VIOLA: Video on Local-Area-Networks

Y.B. Lee and P.C. Wong
Advanced Network Systems Laboratory
Department of Information Engineering
The Chinese University of Hong Kong, Hong Kong
{yblee, pcwong}@ie.cuhk.hk

Keywords

Video Server, Server array, Multimedia Network, Network Protocol, MPEG-I, Video On Demand.

Abstract

We describe a network system architecture called VIOLA (Video-on-LANs) for video-on-demand services on a local area network. The architecture is based on a switched-network topology and a new server array concept, which together allow the system to be scalable from tens to thousands of users. We propose a Buffered Datagram Layer (BDL) for shielding underlying network dependencies, while supporting efficient datagram transmissions for high-bandwidth traffic; a Reliable Datagram Protocol (RDP) for carrying video requests reliably from clients to servers; and Video Transport Protocol (VTP) for efficient video transmissions and receptions. Video is block-interleaved at the server array. We present some preliminary results on a VIOLA prototype being implemented in the Chinese University of Hong Kong.

1. Introduction

Video-on-demand service has many exciting applications. Examples are movies-on-demand, music video (karaoke) on-demand, video magazines, video kiosks, computer-aided-training, and video library, etc. For this reason, there has been a lot of research in recent years on video distribution services and technologies. Hodge [1] gives a general description on the architecture, systems, and applications for video-on-demand service. Up to now, most of the work has been focused on two types of systems. One is for large scale video-on-demand applications [1,2]. Such systems can serve thousands of users through the use of high-end dedicated hardware (e.g., video servers and ATM networks) for video storage and transmission. Another is small scale systems for supporting general multimedia services on a local or wide area computer network [3]. Due to the limited network and server bandwidth, the number of stations is usually small, and the video quality (i.e., frame rate, frame size) is limited. One exception is StarWorks™ proposed by Tobagi and Pang [4] for distributing video on a local area network using a high-performance server. However, such a single-server approach clearly has its limits in supporting more simultaneous users. In this paper, we describe the VIOLA project being done at the Chinese University of Hong Kong. This project was initiated to investigate, design, and implement a scalable architecture for video distribution on a local area network.

In the following, Section 2 presents the system architecture of VIOLA. Section 3 presents the network protocols. Section 4 describes the new server array concept for video distribution. Section 5 presents some preliminary results on our implementation of VIOLA. Section 6 concludes the paper.

2. System Architecture

Our architecture is based on a switched-network topology (Figure 1) connecting a server array to the client stations. The switch fabric can either be an ATM or Fast Ethernet switch, which switches traffic between servers and clients. In a switched network, each network segment has full-speed of transmission. So the aggregate switch throughput is much greater than the individual segment speed. In our architecture, each server can have one or more links connecting to the switch, whereas several clients may share one single link. The bandwidth of server links and client links may be different. For example, each server may use two 100 Mbps Fast Ethernet links, whereas several clients may share a 10 Mbps Ethernet segment. Such an architecture caters for the asymmetric bandwidth requirements of servers and clients, and can support more users by connecting small switches into a large switch fabric, and by adding more servers into the server array.

A VIOLA system consists of VIOLA Client Stations, VIOLA Server Stations in the VIOLA Server Array, and a VIOLA Manager Station. VIOLA server stations are high-end computers equipped with disk arrays, and have one or more network adapters. The client stations are low-end personal computers with a network adapter and an MPEG-1 video decoder. The video data is processed, decoded, and played back at the video monitor of a client, or at a TV set through a video converter. The VIOLA manager handles the system administration tasks such as user authentication, admission controls, accounting, and video file management, etc.

We use the VIOLA prototype implemented in CUHK as an example, which is based on a 16x16 LANNET Ethernet Switch. Each VIOLA server is a Pentium-90 with two Ethernet links. We have four servers, each has 4GB harddisk for video storage, making a total of 16GB storage (~1800 minutes of 1.2Mbps MPEG-1 video). We have eight network segments for connecting client stations, each serving five client stations. So a total of 40 client stations can be supported by four servers. The client-server ratio is 10, and the utilization of both server and client segments are around 60% (6Mbps). We are in the process of ordering a Fast Ethernet switch. Once it is available, we will use 100 Mbps links for servers. We expect the client server ratio will be increased significantly.

