

浅析 LINUX 环境下实时流媒体传输在远程教育平台中的应用

徐丹丽

上海电视大学信息与工程系

摘要: 远程教学中需要传输丰富多彩的多媒体资源, 包括一些实时数据, 网络拥塞、时间等待等问题使得教学效果并不理想, 本文提出了用流媒体方式传送教学资源的方法, 同时具体讲述了在 linux 环境下, 流媒体的传送与接收在远程教育平台网络中的实现。

关键字: 远程教育平台 流媒体 RTP/RTCP 协议

1. 引言

网络技术的日趋成熟带来了远程教育的飞速发展。远程教育平台的开发是以网络为依托, 开发一套完整的教学支撑软件和管理服务软件, 保证整个网络教育的有序运行和管理。远程教育平台包括网络环境、网站(网上课堂)、学生、课件、资源库等要素, 其中网络环境为各种教与学信息的流动提供了有效的通行渠道。

远程教育相对于传统教学来说, 前者教与学的双方通常处在不同的空间。在具体实现平台上需要供教学和教务管理服务, 例如信息发布和浏览、课程浏览和学习管理、多媒体授课学习、远程实验、远程考试等, 要完成所有这些服务, 要求我们将所涉及到的各种信息在远程教育网络上传送, 即服务的实质就是信息的流动。

随着生活水平的提高和网络技术的发展, 人们传输的信息已不仅仅局限于文字的方式, 对网络的使用, 也从简单的信息查询转向视频、音频、动画等多媒体方面。学员们希望通过宽带网络获得交互式的学习体验。

虽然目前使用了各种压缩技术, 普通的视频/音频文件依然包含大量的数据, 人们需要具备足够的本地硬盘空间, 等待较长的时间下载, 之后才能观看。但是事实上, 很多信息学员们在了解后即可删除, 无需保存, 并且希望能够“即点即看”, 不愿等待冗长的下载时间。为解决这个问题, 一方面可以改进视频/音频压缩技术, 一方面可以将教学资源转成“流”式(Streaming)文件, 进行传送和播放。

本文要讨论的内容是流媒体在 linux 环境下, 流媒体的传送与接收在远程教育平台网络中的实现。

2. 流媒体

远程教育平台上, 多媒体资料的网络传输主要有下载(Download)和流式传输(Streaming)两种方式。前者要求先将数据完整地下载至本地计算机后, 才能播放。如同一个人想洗一下手, 规定要先接满一桶水。显然, 这种方式等待时间长, 要求客户端有较大

的硬盘空间。事实上，我们可以拧开水龙头来洗手，随拧随洗，洗完即关。后者就满足了这样的要求。

应用流式传输时，流媒体服务器向用户计算机连续、实时传送数据流，用户不必等到整个文件全部下载完毕，而只需经过几秒或十数秒的启动延时即可播放多媒体文件，同时，文件的剩余部分将在后台从服务器内继续下载。在 Internet 中使用流式传输技术的连续时基媒体（Time-stamp Media）就称为流媒体，通常也将其视频与音频称为视频流和音频流。显然，流式传输克服了下载方式的缺点。

流式传输有顺序流式传输(Progressive Streaming)和实时流式传输(Real-time Streaming)两种方式。其中实时流式传输是实时传送，例如远程互动课堂、远程讨论会等，“实时”是指在一个应用中数据的交付须与数据的产生保持精确的时间关系。

3. 流媒体协议——RTP/RTCP 协议

实时流传输需要遵循相关的协议。目前广为接受的协议是实时传输协议 RTP（Real-time Transport Protocol），它专门针对 Internet 上多媒体数据流的实时传输，由 Internet 工程任务组 (IETF) 作为 RFC3550 发布。RTP 被定义为在一对一（unicast，单播）或一对多（multicast，多播）的传输情况下工作，其目的是提供时间信息和实现流同步。它与其它各种网络协议的关系参见图 1。

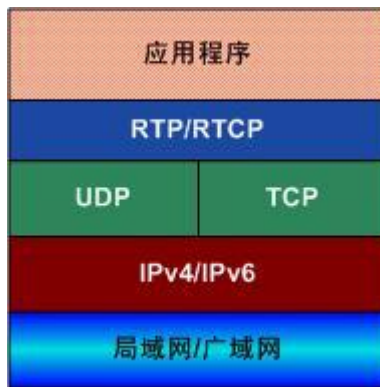


图 1 RTP 与各种网络协议的关系

RTP 本身只保证实时数据的传输，并不能为按顺序传送数据包提供可靠的传送机制，也不提供流量控制或拥塞控制，这些服务需要通过实时传输控制协议来提供。

实时传输控制协议 RTCP（Real-time Transport Control Protocol）负责管理传输质量，在当前应用进程之间交换控制信息。在 RTP 会话期间，各参与者周期性地传送 RTCP 包，包中含有已发送的数据包的数量、丢失的数据包的数量等统计资料，因此，服务器可以利用这些信息动态地改变传输速率，甚至改变有效载荷类型。RTP 和 RTCP 配合使用，才能有效传送网上的实时数据。

