

실시간 영상 다중 전송을 위한 서버의 구현

김 철 우[†] · 이 완 직^{††} · 이 선 우^{††} · 이 경 호^{††} · 한 기 준^{††}

요 약

영상 미디어 전송에 있어 영상 획득, 압축, 전송의 과정이 한 시스템에서만 이루어진다면 실시간 멀티캐스트 전송이 이루어질 수 없을 뿐만 아니라 여러 네트워크를 거치는 동안 전송 지연이 커진다. 본 논문에서는 영상 획득, 압축, 다중 전송의 부하(Load)들을 병렬처리하는 실시간 영상 다중 전송 서버 시스템을 구현하였으며 테스트 크간 처리 속도 차이에 의해 발생하는 흐름 제어 문제를 해결하기 위한 메커니즘을 제시하였다.

An Implementation of Real time Video Multicasting Server

Chul-Woo Kim[†] · Wan-Jik Lee^{††} · Sun-Woo Lee^{††} · Kyung-Ho Lee^{††} · Ki-Jun Han^{††}

ABSTRACT

Real-time video multicast using a single computer system is a difficult task because capturing, compressing and delivering video causes a long delay in internetworking environments. This paper describes an implementation of the real time video multicasting server which processes video capture, compression and multicasting in parallel. This paper also presents a flow control mechanism to solve a problem caused by speed difference between the three tasks.

1. 서 론

멀티미디어는 컴퓨터 통신, 방송 매체 그리고 TV 등의 가전 제품이 하나로 통합되어 종합적으로 서비스를 제공할 수 있는 정보 전달의 총체적 역할을 할 수 있게 되었다. 기존의 저속 통신망으로는 고해상도의 그래픽, 음성, 영상 등의 다양한 서비스를 충족시키지 못하여 고속 대용량의 정보를 전송하는 고속 통신망이 출현했고, 이 하부 통신망의 발달로 인하여 멀티미디어 산업이 눈부신 전진을 보이고 있다[1, 2].

고해상도의 영상, 음성, 동영상 등의 멀티미디어 데

이터들은 용량이 크고 실시간 처리를 요구하기 때문에 이러한 서비스요구를 충족시키기 위하여 B-ISDN, FDDI, DQDB 등의 광대역폭의 고속 하부 통신망을 사용하게 된다.

고속망환경을 이용할 경우라도 여러 망을 묶은 인터넷워킹(internetworking) 환경에서는 서로 다른 타입의 망들을 거치게 되는데 상대적으로 속도가 느린 망에서 병목현상이 일어나고 전체 망의 속도는 낮은 처리율(Throughput)을 가진다. 그리고 다중 전송의 경우에는 수신자가 많으면 많을수록 큰 데이터 크기로 인한 전송 성능 저하의 요인으로 작용하고 있을 뿐만 아니라 실시간 영상의 처리가 어려워지게 된다.

따라서 본 논문에서는 영상 획득, 압축, 다중 전송의 부하(Load)들을 세 대의 시스템에 각각 나누어 처리하는 실시간 영상 다중 전송을 위한 병렬 서버 시

† 정 피 원: 한국통신 통신망연구소
†† 정 피 원: 경북대학교 컴퓨터 공학과
논문접수: 1996년 3월 12일, 심사완료: 1996년 10월 15일

시스템을 설계하고 이를 구현하였다. 고화질 영상을 획득하는 일과, 얻어진 영상을 압축하는 일, 그리고 원격지로 부터 실시간 영상 요구에 대한 영상을 다중 전송하는 일들을 각각 다른 시스템에서 비동기적으로 병렬 처리한다.

본 논문의 구성은 2장에서는 구현 환경을 다루었고 3장에서는 시스템의 개요에 대하여 기술하였으며 4장과 5장에서는 구현 내용과 실험 결과를 각각 나타내었고 마지막으로 6장에서 결론 및 향후 과제에 대해 요약하였다.

