实时传输协议基础

一、实验目的

1、了解实时传输协议 RTP 和实时传输控制协议 RTCP 的基本内容；
2、学习使用 RTP 数据报的信息计算 QoS 参数；
3、学习实时数据传输控制的基本原理；
4、了解媒体内同步和媒体间同步的基本概念，学习时间戳同步技术的应用；

二、实验原理

1、实验预备知识

本讲义中对于所涉及到的 RTP 协议和 RTCP 协议的原理不做详细说明，仅对实验中涉及到的数据报格式部分进行说明。请在实验前进行相关资料的查询（可参阅本讲义“参考资料”部分的文献）。本实验中的实验内容以及示例程序并没有真正实现 RTP 协议和 RTCP 协议，仅仅对其中最基本的部分进行了模拟。

由于实时媒体流（包括视频和音频流）需要进行压缩在进行传输，而媒体压缩的有关知识超出本实验讲义范围，故实验示例程序中将以字符串流来替代实时媒体流。通过对单个或多个字符串流的网络传输来学习同步等基本概念。

本实验需要对TCP/UDP编程有相当程度的了解，UDP编程相关的示例程序在本科生《计算机网络实验》中有相应的部分，必要时可访问实验室网址下载有关程序。实验样本程序采用 VisualC++实现，仅仅提供参考，上机内容中的所有内容并没有要求实现的方式，采用任何编程环境、Linux 用脚本程序等均可以。

2、媒体同步技术

在多媒体系统中，通常利用多种媒体从不同侧面表现同一个主题，此时不同媒体之间就存在时间同步的问题，例如，视频播放时伴音应与口形相吻合，演播幻灯片时解说词与正在显示的内容相对应等。同步是多媒体系统中的一个关键性问题，它与系统中的许多因素有关，如：通信系统、操作系统、数据库、文件以及应用形式等，因此，多媒体系统中同步应在不同的层面上考虑。在实验中涉及的同步技术是时间戳同步技术。

时间戳同步技术方案中，不同的媒体流通过分离的信道传输，媒体间的同步是通过所有媒体流的数据单元都达到一样的端端延迟这一间接的方法而完成的。即给源点中所有相关媒体数据流都强加一个固定的端端延迟，然后在目的结点通过缓存使“流延迟”相等。

发送方给每个数据分组加上时间戳，记下生成该分组的时间，接收方为每个数据流估算延迟，然后计算同步延迟，该延迟通常是所有数据流的最大延迟，接收到的数据缓存到释放时间，才将数据分组提交给用户。以下定义了主要的衡量指标：

释放时间＝时间戳＋同步延迟－提交延迟；

如果您在阅读过程中发现任何疏漏或错误，请尽快联系：network@ustc.edu.cn　cxh@ustc.edu.cn

同步延迟＝Tc＋Tt＋Te＋Td;

流延迟＝Tc＋Tt＋Td。

这种方法通过周期性地更改流延迟来适应网络延迟的变化，接收方利用收到的数据分组里的时间戳来测算实际的延迟。

时间戳一般是采用绝对时间计算的，需要全网域的时钟同步（例如用 Windows2000 的 MediaServer 提供视频服务的时候，就需要将客户端和服务器端的时间进行同步）。还有一种相对时间戳技术，它把多媒体信息中的所有媒体单元放在一个相对时标上，媒体单元在相对时标上的位置决定它的相对时间戳，具有相同相对时间戳的媒体单元同时表现，以此达到媒体同步。

时间戳同步技术最大的特点是接收方基于时间戳实现媒体的同步，同步信息（即时间戳）装入数据分组一起传输，不附加信道，无须另外的同步信息，不改变数据流。但在分组中读写时间戳和计算延迟增加了系统的处理开销。

3、RTP 数据报头标格式

RTP 提供端对端网络传输功能，适合通过组播和点播传送实时数据，如视频、音频和仿真数据。RTP 没有涉及资源预订和质量保证等实时服务。RTP 报文格式中包括固定的 RTP 报文头，可选用的作用标识（CSRC）和负载数据。如果 RTP 所依赖的底层协议对 RTP 报文的格式有所要求，RTP 报文的格式必须进行修改或重新定义。通常，单一的底层数据报文仅包含单一的 RTP 报文。下图为一个典型的 RTP 报文。

