

# 基于混合式分级流标签的 IPv6 组播通信

陈华胜, 徐 瑞, 李伟华

(西北工业大学计算机学院, 西安 710072)

**摘要:** IPv6 网络协议引入流标签概念, 在路由器转发提供 QoS 支持。针对 IPv6 多媒体组播通信的需要, 提出一种混合式流标签分级定义方案实现 IPv6 组播通信的 QoS 支持, 在 IPv6 实验网上进行多媒体组播通信模拟测试, 结果表明该方案能够提供满意的 QoS 支持。

**关键词:** IPv6 协议; 流标签; 组播通信

## IPv6 Multicast Communication Based on Hybrid Flow Label Classification

CHEN Hua-sheng, XU Rui, LI Wei-hua

(Department of Computer Science, Northwestern Polytechnical University, Xi'an 710072)

**【Abstract】** Flow label is introduced in IPv6 protocol, and used to provide routers for QoS implementation purpose of forwarding packets. This paper focuses on the demand of IPv6 multimedia multicast communication, and proposes an IPv6 multicast communication scheme of flow label classification providing QoS using a hybrid approach. The scheme with multimedia multicast communication is tested on IPv6 test network, and satisfactory QoS results are gotten.

**【Key words】** IPv6; flow label; multicast communication

传统的IPv4网络只是尽最大努力(best-effort)来传输,组播这种UDP传输方式由于没有提供有效的检测控制机制,不能提供足够的QoS支持。IPv6定义了一个20位流标签字段,运用于非默认的QoS连接,如多媒体实时数据连接,在路由器对流数据包进行识别和处理,提供QoS支持。由于关于流标签字段的使用细节仍没有定义,因此对于如何有效地利用流标签来提供QoS支持的研究正处于探索阶段。文献[1]提出了一类用于DiffServ的流标签分类器模型,使用该分类器模型可以完成流标签到包头中Traffic Class字段的映射,提出了两个十分重要的思想:(1)在接入网络或骨干网络边缘根据网络分级模型处理流标签;(2)流标签可以进行聚集(trunk),多个流可以合并成“宏流”(macroflow)。文献[2]针对大规模网络中IPv6流标签使用的扩展性问题,基于网络分域模型提出了一种流标签的聚集和分配机制。该机制在确保流标签使用中唯一性的同时,可以节省计算、存储和通信开销,从而改善流标签使用的扩展性。目前已经有许多关于流标签定义的建议向IETF工作组提出,但仍没有一种满足要求成为标准<sup>[3]</sup>。本文从IPv6多媒体组播通信需求出发,给出了一种流标签分级方案,在此基础上设计实现了QoS保证的组播通信。

### 1 混合式流标签分级定义

文献[4]作为IPv6规范,在数据包头中定义了一个20位的数据流标签字段,如图1所示。在其附录A中,作出了使用流标签的如下要求:(1)未使用流标签时必须将其置0。(2)过时的流标签必须在全网路由器将其状态信息全部清除后才能重新分配。(3)所有属于同一个流的数据包必须具有相同的源/目的地址、跳到跳选项包头、路由选择包头,这样在沿途路由器中只需要以流标签作为查表的索引,即可得知流当前状态并决定数据包的处理和转发,而不必理会包头中其他信息。

版本(4位)	类(8位)	数据流标签(20位)	
有效负载长度位(16位)		下一个包头类型位(8位)	跳数极限位(8位)
源地址位(128位)			
目的地址(128位)			

图1 流标签在IPv6包头中的位置

为了更有效地利用IPv6流标签的作用,根据文献[5]提出了混合方式流标签分级定义。流标签的前3位用来定义不同类型的流标签,如表1所示。

表1 流标签前3位含义

位	含义
000	缺省(数据包的传输不需要任何QoS保证)
001	任意数类型
010	Hop-by-Hop扩展包头值类型
011	多域分类器类型
100	端口号和协议格式类型
101	本文流标签类型
110	保留
111	保留