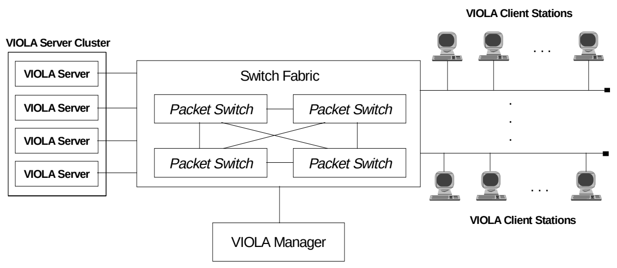

Figure 1 - VIOLA System Architecture

3. Network Protocols

Traditional connection-oriented protocols such as TCP have many drawbacks in video transmissions. Firstly, connection-oriented protocols only support point-to-point connections. In VIOLA, we need one connection from every server to every client, and each client has to demultiplex incoming data streams from multiple servers into one coherent video stream. The software complexity is significant to maintain so many connections. Secondly, connection-oriented protocols guarantee an in-sequence data delivery to the client application. If a packet is lost, future packets arrived are blocked in the protocol layer. This causes delay in delivering video blocks to the playback hardware, and possibly subsequent more packet drops and hence retransmissions at the protocol layer. Thirdly, video data are delay sensitive, so overdelayed retransmissions will not be useful. Finally, these protocols rely too much on backward acknowledgement traffic to maintain an error-free connection. This causes significant processing requirements at the servers. On the other hand it causes significant collisions and reduces network throughput for contention-based networks such as Ethernet.

Connectionless protocols such as IP and UDP have no provision for packet loss recovery. This works well in frame-by-frame compressed video as a little packet loss affects only one frame. Yet in sophisticated compression algorithms (e.g., MPEG) which exploits inter-frame compression, the loss of a packet may affect several video frames, causing objectionable flickers and patches at the output video. While various reliable datagram algorithms have been proposed [5,6], they were designed to meet the requirements for data traffic and wide area networks, so are less appropriate for transmitting time-sensitive control and video packets.

In the following, we propose a novel stack of protocols for video distribution (Figure 2): namely, the Buffered Datagram Layer (BDL), the Reliable Datagram Protocol (RDP), and the Video Transport Protocol (VTP). These protocols can be used to support other kinds of delay-sensitive, stream-type traffic as well.

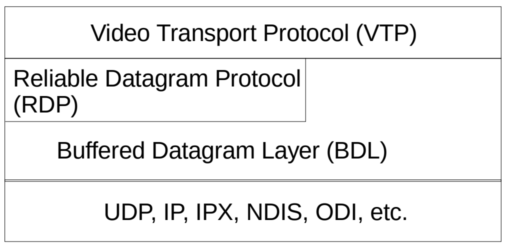
Video Transport Protocol (VTP)
Reliable Datagram Protocol (RDP)
Buffered Datagram Layer (BDL)
UDP, IP, IPX, NDIS, ODI, etc.

Figure 2 - Network Architecture

3.1 Buffered Datagram Layer (BDL)

The BDL is designed to shield the underlying network dependencies and gives a standard API for high-bandwidth datagram transmissions. BDL consists of a resizable circular buffer that is tightly coupled with the VTP layer. Through a specialized memory manager and a technique we called Packet-Header-Reservation (PHR), there is no need of memory copies when data is passed from user-code to the network driver.

3.2 Reliable Datagram Protocol (RDP)

RDP provides a reliable datagram service on top of BDL for control traffic between servers and clients, using an Adaptive Linear Timeout (ALT) algorithm for retransmission. As VIOLA is a LAN-based architecture, the network delay is very small and packets are always delivered in-sequence. We therefore limit RDP to handle packet loss or duplicated packets during transmissions. Additionally, control packets are further reduced through piggybacking acknowledgements within video and control packets.

3.3 Video Transport Protocol (VTP)

VTP consists of two parts: VTP Server and VTP Client. Data packets are transmitted using BDL services while control packets are transmitted using RDP services. At the client, incoming video data are inserted into a circular video buffer composed of fixed size video blocks. VTP incorporates mechanisms for flow-control, traffic shaping, and packet-lost recovery. Specifically, many-to-one connection type is supported directly at the client. Secondly, lost packets will not block data transmission at the server. Thirdly, over-delayed lost packets are not retransmitted. Finally, the upstream control traffic from clients to servers is minimized.

3.3.1 Flow Control

We need flow control mechanisms to ensure that the buffers at the client stations are neither overflown nor underflown. Rate-based flow control can give a smoother traffic and require fewer control packets going back to the server. However, rate control is difficult to implement in most desktop operating systems which do not have real-time capability. In VTP, we use credit-based algorithm together with prefetch-buffering for flow control. In this scheme, the client will pre-fetch a full-size queue of video packets before submitting video blocks to the playback device. At each time the playback device requests a new block, the client will submit a new block to the device, and at the same time sends a request packet to the server for a new block. We use a reasonably large block size for each request to minimize the processing at the servers (each packet causes an interrupt at a server), and to reduce backward traffic. To ensure playback continuity, we assign a client buffer large enough to cater for burst requests from the playback device and for delays in receiving video packets from the server.