RTP 报文头部分各个参数的意义如下：

1	Version	版本	2	协议版本信息
2	Padding	填充	1	如果设置了填充位，表示在该 RTP 包的末尾含有并非有效负载数据的填充位。
3	eXtension	扩展位	1	为 1 时表示有扩展头标。
4	CSRC Count		4	CSRC 作用标识的个数
5	Marker	标志位	1	标志位由具体的应用框架定义。例如，RTP 的 MPEG4 负载格式中，标志位设为 1 标志这是 VOP 的最后一个（或仅有一个）RTP 包。
6	Payload Type	负载类型	7	对音频或视频等数据类型说明，并说明编码方式。
7	Sequence	序列号	16	为了安全，服务器从一个随机初始化值开始，每发

	Number			送一个 RTP 数据包序列号增加 1。客户端可以根据序列号重新排列数据包的顺序，并对丢失、损坏和重复的数据包进行检测。
8	Timestamp	时间戳	32	RTP 时间戳为同步不同的媒体流提供采样时间，用于重新建立原始音频或视频的时序。另外，它还可以帮助接收方确定数据到达时间的一致性或变化（有时被称为抖动）。
9	SSRC	同步源标识	32	帮助接收方利用发送方生成的唯一的数值来区分多个同时的数据流。SSRC 必须是一个严格的随机数。
10	CSRC	作用标识	32 - 32*16	网络中使用混合器时，混合器会在 RTP 报文头部之后插入新的同步源标识，其作用仍是区分多个同时的数据流。在所有 RTP 报文中，开始 12 个字节的格式完全按照图 1 中定义的格式，而 CSRC 标识列表仅出现在混合器插入时。

标准的 RTP 数据报文头部参数对 RTP 支持的所有应用类的共同需要是完整的。然而，为了维持 ALF 设计原则，报文头部还可以通过改变、增加参数实现优化，或适应特殊应用的需要。由子标志位和负载类型段携带特定设置信息，所以很多应用都需要它们，否则要容纳它们，就要增加另外 32 位字，因此，标志位和负载类型允许分配在固定头中。包含这些段的八进制可通过设置重新定义以适应不同要求，例如采用更多或更少标志位。如果有标志位，既然设置无关监控器能观察报文丢失模式和标志位间关系，我们就可以定位八进制中最重要的位。

如果 RTP 协议需要负载其它特殊格式（如视频编码）的音视频数据，所要求的信息应该携带在报文的数据负载部分。所需信息也可以出现在报文头部，但必须总是在载荷部分开始处，或在数据模式的保留值中指出。如果特殊应用类需要独立负载格式的附加功能，应用运行设置应该在现存固定报文头部的 SSRC 参数之后，定义附加固定段。这些设置能使客户端迅速而直接访问附加段，同时，与监控器和记录器无关设置仍能通过仅解释开始 12 个八进制处理 RTP 报文。

注：以上内容参考《RFC1889》中“RTP Fixed Header Fields”部分。我们在实验中发送端生成 RTP 数据报头标中的前 12 字节（96 比特）的固定信息，并在接收端分析这些信息。

4、RTCP 传输控制协议

RTP 本身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，它依靠 RTCP（Real－time Control Potocol）传输控制协议提供这些服务。RTP 的控制协议 RTCP 通过在会话用户之间周期性地递交控制报文来完成监听服务质量和交换会话用户信息等功能。根据用户间的数据传输反馈信息，可以制定流量控制的策略，而会话用户信息的交互，可以制定会话控制的策略。

RTCP 数据报和 RTP 数据报不同，RTP 数据报仅仅具有一种，可通过扩展头标等方法实现灵活的数据报内部结构；但是 RTCP 数据报有几种类型如：发送端报告（SR）、接收端报告（RR）、源描述项（SDES）、结束标识（BYE）、应用程序说明（APP）等等。其报文头标格式如下：