2. 구현 환경

본 연구는 먼저 시스템을 데스크별로 각각 분할하여 설계하였고 Windows 환경에서 수행되는 응용 프로그램으로 구현하였다. 시스템 개발에 사용된 도구는 MSVC(Microsoft Visual C++)와 VFWDK(Video for Windows Development Kit) 등이 있다. 영상 획득에는 Video Blaster SE 영상 보드를 이용하였으며 압축 시스템에서의 압축 유틸리티는 VFWDK에서 제공하고 있는 ICS(Installable Compressor Service)를 사용하였다[3, 4, 5]. 그리고 데이터 교환에 사용된 프로토콜은 10Mbps 전송률을 가지는 인터넷 LAN상에 TCP/IP와 본 연구실에서 개발한 Modified XTP를 사용하였다[6, 7]. 영상 획득과 압축 시스템은 Windows 3.1이 운영되고 있는 두 대의 486PC상에서 수행하고 다중 전송을 위해 Windows NT가 운영되고 있는 486PC를 사용하였다[8, 9, 10].

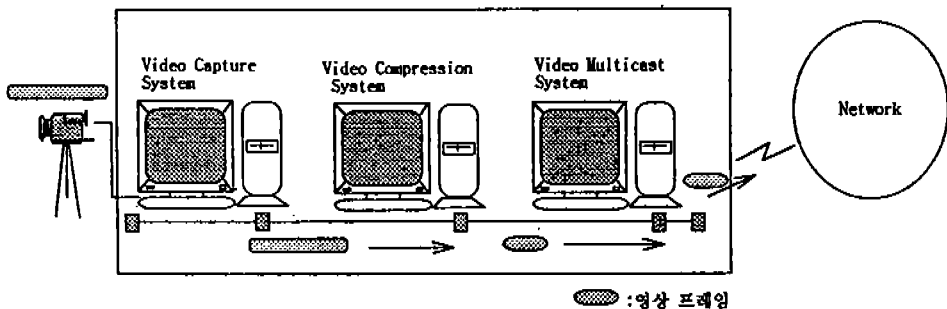
3. 시스템 개요

다중 전송 서버 시스템은 (그림 1)과 같이 영상 획득 시스템, 영상 압축 시스템 다중 전송 시스템 3개의 서브 시스템들이 차례로 연결된 파이프라인(Pipeline) 구조를 갖는다.

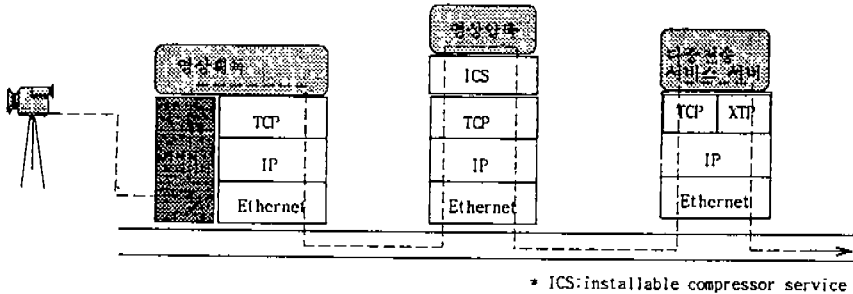
영상 획득 시스템(Video Capture System)은 영상 압축 기능을 가지고 있지 않은 이미지 획득 보드만으로 영상을 획득하고 Windows 환경에서의 이미지 포맷인 DIB(Device Independent Bitmap) 형태로 데이터를 변환시킨 후 영상 압축 시스템으로 프레임을 전달하는 기능을 담당한다[3, 4, 5]. 영상 압축 시스템(Video Compression System)은 영상 획득 시스템으로부터 DIB 형태의 프레임들을 수신한 후 이것을 S/W 버전의 Windows 압축 서비스를 이용하여 압축을 수행하여 압축된 DIB 프레임을 만든 다음 멀티캐스트 시스템으로 압축 프레임을 전송한다. 영상 다중 전송 시스템(Video Multicast System)은 원격지로부터 실시간 영상 요구에 대한 이미지 다중 전송을 담당하는데 영상 압축 시스템으로부터 압축된 프레임을 수신하고 영상 서비스에 대한 요구 데이터를 클라이언트로 멀티캐스트 한다.

데이터 교환에 사용된 프로토콜은 (그림 2)와 같이 인터넷 LAN상에 TCP/IP와 Modified XTP를 사용하였다.

