Effective I/O Scheme Based on RTP for Multimedia Communication Systems

Nam-Sup Park and Chong-Sun Hwang
Department of Computer Science and Engineering, Korea University, Seoul, Korea
E-mail: {uspark, hwang}@disys.korea.ac.kr
Received June 8, 2005; revised December 9, 2006.

Abstract The prime standard for audio/video transport in IP networks is the Real-time Transport Protocol (RTP), and it is targeted at useful services for the transport of real-time multimedia data. RTP was originally designed for use in multicast conferences, using the lightweight sessions model. RTP (in particular, the data part) is so tightly coupled to the application that a number of people have developed libraries that implement RTP. However, little is known about the RTP overheads between user area and kernel area within operating system. Actually, unnecessary copying between user area and kernel area lowers the system efficiency. In this paper, we present the design and implementation of Enhanced Multimedia Input/Output Scheme based on LINUX. We brought focus to the crossover architecture supporting RTP. Our contributions are able to be summarized into two components: 1) Enhanced Input/Output (EIO) scheme based on LINUX improves the transmission speed by reducing the overheads generated from data copy and context switch between user area and kernel area. And this enables server-based system to transport multimedia data more efficiently. 2) Furthermore, Enhanced Input/Output scheme with RTP (EIORTP) scheme supports efficient multimedia data transmission architecture. The two schemes improve the performance of massive multimedia data transmission.

Keywords kernel area, system call, EIO, RTP, EIORTP

1 Introduction

There has recently been a flood of interest in the delivery of multimedia services in the Internet. The growing popularity of Internet telephony, streaming audio and video services (such as those provided by Real Audio), and the Mbone are all indicators of this trend. For the purpose of supporting these applications, standards are being developed to insure the inter-operability. These include the ITU-T H.323 specification for Internet telephony, the Session Initiation Protocol (SIP) for multimedia session initiation, and RTSP for controlling multimedia servers on the Internet. Moreover, RTP along with its associated profiles and payload formats, is the key standard for multimedia transport in IP networks.

RTP, defined in [4], can be used for transporting common formats such as PCM, GSM, and MP3 for sound and video formats. Also it can be used for transporting proprietary sound and video formats. It is also complementary to other important real-time interactive protocols. Besides, a number of people have developed its libraries only in application area. However, little is known about the crossover issue between kernel area and user area. In this paper, we focus on the overheads caused by frequent data copy and context switch between kernel area and user area.

On the other hand, large scale multimedia-on-demand (MOD) servers, which are capable of providing the independent and interactive access to a large number of concurrent clients having a vast amount of multimedia information, will be required to enable a widespread deployment of multimedia applications. MOD servers providing location transparent, interactive concurrent access to hundreds of thousands of independent clients will be an important component of the future information super-highway infrastructure. Many researches related to the large scale MOD servers have been proposed and implemented. These researches especially concentrated on the reduction of the overheads generated from data copy in the I/O subsystem.

In this paper, we propose the crossover issue about RTP. In consideration of previous work, we propose that EIO reduce the overheads caused by data copy and context switch between the user area and the kernel area. Furthermore, EIORTP supports the efficient soft real-time multimedia data transmission architecture. We have designed and implemented an instance of EIORTP scheme based on the LINUX operating system. Compared with the existing LINUX, we achieved about 14.28%–25.36% improvements in the transmission speed rate.

The remaining of this paper is organized as follows. Section 2 discusses prior approaches of RTP and I/O subsystem. In Section 3 we propose an enhanced Input/Output scheme based on LINUX. The experimental results of EIORTP are evaluated in Section 4. Furthermore, the theoretic analysis of the EIO/EIORTP scheme is depicted in Section 5. Finally, Section 6 concludes this paper.

2 Prior Approaches

Existing real-time systems generally implement fixed-priority and preemptive schedulers. Thus, they are not suited for a general-purpose system shared by
real-time (continuous-media), interactive and batch applications\textsuperscript{13}. On the other hand, current practice in general-purpose system is insufficient for the desired quality of delivery of continuous media. Therefore, we focus on the memory access and data copy based on legacy general-purpose system such as LINUX.

We analyze MARS system\textsuperscript{9} and RTP in this section. MARS project rectified data copy problems and improved the kernel architecture based on Multimedia data processing, and RTP is widely used to the transparent protocol for voice and video on the Internet.

### 2.1 Existing File I/O Scheme and MARS

Fig.1 illustrates the layered architecture used by the storage and network I/O systems of current LINUX\textsuperscript{16,17}. As depicted in Fig.1, existing file and network I/O have the following shortcomings. Unnecessary data copy is a significant demerit of the performance. Clearly, in a MOD server implemented in a user area, the data transfer path from a disk to the network interface involves two memory copies: the first copy (in response to a read() call) moves data from the kernel buffer cache to a user area buffer. The second copy (in response to a send() call) by the socket layer is processed on the contrary. This approach works well for small-sized accesses in a general purpose I/O subsystem, such as traditional text and binary file accesses.

![Conventional data transmission mechanism.](image)

However, multimedia data such as audio, video and animations do not need to process any data copy, that is, unnecessary additional data copy results in the drawback of the performance of the multimedia data processing. Large amounts of data copy from context switch not only take processor time but also waste the system memory and the bus bandwidth. As a result, the buffer cache blocks can be reused only if several sessions reading the same file are able to process the locking algorithms in regular sequence.

MARS has the following scenario as shown in Fig.2. The top side shows the context switch, and the bottom side shows copy operations. By using stream\_read(), MARS has cut in half the amount of data the kernel has to copy. This yields reasonably good results when a lot of data is being transmitted as follows.

Step 1: the stream\_read() system call causes the file contents to be copied into a kernel buffer by the DMA engine. Then, the buffer is shared with the user process, without any copy being performed between the kernel and user memory spaces.

Step 2: the stream\_send() system call causes the kernel to copy the data from the original kernel buffers into the kernel buffers associated with sockets.

Step 3: the third copy happens as the DMA engine passes the data from the kernel socket buffers to the protocol engine.

![Data copy and context switch in two system calls.](image)

MARS rectifies these limitations as described in Fig.1 and proposes a very sophisticated multimedia processing system. It provides the following merits: 1) a new kernel buffer management system called MMBUF (Multimedia Memory Buffer) which shortens the data path from a storage device to network interface, 2) fair queuing within the SCSI driver for equitable resource sharing between real-time and non-real-time streams, and 3) integration of these new OS services with a CPU scheduling mechanism named RTU\textsuperscript{8}. However, MARS has the following drawbacks: 1) it has hidden pitfalls when using the MMBUF and write method. In case of a bad memory access, the write system call will be interrupted by the bus error signal SIGBUS, and 2) it needs the additional context switch.

### 2.2 Multimedia Data Transmission Based on RTP

RTP\textsuperscript{4} provides end-to-end delivery services of real-time data such as interactive audio and video. RTP