每种 RTCP 数据报的功能及静荷数据格式说明请参阅 RFC1889，根据协议定义，静荷数据的长度必须是 32 比特的整数倍，此处不进行解释。下表是 RTCP 报文的头标含义解释：

1	Version	版本	2	协议版本信息
2	Padding	填充	1	如果设置了填充位，表示在该 RTP 包的末尾含有并非有效负载数据的填充位。
3	reception report count	报告数	5	数据报中所含有的报告数目。
4	Packet Type	类型	8	数据报的类型(发送端报告或接收端报告等)
5	length	长度	16	数据报的长度

在实验的示例程序中并没有实现 RTCP 协议规范，下面介绍一下静荷数据中一般应该具有的信息。下表中即最基本的信息项。

信息项	bits	信息项含义
SSRC	32	和 RTP 数据报中的同步源标识对应，表示是该数据流的控制报文信息
fraction lost	8	
cumulative number of packets lost	24	累积丢失的 RTP 数据包数目
extended highest sequence number received	32	
interarrival jitter	32	抖动参数
last SR timestamp (LSR)	32	
delay since last SR (DLSR)	32	

5、RTP/UDP/IP 结构

下图所示即为 RTP/UDP/IP 数据包的头标格式：

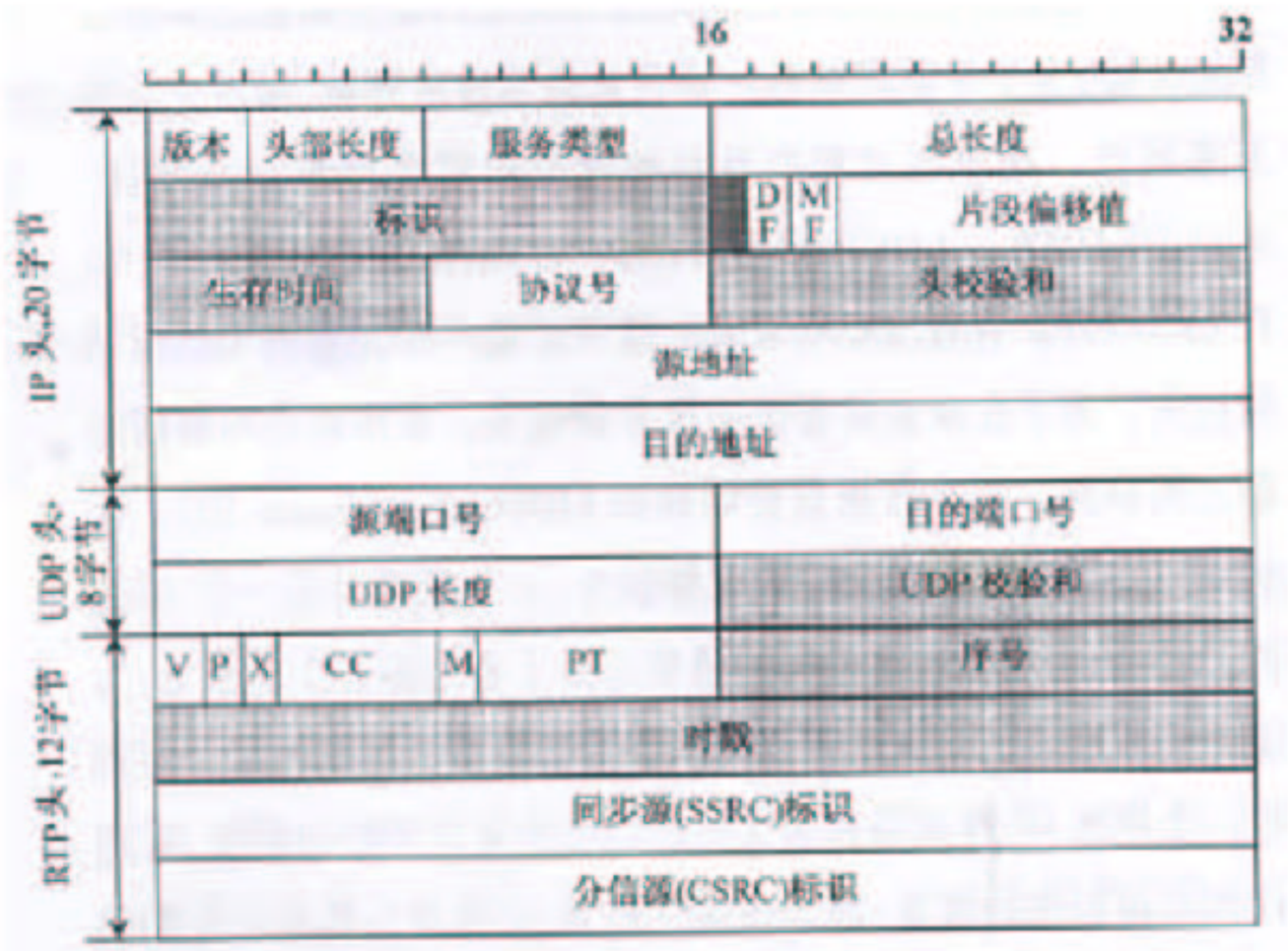

图 1　RTP/UDP/IP 报头的构成

报头的总开销包括：RTP(12 字节)+UDP（8 字节）+IP（20 字节）＝40 字节。显然 40 字节的开销十分可观，在本实验的示例程序中最简单的媒体流是一个单字节形成的数据包，此时报头占整个数据包的 97.6%。为了提高传输效率，头标压缩的概念近年也比较受人关注（在无线通信的情况下尤为热烈），有兴趣的同学可以查阅相关的资料。

6、示例程序介绍

基本情况

示例程序由发送程序和接收程序构成，设计为在单机上运行（这样便于实验）；在双机情况下效果会更好一些（此时网络因素对数据流的影响比较明显）。

示例程序用两个字符串流模拟两个不同的数据流，没有用真正的音频数据和视频数据，故而在程序中也没有实现 RTP 协议中有关数据编码类型的相关内容。

RTP 报文头标

在《RFC1889》中提供了通用数据结构定义的建议如下：

1、RTP 数据头标格式

```
typedef struct {
    unsigned int version:2;    /* protocol version */
    unsigned int p:1;          /* padding flag */
    unsigned int x:1;          /* header extension flag */
    unsigned int cc:4;         /* CSRC count */