영상 획득 시스템과 영상 압축 시스템간의 프레임 전달은 TCP/IP 프로토콜을 이용한 양방 채널로 이루어진다. 영상 압축 시스템과 다중 전송 시스템간의 데



(그림 1) 실시간 다중 전송 서버의 구조
(Fig. 1) Real time video multicasting server structure



(그림 2) 다중 전송 서버의 프로토콜 구조
(Fig. 2) Real time video multicasting server protocol structure

이더 전달 역시 TCP/IP 프로토콜을 이용한 양방 채널로 이루어진다. 다중 전송 시스템은 영상 압축 시스템으로부터 압축된 프레임은 TCP/IP 프로토콜을 이용하여 수신하지만 멀티캐스트 서비스는 본 연구실에서 개발한 Modified XTP 프로토콜을 이용하여 수행한다[7]. Modified XTP는 기능적 측면에서 기존의 트랜스포트 프로토콜 TCP, UDP, TP4의 프로토콜 기능 외에 전송률 제어 기능, 선택적 에러 제어 기능을 가지는 고속 수송 프로토콜로서 수송 계층에서 신뢰성 있는 멀티캐스트 기능을 제공한다는 장점이 있다.

4. 시스템 구현

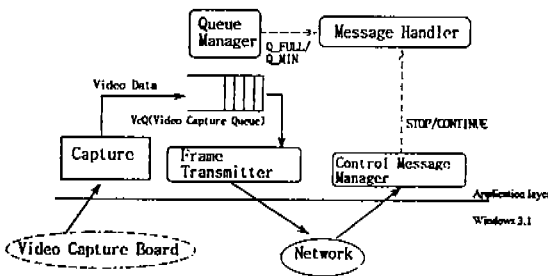
4.1 영상 획득 시스템

4.1.1 영상 획득 시스템 구조

영상 획득 시스템은 영상 획득 디바이스로부터 전달 받은 영상을 압축 시스템으로 전달하기 위한 시스템으로

로 (그림 3)과 같은 오브젝트 모듈로 구성되어 있다.

영상 획득 모듈(Video capture module)은 영상 보드로부터 획득한 이미지를 영상 획득 큐(Video Capture Queue; 이하 VcQ라 표기함)에 삽입한다. 프레임 전송 모듈(Frame transmission module)은 VcQ로부터 프레임 단위로 네트워크상으로 보내며 이 프레임은 압축 시스템으로 보내져 압축이 된다. 이때 큐관리모듈(Queue management module)은 VcQ의 상황을 감시하고, 네트워크 전송 속도가 영상 획득 속도보다 느려지게 되면 큐 자원의 과중한 사용을 방지할 수 있도록 제어 메시지(Q_FULL라 표기함)를 발생시킨다. 그리고 큐의 길이가 미리 정의된 최소 값(2프레임)이 되면 다시 제어 메시지(Q_MIN라 표기함)를 발생시켜 영상 획득 동작과 네트워크 전송 동작을 병행 처리한다. 영상 획득 동작과 네트워크 동작은 하드웨어로 수행되므로 큐를 사용하여 병행 처리할 수 있다. 그리고 흐름 제어를 위하여 압축 시스템으로부터 제어 메시지를 수신할 수도 있는데 이를 위하여 제어 관리 모듈(Control message management module)을 만들어 두었는데 이 모듈은 정지 메시지(IDM_STOP라 표기함)와 계속 메시지(IDM_CONTINUE라 표기함)의 두 가지 메시지를 가지고 있다. 정지 메시지가 수신되면 영상 획득 동작과 전송 동작을 멈추고 IDM_CONTINUE가 올 때까지 아무런 동작을 하지 않는다. IDM_CONTINUE가 압축 시스템으로부터 오면 영상 획득과 영상 데이터 수행을 계속하게 된다.



(그림 3) 영상 획득 시스템 구조
(Fig. 3) Video capture system structure

4.1.2 영상 획득 시스템 상태 흐름 및 동작

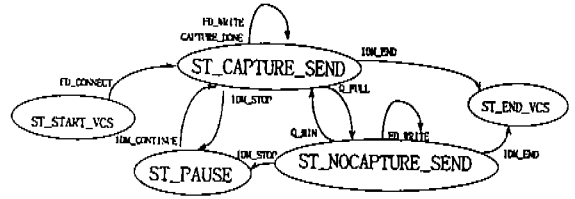
영상 획득 시스템은 ST START ST CAPTURE

SEND, ST_NOCAPTURE_SEND, ST_PAUSE, ST_END상태를 가진다. ST_START상태에서는 시스템 초기상태와 압축 시스템과 연결 설립을 하고 ST_CAPTURE_SEND상태로 천이한다.

