP 架构，客户端应用程序基于移动终端而开发。

程序开发阶段，在 pc 端使用 SecureCRT 软件对系统进行调试，通过串口控制嵌入式平台，将之前交叉编译好的 LAMP 二进制文件拷贝到板载固态硬盘中，在 SecureCRT 软件下通过命令，将二进制文件拷贝到 linux 系统对应的文件目录下。USR-WiFi232-D2 模组需配置为 AP 模式，设置 IP 地址为：10.100.100.100。最后配置嵌入式系统初始化文件，设置上电自动开启 Apache 服务器程序。

#### 3.2 测试结果与分析

嵌入式平台与 pc 连接上后，使用 ifconfig 命令查看服务端的 IP 地址，使用 ps 命令查看 Apache 是否正常运行，测试结果如下：

```
[root@CHD /] # ifconfig
[root@CHD /] # eth0 Link encap:Ethernet HWaddr
08:90:00:A0:90:90inetaddr:10.100.100.100Bcast:10.255.
255.255Mask:255.0.0.0UPBROADCAST RUNNING
MULTICAST MTU:1500 Metric:1
```

```
[root@CHD /] # ./Apachectl start
[Fri Sep 29 05:14:37 2000] [alert] httpd: Could not
determine the server's fully qualified domain name, using
127.0.0.1 for ServerName
```

从上面结果测试结果可以看出，配置 Apache 服务器时，将 IP 地址默认设置为 10.100.100.100。使用 Apachectl 命令启动 Apache 服务器，从上面打印的信息可以看

出，服务器正常启动。如图 6 所示，Apache 启动后默认同时开启多个进程，多个用户同时访问 Apache 服务器，可缩短服务器的响应时间，给用户稳定可靠的数据。

```
1030 root      0 SW   [flush-31:2]
1049 root      7636 S   /usr/local/apache-arm/apache/bin/httpd
1051 nobody    7636 S   /usr/local/apache-arm/apache/bin/httpd
1052 nobody    7636 S   /usr/local/apache-arm/apache/bin/httpd
1053 nobody    7636 S   /usr/local/apache-arm/apache/bin/httpd
1054 nobody    7636 S   /usr/local/apache-arm/apache/bin/httpd
1055 nobody    7636 S   /usr/local/apache-arm/apache/bin/httpd
1056 root      3316 R   ps
```

图 6 Apache 开启多进程

调试完成后，重新启动嵌入式平台，移动终端确认连接上 WiFi 信号后打开浏览器，输入 IP 地址 10.100.100.100，回车即可访问服务器中的视频资源。手机访问服务器主界面如图 7 所示，视频播放如图 8 所示。



图 7 手机访问服务器主界面



图 8 视频播放功能

为了测试视频传输过程中的带宽稳定性和用户观看视频的流畅度，将视频的传输速率作为系统性能的核心指标<sup>[22]</sup>。测试方法是在车载服务器和手机终端上分别安装网络性能测试工具 iperf 来对客户端和服务端之间的网络传输速率进行检测，服务器系统安装 iperf 后在终端输入命令 iperf -s 开启服务端，手机终端则需要打开 iperf 输入命令 iperf -c 10.10.100.100-t 60，测试 60 s 内的网络传输速率。如图 9 所示为 60 s 内网络传输速率与时间的关系图，图中的虚线为网络传输速率平均水平线。对图中数据结果的分析可知，车载服务系统所在网络的码流传输速率基本稳定

在 35 Mbps 到 56 Mbps 区间内, 均值为 48.41 Mbps, 方差为 17.61, 离散程度较小, 传输速率在一段时间内保持相对稳定的状态。由于流畅播放高清视频所需要的带宽至少为 4 Mbps, 系统的实时带宽远高于播放高清视频所需的带宽且系统网络较稳定, 故网络性能测试结果满足系统设计的要求。

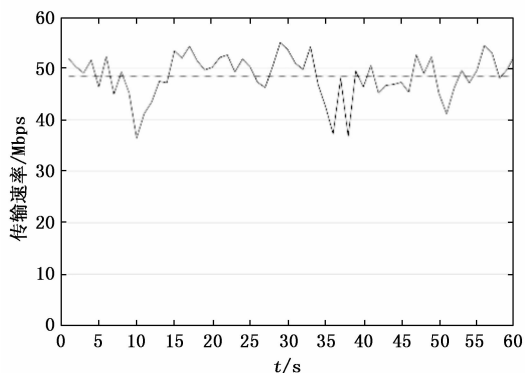


图 9 传输速率与时间的关系图

根据列车车厢所限制的空间范围, 测试其范围内不同位置的网络传输速率及其带宽稳定性, 将移动设备在距离服务器的不同位置作为网络信号测试点, 统计测试点距离车载服务器 1 m 至 15 m 之间网络传输速率随距离的变化情况, 统计结果如图 10 所示。实验结果表明, 随着客户端与服务器之间传输距离的增大, 码流传输速率逐渐减小, 变化范围在 18 Mbps 到 50 Mbps 之间。可以看到, 在列车车厢所允许的最远距离下, 传输速率最小为 18.2 Mbps, 满足系统对于传输高清视频码流速率的要求, 能够让用户在有限空间内获得良好的观影体验。