后17位用来定义多媒体类型和QoS参数。不同应用对QoS参数的要求包括带宽、延迟、延迟抖动、丢包率、缓冲区需求等。其中丢包率和延迟抖动通常保持最小值,不在流标签中考虑。因此,流标签中的QoS参数包括带宽、延迟和缓冲区需求。

(1)多媒体类型。流标签第3位、第4位用来表示媒体类

**基金项目:** 国家部委基金资助项目(413150801); 陕西省自然科学基金资助项目(2004BS100002)

**作者简介:** 陈华胜(1981-), 男, 硕士研究生, 主研方向: 决策支持技术, IPv6多媒体组播通信; 徐 瑞, 博士研究生; 李伟华, 教授、博士生导师

**收稿日期:** 2007-01-06 **E-mail:** xiaochen@hotmai.com

型。如表 2 所示。剩下 15 位定义带宽、延迟和缓冲区。

表 2 流标签第 3 位、第 4 位含义

位	含义
00	音频流及类似
01	视频流及类似
10	预留
11	预留

(2)带宽。用前 5 位表示。根据一般视频会议 15 帧采集速率, 32 KB/s 速率可以满足。5 位全为 1 为带宽最大值。计算公式如下: 带宽 =  $2^B \times 4$  Kb/s,  $B$  是由 5 位表示的十进制数。

(3)缓冲区。方案引入组播缓冲区设计, 大小按数据包数计算, 用带宽之后的 5 位表示。计算公式如下:

缓冲区 =  $2^B \times \text{Packet}$ ,  $B$  是由 5 位表示的十进制数。

(4)延迟。最后 5 位表示应用所能容忍的最大延迟。计算公式如下: 延时 =  $2^B \text{ ms}$ ,  $B$  是由 5 位表示的十进制数。

流标签的细分方案见图 2 所示。

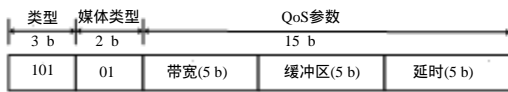


图 2 流标签的细分方案

## 2 QoS 支持的 IPv6 组播通信设计

### 2.1 总体设计

方案主要分成 QoS 处理模块和 IPv6 组播通信模块两大部分, 如图 3 所示。多媒体数据采集主要由专用 API 完成, 视频数据采集可以使用 DirectShow 完成采集, 将所得的二进制帧数据压入缓冲区中。

QoS处理模块侧重研究分组延时。模块依据多媒体数据类型和精度考虑, 如音频数据帧比较小, 但延时要小和速率要高。视频数据帧依据压缩算法不同大小分布各有特点。如果选用的是Divx(MPEG-4)压缩算法, 摄像头使用 176 × 144 分辨率情况下, 压缩后关键帧大小在 3 KB~5 KB之间, 非关键帧均小于 1 KB。综合考虑可以设定 4 级数据包, 即  $2^{6+I}$  Kb( $I = 0, 1, 2, 3$ )。

目前 Windows 平台支持的 IPv6 组播侦听发现(MLD)协议, 是由 RFC2236 定义的与 IGMPv2 等价的 IPv6 协议。MLD 定义了主机和路由器之间交换的一系列报文, 路由器使用这些报文来发现在它所连接的子网上有主机在侦听的多播地址的集合。IPv6 组播通信模块使用 WinSocket 2.0 版本实现, 套接字选取遵守 RFC2553 和 RFC2292 规范。IPv6 组播通信方案见图 3。