ST_CAPTURE_SEND상태에서는 영상 획득과 데이터 전송을 비동기적으로 수행한다. VcQ가 모두 차면 영상 획득 동작을 정지시키고 ST_NOCAPTURE_SEND상태로 천이한다.

ST_NOCAPTURE_SEND상태에서는 영상 획득 동작은 이루어지지 않고 VcQ에 데이터를 프레임 단위로 네트워크상으로 송신하는 동작만 수행한다. VcQ의 상태가 2프레임 미만일 때에는 다시 ST_CAPTURE_SEND상태로 천이하며 이때 영상 압축 시스템으로부터 ST_STOP을 받으면 ST_PAUSE상태로 천이하고 데이터 동작과 획득 동작 모두 정지시킨다. 영상 압축 시스템으로부터 IDM_CONTINUE를 받으면 다시 ST_CAPTURE_SEND상태로 천이되고 압축과 영상 전송 동작이 계속 실행된다. ST_CAPTURE_SEND상

태 또는 ST_NOCAPTURE_SEND상태에서는 모두 사용자 메시지로 정의된 종료 메시지인 IDM_END를 받으면 ST_END상태로 천이하고 시스템을 종료시킨다. 상태 천이에 따른 자세한 동작을 (그림 4), <표 1>과 같이 요약하였다.



(그림 4) 영상 획득 시스템 상태천이도
(Fig. 4) Video capture system state transit diagram

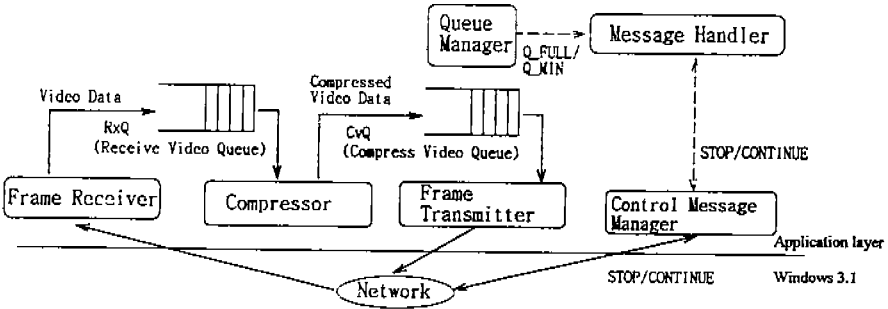
4.2 압축 시스템

4.2.1 압축 시스템 구조

영상 압축 시스템은 영상 획득 시스템으로부터 전

<표 1> 영상 획득 시스템 상태 천이 테이블
(Table 1) Video capture system state transit table

Current State	Event	Action	Next State
ST_START_VCS	FD_CONNECT	영상 획득 드라이버 실행, 영상 압축 시스템과 연결 설립	ST_CAPTURE_SEND
ST_CAPTURE_SEND	CAPTURE_DONE	영상 획득 드라이버로부터 획득한 영상을 큐에 삽입	ST_CAPTURE_SEND
	FD_WRITE	VcQ로부터 한 프레임을 삭제하고 영상 압축 시스템으로 전송	ST_CAPTURE_SEND
	Q_FULL	영상 획득 드라이버 정지시킴	ST_NOCAPTURE_SEND
	IDM_STOP	영상 획득 드라이버와 네트워크 동작 정지	ST_PAUSE
	IDM_END	영상 압축 시스템과 연결 해제	ST_END_VCS
ST_NOCAPTURE_SEND	FD_WRITE	VcQ로부터 한 프레임을 삭제하고 영상 압축 시스템으로 전송	ST_NOCAPTURE_SEND
	Q_MIN	영상 획득 드라이버 재동작 시킴	ST_CAPTURE_SEND
	IDM_STOP	영상 획득 드라이버와 네트워크 동작 모두 정지	ST_PAUSE
	IDM_END	영상 압축 시스템과 연결 해제	ST_END_VCS
ST_PAUSE	IDM_CONTINUE	영상 획득 드라이버 재동작/송신 네트워크 동작 계속	ST_CAPTURE_SEND
ST_END_VCS	none	영상 획득 드라이버 동작 정지, 큐의 모든 프레임을 삭제, 시스템 종료	ST_END_VCS



(그림 5) 영상 압축 시스템 구조
(Fig. 5) Video compression system structure

달받은 영상을 압축한 후 다시 다중 전송 시스템으로 전달하기 위한 시스템으로 (그림 5)와 같은 오브젝트 모듈로 구성되어 있다.