    unsigned int m:1;          /* marker bit */
    unsigned int pt:7;         /* payload type */
```

```
        u_int16 seq;                /* sequence number */
        u_int32 ts;                 /* timestamp */
        u_int32 ssrc;               /* synchronization source */
        u_int32 csrc[1];            /* optional CSRC list */
    } rtp_hdr_t;
```

2、RTCP 通用头标结构

```
    typedef struct {
        unsigned int version:2;   /* protocol version */
        unsigned int p:1;         /* padding flag */
        unsigned int count:5;     /* varies by packet type */
        unsigned int pt:8;        /* RTCP packet type */
        u_int16 length;           /* pkt len in words, w/o this word */
    } rtcp_common_t;
```

3、示例程序中完成以下内容：

（1） RTP 数据报头标数据结构的定义、使用，生成一个完整的 RTP 数据包；此部分功能在发送程序中。

（2） 已有 RTP 数据包格式分解（根据给定的 RTP 数据报提取头标信息和数据信息并显示）；此部分功能在接收部分程序中。

时间戳格式

NTP 时间在网络时间协议中定义（Network Time Protocol），NTP 时间记录的是当前时间距离 1900 年 1 月 1 日 0 时的相对时间，（值得注意的是：在 Windows 和 Linux 操作系统中利用时间函数获取的时间信息也是 64 比特的无符号整数，但是该数值表示的是从 1970 年 1 月 1 日 0 时开始的相对时间，故本地计算机的时间表示和 NTP 协议中采用的时间表示需要经过专门的换算）。当前 NTP 协议中定义的时间格式导致 2036 年时 64 比特重新归零，此时需要特殊处理。在 UNIX 下，获取系统时间的函数是 gettimeofday()，返回值＋2208988800u 则可以得到 NTP 时间戳；在 Windows 下可以使用 GetSystemTimeAsFileTime（）获得一个 64 比特的时间信息，再经过转换形成距离 1900 年 1 月 1 日 0 时的的毫秒数。