3.3.2 Traffic Shaping

The flow control mechanism attacks the starvation problem at the client buffer. However, congestion can occur at the underlying network hardware, at the switch fabric and at the client station. The problem is amplified by the bursty nature of video traffic. In our experiments, we found that large-size bursts can cause significant packet loss at the client. This is because the network buffer is limited in capacity. If the client is busy submitting video blocks to the playback device, it will not be possible to read packets from the network buffer in time.

To reduce the possibility of congestion, we propose the Batch-Round-Robin (BRR) algorithm to perform traffic shaping at the server. The server has a fixed number of transmit queues. Each transmit queue can store a video block for a particular client. When a transmit queue is empty, the server will remove a request from the request queue, read a disk block and store into the transmit queue. At each round of service, the server will send a predefined amount of packets in a burst from one transmit queue, and then switch to another transmit queue. Artificial delay is added to shape the traffic if there are too few clients for interleave transmission. The BRR algorithm ensures that the size of each packet burst is limited, packet bursts are interleaved, and the overall rate of bursts can be controlled.

3.3.3 Packet Loss Recovery

From preliminary observations with our prototype, we found that there is still a very small packet loss at the client, even if traffic shaping is applied. To cope with this, we use a scheme called Time-Constrained-Retransmissions (TCR), taking into account the time-sensitive characteristic of video traffic. First, TCR will retransmit only if a packet has a high probability of being received in time. Therefore overdelayed packets will not be retransmitted. Secondly, lost packets will not block future packet receptions. We use a Direct-Buffer-Insertion (DBI) technique that packets are inserted with a pointer directly into the video playback buffer. Thirdly, retransmission requests are piggy-backed on video request packets to reduce the number of control packets. The TCR scheme is therefore coupled with the video request transmission. Whenever the client wants to transmit a video request, it will scan through the list of received video packets. If there are losses that can be recovered before playback occurs for that video block, retransmission requests will be sent to the server for retransmissions. We found that after using the TCR scheme, all packet losses are recovered as observed from the video playback.

4. VIOLA Server Array

As a counterpart to interleaved memory and disk arrays, we propose a server array for video storage. Video data is block-interleaved into an array of servers. The VIOLA client station requests video blocks from the servers on a round-robin basis. Every server operates independent of each other so that no inter-server communication overhead is required. There is no data duplication among the servers. We are now investigating server-level fault-tolerance architectures by using redundant servers. Within each server, data packets for multiple clients are scheduled for transmission using the algorithm as described in Section 3.3.2. This can reduce network congestion when multiple servers send data to the same client simultaneously. The server array concept allows the system to be scalable for supporting more clients, simply by adding more servers.

5. Preliminary Results

At the time of this writing, we have implemented a fully operational VIOLA prototype, named *NETBLASTER*, which supports server array, multiple-clients operation (Figure 3). NETBLASTER allows us to test our protocol suite and to evaluate the system performance. We discuss here the results so far and some of our experiences.

5.1 Packet Loss

We found that packets could indeed get lost at the client station when other tasks (e.g., video playback, disk reads) are being executed, especially those involving hardware I/O. Moreover, the packet loss probability varies greatly with different network cards. If a server sends a longer burst of video packets while the client is playing video, the client will lose more packets (Table 1). The loss can be significant even if a Pentium 90 PC and an Intel Pro 10/100 network adapter is used at the client station. Therefore we need traffic shaping and loss recovery schemes as described in Section 3.3.

5.2 Memory Management

In VIOLA, vast amounts of data are processed and passed around in memory. Therefore the overhead in allocating and deallocating memory blocks becomes significant. We observe that general-purpose memory management techniques could not satisfy the processing requirements. Therefore we designed and implemented our own memory manager: FastMemoryManager (FMM). Our measurements show that the FMM outperforms the general purpose memory managers provided by the OS and compiler by 30 times or more (Table 2). This is possible as the FMM only handles fixed-size memory blocks. This is the reason why we use fixed-size video blocks instead of the conventional variable-size frame-based transmissions.