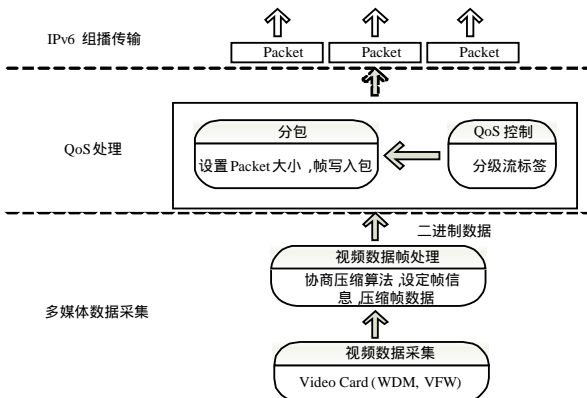


图 3 QoS 支持的 IPv6 组播通信方案

### 2.2 数据结构

### 通用数据包定义

```
struct COMMON_DATA{
    DWORD dwDataType; //媒体类型
    DWORD dwPacketId; //数据包序号
};
```

大小不同的各种数据包可以从 COMMON\_DATA 继承。

### 2.3 分级计算步骤

接收端观察 QoS 参数。发送端包速率为  $P_v$ , 参考调整延时为  $P_d$ , 应该满足  $P_d < P_v$ 。随机间隔  $t$  观察  $d$  包数据包, 第  $i$  包延时时为  $P_{t_i}$ , 定义式(\*):  $\sum_{i=1}^d P_{t_i} < d \times P_d$ 。由于解压播放处理不属于方案考虑的 QoS 范围, 因此计算过程到从缓冲区读出送入处理线程停止。步骤见图 4 所示: (1)数据包到达后先计算时间戳, 得到该包延时  $P_{t_i}$ ; (2)计算公式(\*), 转(3), 并将数据包压入缓冲区, 如果缓冲区满足不了需要, 调整缓冲区; (3)满足公式(\*), 查看  $i$  是否到达  $d$ , 是则清 0, 把比当前好的时延赋予  $P_d$ , 否则继续累加,  $P_d$  不变; (4)满足不了公式(\*), 将缓冲区数据, 延时等参数写入 CTRL\_FLOWLABEL 中, 向发送方返回, 转(6); (5)线程从缓冲区取出数据处理; (6)计算结束。

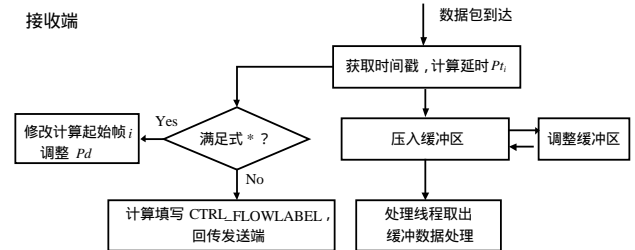


图 4 接收端分级流标签步骤

由接收端向发送端返回 CTRL\_FLOWLABEL, 在发送端触发包调度和改写流标签机制。

## 3 测试模拟及分析

方案参考 Microsoft 关于 IPv6 实验室建议搭建实验网络, 目前 Windows XP 和 Server2003 版的 IPv6 协议均不支持 IPv6 组播转发, 因此由计算机模拟的路由器替换为支持 IPv6 思科路由器。视频数据用 DivX 压缩算法压缩, 根据 179 × 144 分辨率处理后帧特点, 选用第 4 级数据包进行测试。考虑到实际网络状况, 方案选用 10Mb/s 无线, 10Mb/s/100Mb/s 有线 3 个节点进行传输测试。测试时有线接收端观察无线接收端, 无线接收端观察的是 10Mb/s 有线发送端。测试结果表明, 方案能根据网络状况有效地进行调节, 延时在满意的区间内起伏。32 KB/s 的情况下有好的延时表现, 而 128 KB/s 下波动较前者大, 不过总体性能还能满意。测试中还发现, 有线向无线发送的过程中, 随着速度的增加丢包率有所提高。128 KB/s 下达到 0.14%, 见图 5、图 6。

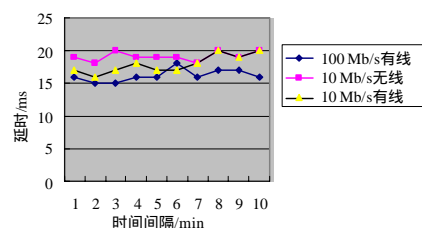


图 5 32 KB/s 下的延时表现

(下转第 124 页)