在 Internet 上有很多 NTP 服务器提供时间信息服务，管理员身份的用户可以为自己的计算机指定一个时间服务器，这样系统就可以自动和时间服务器保持同步的时间。互相通信的计算机之间保持时间的同步时非常重要的，最初的时候人们提出简单时间服务（daytime 服务，多媒体通信实验室的 Linux 服务器就提供这样的服务，可以通过简单地查询相应地计算机端口获取时间信息）；随着实时通信地日益增多，NTP 地提出提供了时间同步相关的强大地保证。保持计算机时间地同步即保证互相通信地计算机之间地时间戳具有可参照性，例如：将接收方收到数据包时地时间戳和数据包中含有地时间戳进行比较可以获得精确地数据包延迟信息。虽然形成 RTP 数据报部需要 NTP 时间戳的信息，但是生成 RTCP 数据报需要 NTP 时间戳的信息，故在此做简要介绍。Internet 的很多协议中都含有时间戳的定义，一般在定义的时候会和 NTP 时间戳对比，在阅读文献的时候请留意。

RTP 报文头标中的时间戳长度是 32 比特，和 NTP 时间戳不同，根据 RTP 协议定义，这是一个随机数加偏移量形成的无符号整数。偏移量是根据当前流媒体得取样频率进行计算的，如果是采样率为 8k 的语音信号，每 20 毫秒形成一个 RTP 数据报，则相邻两个 RTP 数据报的时间戳差值为 160。

一般来说，实时通信地程序必须具备 NTP 协议定义的功能以保证实时通信地管理和监

控顺利进行。在本实验中，为了简化程序并没有实现此功能，程序设计为发送、接收均在本地计算机上运行，发送方和接收方都以计算机本地时间为准，可认为是两个同步的计算机在进行网络通信。

示例程序中完成了以下内容：

（1） 由本地计算机时间生成 64 比特 NTP 时间戳，由于示例程序中没有实现 RTCP 协议故没有用到NTP 时间戳，理解 NTP 时间戳生成需要了解 Windows 下时间控制函数。

（2） 由本地计算机时间生成32 比特 RTP 时间戳。根据 RFC1889 中对 RTP 时间戳的定义，程序中 RTP 时间戳是由一个随机数＋数据报生成时刻的数据偏移量得到；这个偏移量和数据流的发送速率有关系，示例程序中偏移量如下定义：数据报生成时刻距离上一个数据报生成时刻的这一段时间中按照正常稳定的发送速率应该发送的字节数。如在实验过程中对这种理解有疑问，请与实验室联系。

Qos 参数计算

在示例程序中仅仅完成对单个数据流抖动参数的计算。根据 RFC1889 中定义，计算抖动（interarrival jitter）参数需要以下步骤：

假定 Si 是第 i 个接收到的 RTP 数据报的时间戳、Ri 是接收到第 i 个 RTP 数据报时刻由接收端计算的时间戳。

D(i,j)=(Rj-Ri)-(Sj-Si)=(Rj-Sj)-(Ri-Si)

定义数据流抖动（jitter）参数 J：

J=J+(|D(i-1,i)|-J)/16

在示例程序中分别计算了两个独立的数据流的抖动参数，在实际测试过程中，笔者认为接收端程序中检测到的数据抖动由下列原因引起：发送端数据报处理时间导致的抖动、接收端数据报的处理时间导致的抖动、网络导致的抖动。由于示例程序在单机运行，对于网络因素不能很好的观察，但是接收端处理时间非常重要。