5.3 Server Capacity

Using all the techniques described above, we found that a single Pentium-90 server can support up to 10 client stations on two Ethernet segments. Each client is playing full-motion, TV-quality, 1.2 Mbps MPEG-1 video. Figure 4 shows the results for one server supporting up to ten clients. As can be seen from the figure, the server CPU utilization is roughly proportional to the number of active clients. Note that the server is tested using two 10Mbps PCI Ethernet cards, each segment serving five clients. At the time of writing, the server array is just completed and we are waiting for the arrival of a Fast Ethernet switch. We hope that we will publish more results in a later paper.

6. Conclusion

We have described a novel architecture called VIOLA for supporting video distribution on LAN. With currently available hardware and software, we developed a prototype that can provide full-motion MPEG-I video services with interactive user controls. The system runs on standard desktop PCs (486s or above) and standard operating systems (Microsoft Windows). Hence the system can run concurrently with other PC applications. We have described the system architecture, the network protocols, and the server array mechanisms for supporting a scalable video-on-demand system.

7. Acknowledgement

This research is partly supported by the Industrial Department of the Hong Kong Government, and the Ho-Sin-Hang Endowment Fund.

References

[1] W. Hodge, S. Mabon, J.T. Powers, Jr., "Video On Demand: Architecture, Systems, and Applications," *SMPTE Journal*, September 1993, pp. 791-803.

[2] H. Armbrüster, K. Wimmer, "Broadband Multimedia Applications Using ATM Networks: High-Performance Computing, High-Capacity Storage, and High-Speed Communication," *IEEE Journal on Selected Areas in Communications*. vola. 10(9), 1992, pp. 1382-1396.

[3] S. Gibbs, D. Tsichritzis, A. Fitas, D. Konstantas, Y. Yeorgaroudakis, "Muse: A Multi-Media Filing System," *IEEE Software*, 4(2):4-15, March 1987.

[4] F.A. Tobagi, J. Pang, "StarWorks^{TM} - A Video Applications Server," *IEEE COMPCON* Spring '93.

[5] C. Partridge, "Implementing the Reliable Data Protocol (RDP)," In *Proceedings of the 1987 Summer USENIX Conference*, Pheonix, Ariz., 1987, pp. 367-379.

[6] W. Steven, *Unix Network Programming*. Addison-Wesley, Reading Mass., 1991.

Burst Size (Packets)	Loss (1.5Mbps)
10	0%
20	~ 2%
50	~ 10%
100	~ 15%

Table 1 - Burst size versus Packet-Lost

Memory Manager	# of Blocks x Size (Bytes)	
	100x10x1472 (e.g. Packet)	100x10x32384 (e.g. Disk)
MS-Windows *GlobalAlloc/Free*	523 ms	844 ms
Borland C++ *new/delete*	493 ms	967 ms
FastMemoryManager *Alloc/Free*	15 ms	15 ms

Table 2 - Memory Management Performance Comparisons (486DX-66)

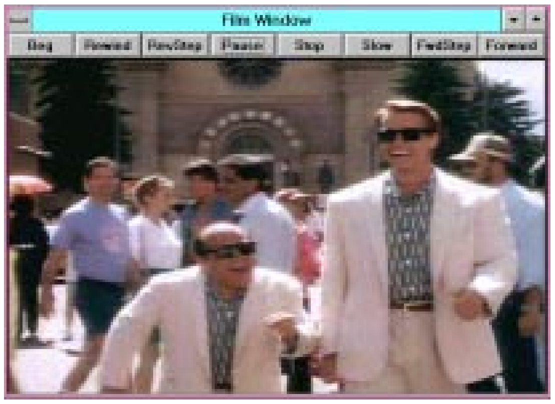

Figure 3 - NETBLASTER playing a sample MPEG-1 stream in a Film Window.

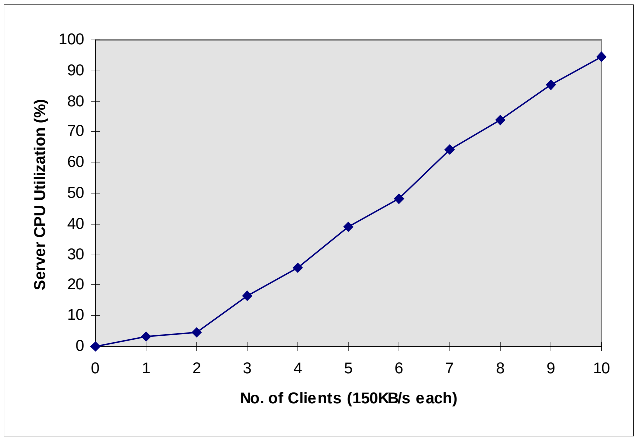

Figure 4 - VIOLA Server Utilization (P5-90, 32MB RAM)