Implementation of Adaptive Buffer in Video Receivers Using Network Processor IXP 2400

Kandasamy Anusuya, Karupagouder Thirunavukkarasu, and Subha Sundaresan
Faculty in Electronics and Communication Engineering, PSG College of Technology, India

Abstract: *New services such as non-interactive video streaming, which demand higher bandwidth have become popular with the introduction of broad band networks. But the quality of streamed video is impaired by the factors such as packet loss, congestion, delay and jitter in the network. Hence, to improve the video quality, an adaptive data rate based play out buffer management scheme is proposed. It mainly considers the play back time and play out buffer size of the video player used at the receiver. Moreover in recent designs, the Network Processors are used in a wide range of networking embedded systems, including multi service switches, routers and so on. The Network Processorses are fully programmable Processors which performs number of simultaneous operations. This ensures full network performance and also accommodates complex services on packet basis. Hence, the proposed scheme is implemented in the Network Processors, IXP 2400 and its performance is evaluated for different data rates. It is observed that the packet loss is reduced and the buffer utilization is improved.*

Keywords: *Buffer utilization, network processor, play out buffer, play back time, video streaming*

Received February 6, 2007; accepted November 6, 2008

1. Introduction

The video streaming rate has been increased with the increased bandwidth of the broad band networks. Because of this, the video player at the receiver end suffers from managing the size of its play out buffer. There are two approaches available already to solve this problem [5]. The first approach adapts the streamed content to the current network conditions at the end terminals and is called end-to-end QoS control. It mainly concentrates on congestion control, error control and error-resilient coding. The second approach known as network-centric has queue management mechanisms and transcoders. The queue management mechanism prevents congestion by dropping the excessive packets, and the transcoding in nodes create a new version of the streamed content adapted to current network conditions.

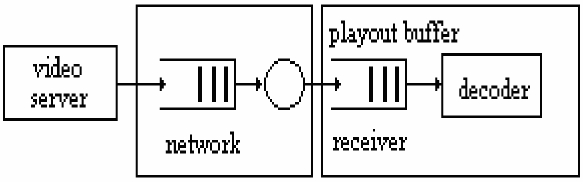

Figure 1. Video transmission.

Figure 1 shows the simplified transmission path [2] for video streaming. IP packets from the video server are transmitted through the network. These packets undergo the delay due to processing, buffering and propagation in the network. Hence to overcome this delay, the packets are stored in the play out buffer of the receiver and are played back during its play-back time. If the transmission rate of the video server increases, then the play out buffer capacity exceeds the limit at the video player. Therefore, the buffer size at the receiver end has to be varied dynamically according to the transmission rate.

In this paper, an adaptive buffer management technique to reduce the packet loss at the video player is discussed. Further, the maximum buffer size that can be offered at the maximum transmission rate of the video packets is also considered. This model is implemented with IXP 2400 Network Processor (NP). Since IXP2400 has multi threading facility, more number of video packets can be handled simultaneously. The paper is organized as follows. Since the data rate of the received video packets directly affects the adaptive buffer size, the buffer management is explained in section 2. The architecture and salient features of IXP 2400 are discussed in section 3. Section 4 proposes the system model and its implementation in NP is explained in section 5. The performance of the algorithm with the non-adaptive and adaptive buffers are evaluated and compared in section 6. Finally, section 7 concludes the article.

2. Buffer Management

The transmission of video over IP-based networks requires the segmentation of video frames into IP-packets. Generally, the maximum size of an IP-packet is constant throughout the path and is smaller than the average size of a video frame. The transmitted data rate is measured by counting the number of packets reached at the receiver during the time interval of T seconds along with their packet size. Since the NP receives packets of fixed size called m-packets, it is easy to measure the data rate as shown in equation 1.

Data Rate = (Number of packets received in T Sec xm packet size) / T seconds (1)

Consequently, there are two types of buffer management strategies - the uncontrolled buffer management and controlled buffer management. In the first scheme, the recently arrived packets are dropped from the tail end, whenever the buffer over flows. In the second scheme, once a buffer over flow condition is about to reach, the buffer size is increased to accommodate the new packets by also considering the maximum buffer size. The conventional video players like VLC players have an internal buffer capacity of 300 bytes and the uncontrolled buffer management technique is used. In this paper, the controlled buffer management technique is proposed and the buffer size is calculated as in equation 2.

Buffer size = Data Rate x Playback time (2)

As the playback time of the video players varies from player to player, a dynamic buffer size has to be allocated for the varying data rate and play back time.