三、实验内容

1、同步机制

（1）单个媒体流的同步

在发送程序中添加相应模块模拟乱序发送，即认为制造不同步的情况，在接收程序中做相应的同步处理。顺序发送时发送方按照下面的顺序发送字符串：

AAAAAAAAAABBBBBBBBBBCCCCCCCCCCDDDDDDDDDDEEEEEEEEEEFFFFFFFFFFGGGGGGGGGGHHHHHHHHHHIIIIIIIIIIJJJJJJJJJJ

上机内容：

A、修改发送程序使之不按照上面的程序发送，然后再接收端整理顺序；

B、修改发送程序少发送某个字符串，接收端在缺少的相应位置填充特殊的标识字符串“##########”。

（2）多个媒体流的同步

示例程序中同时发送两个数据流的时候，期望的顺序是：

AAAAAAAAAAaaaaaaaaaBBBBBBBBBBbbbbbbbbbbCCCCCCCCCCccccccccccDDDDDDD
DDDddddddddddEEEEEEEEEEeeeeeeeeeeFFFFFFFFFFfffffffffGGGGGGGGGGgggggggggggH
HHHHHHHHHHhhhhhhhhhhIIIIIIIIIIiiiiiiiiiiJJJJJJJJJJjjjjjjjjjj

在实际程序运行的时候，接收端显示的顺序为：

AAAAAAAAAAaaaaaaaaaabbbbbBBBBBBBBBBbbbbbccccCCCCCCCCCCccccDDDDDDD
DDDddddddddddeeeeeEEEEEEEEEEeeeeefffffFFFFFFFFFFfffffggggGGGGGGGGGGgggggh
hhhhHHHHHHHHHHHhhhhhiiiiiIIIIIIIIIiiiiijjjjjJJJJJJJJJjjjjj

上机内容：

A、修改发送程序，使之按照正确的顺序发送，此时需要处理两个发送进程之间的同步；

B、修改接收程序，使接收到的数据流按照正确的顺序显示。

2、用 RTP 数据报传输单个音频数据流

上机内容：参考《实时语音传输实验》中的有关部分，根据示例程序提供的 RTP 数据报的格式将音频数据流封装为 RTP 数据报进行传输。

3、生成 RTCP 数据报

上机内容：

A、在示例程序中仅仅计算了抖动参数，并没有生成相应的 RTCP 数据报，修改接收端程序：根据抖动参数、NTP 时间戳 等必要信息生成最简单的 RTCP 数据报。

B、在发送程序中添加接收 RTCP 数据报的相应部分，显示自己生成的 RTCP 数据报。

四、思考题

1、单个媒体内的同步和不同媒体流之间的同步在处理方式上有什么不同？讨论这两种同步可能采用的机制。
2、绝对时间戳和相对时间戳在进行同步处理的时候有些什么不同，请阐述个人对这两种方式有缺点的认识。
3、在广域网的环境下，利用 RTP/RTCP 协议实现点到点的实时通信时是否需要沿途所有的路由器和交换机支持，为什么？RSVP 协议呢？
4、有人说："RTP/RTCP 协议是应用层协议、SCTP 协议是传输层协议"。在本实验中没有涉及 SCTP 协议，请阅读有关资料并判断该结论是否正确，说明理由。列举您所知道的基于 RTP/RTCP 协议地应用程序和基于 SCTP 协议的应用程序（和系统），那种协议的应用范围更广泛？
5、在 RTP 协议中，对组播有什么支持？参考本科生《计算机网络实验》中的组播实验内容可以做支持组播的 RTP 程序，这样需要完成那些功能？

五、参考资料

【1】IP 电话中的压缩技术及测试
http://www.computertelephony.com.cn/article/01122901.htm
【2】多媒体通信相关技术
http://info.wri.com.cn/studyreport/report3.htm
【3】RTP: A Transport Protocol for Real-Time Applications
http://www.ietf.org/rfc/rfc1889.txt
【4】Stream Control Transmission Protocol
http://www.ietf.org/rfc/rfc2960.txt
【5】VC 时间控制函数
http://www.hooyang.com/hy_tc/lantools/t_v_c++/vc++_20010708_0027.htm