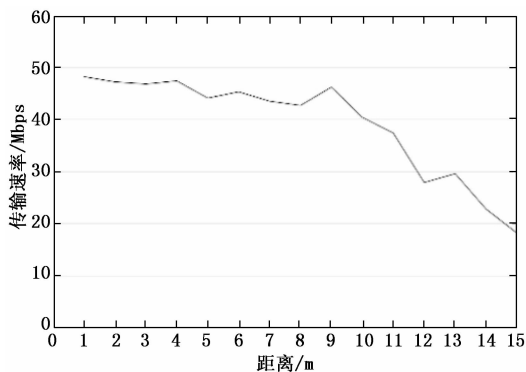


图 10 距离与视频传输速率的关系图

## 4 结束语

本文利用嵌入式硬件的特点, 结合 LAMP 架构的优点, 提出了一种高速移动场景下的平稳宽带车载视频点播系统的设计方案, 设计出了硬件开发平台和应用软件, 客户端与服务器之间是基于 HTTP 协议的 B/S 模式, 服务器端采用 LAMP 框架部署网站。最后分别完成了服务器端和客户端的工作, 实现了视频点播功能, 经测试硬件平台稳定可靠且网络性能良好, 服务器稳定流畅地给用户提供了视频点

播服务, 满足设计需求。在高速移动场景下, 面对大量的用户群如何均衡带宽负载, 从而提升车载服务器的服务能力和网络性能, 是后期研究的一个主要方向。

## 参考文献:

- [1] 李晴飞, 朱 斌, 周琳凯. 一种基于竞争注册和统筹入网的卫星通信系统管控设计 [J]. 数字通信世界, 2019 (9): 19-40.
- [2] 周贤军, 杨恢东. 基于嵌入式 Linux 的网络视频点播系统设计 [J]. 电视技术, 2011, 35 (16): 64-67.
- [3] 陈 晨, 殷海兵. 基于嵌入式流媒体服务器的视频点播系统 [J]. 电视技术, 2016, 40 (9): 41-45.
- [4] 杨 坤. 面向智慧矿山管控平台的视频点播技术 [J]. 计算机系统应用, 2022, 31 (3): 318-325.
- [5] 刘海波, 于鸿洋, 张 萍. 基于嵌入式 Linux 的网络机顶盒设计与实现 [J]. 电视技术, 2010, 34 (S1): 108-109.
- [6] 张怀凯. 基于 LAMP 的嵌入式视频点播系统设计与实现 [D]. 西安: 长安大学, 2017.
- [7] 闫 巧, 李保广. 嵌入式 3G 路由器实用功能的设计与实现 [J]. 计算机工程与设计, 2014, 35 (5): 1634-1638.
- [8] 杨李博. 基于 Wi-Fi 的无线传感器网络节点设计与实现 [D]. 上海: 上海交通大学, 2014.
- [9] 梁少芳, 刘 治, 周 英, 陈升国. 基于 S3C2410 的 VoD 系统构建 [J]. 电视技术, 2010, 34 (S1): 76-78.
- [10] 韦东山. 嵌入式 Linux 应用开发完全手册 [M]. 北京: 人民邮电出版社, 2008.
- [11] 江俊杰, 王志明. 基于 X264 的嵌入式视频监控系统设计 [J]. 计算机工程与设计, 2013, 34 (12): 4199-4203.
- [12] 李 娜. 多媒体可信宽带网可靠性测试与分析系统的设计与实现 [D]. 成都: 电子科技大学, 2014.
- [13] 王庆志. 基于 Web 服务器的结构研究与设计 [D]. 北京: 中国石油大学, 2010.
- [14] 郑增乾, 王 锴, 赵 涛, 等. 带宽和时延受限的流媒体服务器集群负载均衡机制 [J]. 计算机科学, 2021, 48 (6): 261-267.
- [15] 曾 敏. 基于 Vtools 的虚拟仿真系统的优化 [J]. 教育教学论坛, 2013 (20): 184-185.
- [16] 王 淇. 嵌入式视频点播系统的设计与实现 [D]. 武汉: 华中科技大学, 2011.
- [17] 李庆城, 彭 洁, 宫晓利, 等. 嵌入式 HTML 文档解析器的设计与实现 [J]. 计算机工程, 2009, 35 (9): 258-260.
- [18] RICHARD STEVENS W, STEPHEN A RAGO. Unix 环境高级编程 (3 版) [M]. 戚正伟, 张亚英, 尤晋元, 译. 北京: 人民邮电出版社, 2014.
- [19] 赵国安, 郁 斌, 薛林强. 基于 Linux 嵌入式原理与应用开发 [M]. 北京: 清华大学出版社, 2011.
- [20] 袁小平, 鲍 捷, 张明涛, 等. 基于嵌入式 Web Server 的煤矿视频监控系统研究 [J]. 电视技术, 2014, 38 (9): 232-234.
- [21] 刘立军. 嵌入式网络数字带宽测量监控系统设计与实现 [J]. 计算机测量与控制, 2014, 22 (12): 4157-4159.
- [22] 侯文放, 王 辉, 曾 波. 数据中心网络中基于传输速率分配的 TCP 协议 [J]. 计算机测量与控制, 2020, 28 (3): 227-230.