现在大部分的网络视频点播，远程教育都是采用 RTP/RTCP 协议来实现的。

4. Linux 环境下实时流媒体传送的实现

我们在采用 B/S 构架、运行 Linux 环境的远程教育平台上，分析如何实现实时流媒体传送，这里选用当前最流行的 RTP 协议。Linux 是一个开放的平台，目前已有很多开放源代码的 RTP 库，如 LIBRTP、JRTPLIB，这些库按照 RFC 3550 约定为具体应用搭建基本框架，充分利用这些资源，可以大大简化我们的开发过程。JRTPLIB 是一个用 C++写的面向对象的 RTP 库，严格遵循 RFC 3550 约定，可以运行在 Windows、GNU/Linux、Solaris 等多种操作系统上，对于熟悉 c++的人来说的确是一个非常不错的选择。下面就以 JRTPLIB 为例进行说明。

(一) 环境搭建

首先从相关网站上下载 JRTPLIB 库文件新版本，它们被要所成不同的格式，如 `jrtp-lib-3.1.0.tar.bz2`、`jrtp-lib-3.1.0.tar.gz`，选择需要的格式下载。本例在 Linux 环境中开发，使用 `jrtp-lib-3.1.0.tar.bz2` 即可。下载后解压缩，进行配置并编译，通过后，开始安装。

(二) 初始化处理

同其他的应用程序相类似，程序开始部分要完成一些初始化工作，例如头文件的添加、类的添加等等。首先我们要生成 RTPSession 类的一个对象 `sess` 来表示此次 RTP 会话，然后调用 RTPSession 类的 `Create()`函数来对其进行初始化操作，其参数是此次 RTP 会话所采用的套接字端口号。

```
RTPSession sess;  
  
status = sess.Create(5002);
```

设置恰当的时基 (time-stamp) 单元，可以通过调用 RTPSession 类的 `SetTimestampUnit()`函数来实现的。例如，当使用 RTP 会话传输 6600Hz 采样的音频数据时，实现如下：

```
sess.SetTimestampUnit(1/6600);
```

接下来还要添加目的地 IP 地址，该地址为 `RTPIPV4Address` 类型，使用 RTPSession 类 `AddDestination()`函数实现。

```
status = sess.AddDestination(addr);
```

同样实现删除、清空目的地址可调用 `DeleteDestination()`和 `ClearDestinations()`等函数。

(三) 数据发送

初始化后，接下来可以调用 RTPSession 类的数据发送函数 `SendPacket()`，向所有目的地发送流媒体数据。这是一个重载函数，具有多种调用形式（参看 RTPSession 类对该函数的定义），例如在下面的调用形式：

```
int SendPacket(void *data,int len,unsigned char pt,bool mark,unsigned long timestampinc);
```

其中第一个参数是待发送的数据缓冲区地址，第二个参数是待发送数据的长度，后面依次是 RTP 负载类型、标识和时基增量。

对于同一个 RTP 会话双方，负载类型、标识和时戳增量相同，可以对它们提前设置，调用 RTPSession 类的 SetDefaultPayloadType()、SetDefaultMark()和 SetDefaultTimeStampIncrement()函数就可实现。这时可以调用下面相对简化的函数形式发送数据：

```
int SendPacket(void *data,int len);
```

这里只需要指出数据地址和长度就可以了，例如：

```
sess.SendPacket(buff,10);
```

（四）数据接收

接收实时流媒体数据，通过调用 RTPSession 类的 PollData()函数来接收收到的 RTP 或 RTCP 数据报。同一个 RTP 会话中允许有多个参与者(源)，通过调用 GotoFirstSourceWithData()和 GotoNextSourceWithData()函数来实现源的遍历，通过 GetNextPacket()函数提取 RTP 数据报。

```
if (sess.GotoFirstSourceWithData()){  
  
    do{  
  
        RTPPacket *pack;  
  
        while ((pack = sess.GetNextPacket()) != NULL)  
  
            {printf("Got packet !\n"); delete pack; }  
  
    } while (sess.GotoNextSourceWithData());  
  
}
```

这里定义了三种接收模式，每种接收模式都具体规定了哪些收到的包会被接受，哪些包会被丢弃。模式的设定调用 RTPSession 类的 SetReceiveMode()函数就可以实现。远程教育平台通常只对通过权限验证的用户发送数据。管理这些用户可以通过调用 AddToAcceptList ()、DeleteFromAcceptList 和 ClearAcceptList ()函数来实现。

5. 小结

以宽带为基础，流媒体不仅可以进行单向的视频点播，还能够提供真正互动的课堂，如互动课堂、远程试验。目前，流媒体在中国的宽带建设中被列为最主要的应用之一，流媒体

技术将成为未来互联网应用的主流，远程教学中需要传输丰富多彩的多媒体资源，远程教育平台中将资源封装成流媒体的格式，采用流式传输，将会带来更好的教学效果。

参考资料:

1. RTP 协议相关资料 www.ietf.org/rfc/rfc3550.txt
2. JRTPLIB 库相关下载及资料 <http://research.edm.luc.ac.be/jori/jrtplib/jrtplib.html>
3. 《TCP/IP 网络互连 第 3 卷 客户/服务器编程及应用 Linux/POSIX Sockets 版（影印版）》人民邮电出版社
4. 《Windows 网络编程技术》机械工业出版社 Anthony Jones 等著