3. IXP 2400 Network Processor

IXP2400 NP combines the performance of a high speed processor core with the eight programmable multithreaded micro engines [1, 3]. Highlights of the processor include, (1) Integrated high-performance, low-power 32-bit Intel XScale core for processing complex algorithms, route table maintenance, and system-level management functions, (2) Hyper Task Chain processing technology enables deep packet inspection via software pipelining at 2.5 Gbps, (3) eight fully programmable micro engines for packet forwarding and traffic management on a single chip, supporting 5.4 Giga-operations per second, (4) deep packet inspection: 14 million en-queue / de-queue packet operations per second, supporting packet processing of minimum 40 bytes PoS packets, (5) standards-based interfaces, (6) comprehensive development environment, and (7) Low power consumption. Figure 2 shows the block diagram of IXP2400. The Micro Engines (MEs) do most of the programmable Packet processing [4]. The MEs have access to all shared resources (SRAM, DRAM, Media Switch Fabric *etc*). It could be used for the designs of wired networks. It does not support wireless environment. The Developer Workbench is an Integrated Development Environment (IDE) for IXP2400 Micro Engines.

The micro codes of the proposed algorithm are generated, edited, compiled, linked, debugged and run on IXP 2400 micro engines using this IDE.

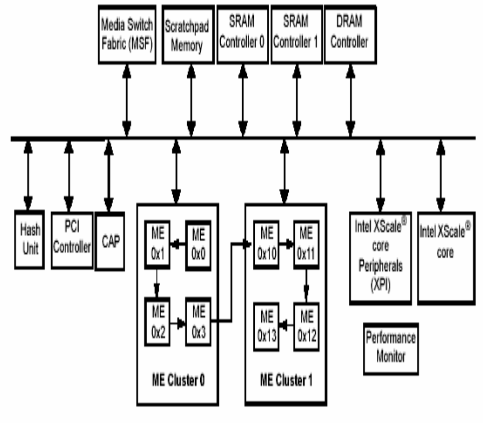

Figure 2. Block diagram of IXP2400.

4. Proposed Model

Figure 3 shows the model of the proposed system. In this model, the data rate is measured by the Data Rate Estimator and the required buffer size for the measured data rate and play back time is calculated by the Adaptive buffer estimator at the receiver. Hence the size of the play out buffer is decided dynamically by considering the maximum buffer size.

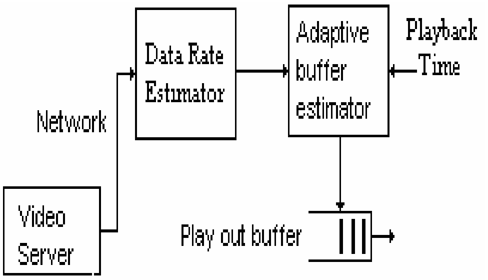

Figure 3. Block diagram of the proposed model.

5. Implementation

This model is implemented in the Developer Work bench of IXP2400 NP. The model uses only the four threads of a single Micro-engine. One thread is for receiving the packets and the remaining three are used for Data Rate Estimation, Adaptive Buffer Size calculation and data Transfer to the player respectively. The implementation procedure is shown in Figure 7. The maximum buffer size offered for the worst case for maximum transmission rate is fixed. If the required buffer size exceeds this value, then the maximum buffer size is allocated as the required buffer size. Under this condition, the packet loss is unavoidable.

6. Performance Analysis

The algorithm is executed for the buffer size of 300 to 400 bytes. The data rates of 64Kbps, 128Kbps, 256Kbps, and 512 Kbps are considered. Moreover, the average play back time is fixed at 10ms. The buffer utilization is observed for the non- adaptive and adaptive buffered algorithms. It is noted that the buffer utilization decreases with the increase in the data rate and this results are plotted in Figures 4 and 5, respectively. The performance of the adaptive buffer algorithm is better than the other one. But, it seems to be poor at 512 Kbps, because of the buffer over flow (that is, the maximum buffer size is exceeded) condition. Even though the operating frequency of IXP 2400 is 600 MHz and its memory support is upto 2 Gb, the proposed algorithm is implemented only for the conventional video players with the memory capacity of 300 to 400 bytes. This factor limits the data rate to 512 Kbps. Figure 6 compares the buffer utilization of non-adaptive and adaptive algorithms.

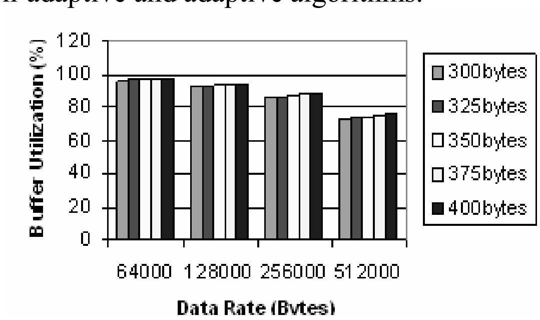

Figure 4. Buffer utilization for non- adaptive algorithm.

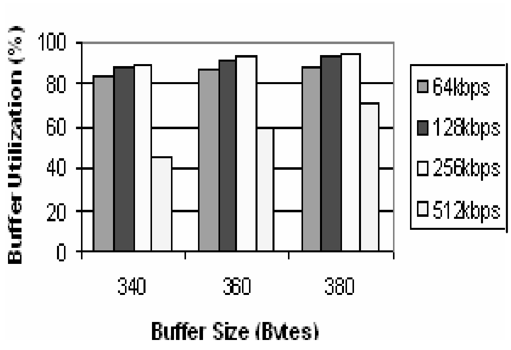

Figure 5. Buffer utilization with adaptive algorithm.

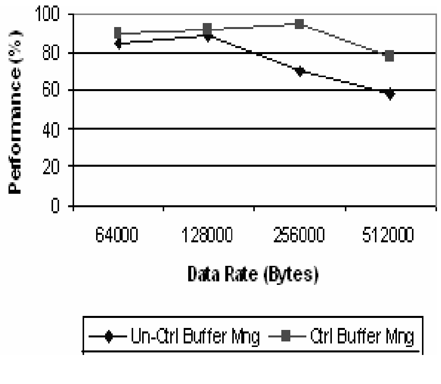

Figure 6. Performance comparison of algorithms

Configure IXP2400 to activate only the even threads.

Set the m-packet size as 128 bytes.

Rx Driver:

Initialize the RX_THREAD_FREELIST, to interrupt Thread-0 whenever a new packet arrives.

Wait for signal from MSF interface to indicate the data reception in RBUF.

Check for error in the received packet using the RX_STATUS_WORD of CSR register.

If there is an error, discard the packet; otherwise read data from MSF and store it in the transfer registers and activate Thread-2.

Data Rate Estimator:

Increment the number of packets received and the read the size of the packet in bytes.

Calculate the data rate using the formula given in Equation (1). Store this value in the scratch pad memory.

Activate Thread-4.

Calculation of Required Buffer Size:

Using the measured data rate and average playback time, calculate the required buffer size and store it in the scratch pad memory.

If required buffer size is greater than the maximum buffer size, then allocate the maximum buffer size as the required buffer (up to the maximum memory support of DRAM / SRAM in IXP 2400)

Active Thread-6.

Data Transfer Unit:

Allocate a memory block size given by Thread-4 in DRAM.

Transfer the data in the Transfer Register to the memory block and transmit through ports.

Activate Thread-0, to receive the next packet.

Figure 7. Implementation procedure.

7. Conclusion

The performance of the proposed algorithm is comparatively better in minimizing the packet loss and improving the buffer utilization under varying transmission rates. Hence, this technique helps a conventional video player to accept the variable data rates and allocate the play out buffer efficiently. Moreover the system uses minimum amount of memory during the low transmission rates.

This model could be enhanced with the delay prediction at the receiver end. With that enhancement, the delay encountered by the video packets before reaching the player could be exactly predicted and based on that, the buffer allocation could be done more precisely.

Reference

[1] Amy R., Subhabrata S., and Jacobus M., "*Video Quality Estimation for Internet Streaming*", AT&T Labs-Research, 2006.

[2] Douglas E., *Network System Design Using Network Processor*, Pearson Education Publication, 2003.

[3] Erik J. and Aaron R., *IXP2400/2800 Programming*, Intel Press, 2004.

[4] Intel Corporation, *Intel IXP2400/2800 Network Processor Hardware Reference Manual*, 2002.

[5] Zigmun O. and Marc C., "Enhancement of Video Streaming QoS with Active Buffer Management in Wireless Environments," *in Proceedings of the 13th European Wireless 2007*, Paris University of Stuttgart, 2007.

Kandasamy Anusuya has her BE in electronics and communication engineering from Government College of Technology, Coimbatore, India in 1993, and ME in applied electronics from PSG College of Technology, Coimbatore, India in 2004. Her area of interest includes embedded systems, wireless networking and multi core architectures.

Karupagouder Thirunavukkarasu is a scholar of Electronics and Communication Engineering, PSG College of Technology, Coimbatore, India. His area of interests includes secured routing, and switching systems design.

Subha Sundaresan has her BE in electronics and communication engineering, ME in applied electronics from the Anna University, India in 1984 and 1986, respectively. She completed her PhD from Bharathiar University, Coimbatore in 2000.