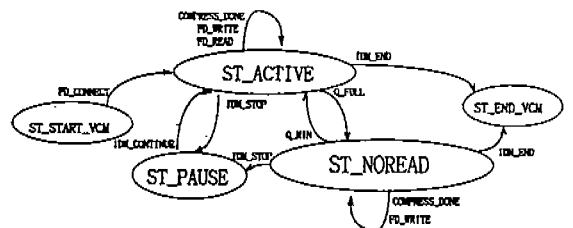
프레임 수신 모듈(Frame receive module)은 영상 획득 시스템으로부터 전달받은 프레임을 수신 데이터 큐(Receive video queue; 이하 RxQ라 표기함)에 삽입하며 이때 압축 모듈(Compression module)은 RxQ에 있는 데이터를 프레임 단위로 압축을 실시한다. 압축이 완료된 프레임은 압축 프레임 큐(Compress video queue; 이하 CvQ라 표기함)에 삽입되며 프레임 전송 모듈(Frame transmission module)은 CvQ의 프레임을 다중 전송 시스템으로 전송한다. 이 때 다중 전송 시스템과의 흐름 제어를 위하여 영상 획득 시스템에서와 마찬가지로 정지/계속 메시지를 사용하였는데 큐 관리모듈(Queue management module)은 CvQ를 감시하고, 정의된 최대 길이(15 프레임)를 넘으면 획득 시스템으로 정지 메시지를 보내며 압축 시스템 역시 CvQ 길이가 2프레임 이하로 줄게 되면 영상 획득 시스템에 IDM_CONTINUE를 보내어 영상 데이터를 전송받는다. 그리고 이 경우 다중 전송 시스템으로부터 흐름 제어를 위해 정지와 계속 메시지를 받을 수 있다. 정지 메시지를 받으면 계속 메시지를 받을 때까지 정지 상태로 됨과 동시에 영상 획득 시스템에도 역시 정지 메시지를 보내어 새로운 프레임 생성을 막는다. 정지 상태는 계속 메시지가 전달될 때까지 계속 유지된다.

4.2.2 압축 시스템 상태 흐름 및 동작

영상 압축 시스템은 ST_START, ST_ACTIVE, ST_NOREAD, ST_PAUSE, ST_END상태들을 가진다. ST_START상태에서는 시스템 초기화와 영상 획득 시스템 그리고 다중 전송 시스템과 연결을 설정을 하고 ST_ACTIVE상태로 천이한다.

ST_ACTIVE상태에서는 영상 획득 시스템으로부터 보내온 영상 데이터를 RxQ에 삽입하는 일과 이 큐로부터 프레임 단위로 데이터를 압축하여 CvQ에 삽입, 그리고 압축 데이터를 네트워크로 전송시키는 일을 각각 비동기적으로 수행한다. 큐 관리 모듈은 CvQ길이 상태만 감시하고, 큐가 제한된 자원을 모두 사용하였다면 ST_NOREAD상태로 천이한다.

ST_NOREAD상태에서는 영상 획득 시스템으로 IDM_STOP을 송신하여 압축 영상 프레임 전송은 중지시키고 영상 획득 시스템으로부터 이미 수신한 데이터에 대해서만 프레임 단위로 압축을 수행하고 RxQ가 비면 ST_PAUSE상태로 천이한다.



(그림 6) 영상 압축 시스템 상태천이도
(Fig. 6) Video compression system state transit diagram

〈표 2〉 영상 압축 시스템 상태 천이 테이블
 〈Table 2〉 Video compression system state transit table

Current State	Event	Action	Next State
ST_START_VCM	FD_CONNECT	영상 획득 시스템, 영상 압축 시스템과 연결 설립	ST_ACTIVE
ST_ACTIVE	COMPRESS_DONE	압축된 프레임을 CvQ에 삽입 후, RxQ로부터 한 프레임을 삭제하고 압축 시작	ST_ACTIVE
	Q_FULL	영상 획득 시스템으로 정지 메시지(IDM_STOP) 전송, 수신 네트워크 동작 정지	ST_NOREAD
	FD_WRITE	CvQ로부터 한 프레임을 삭제하고 다중 전송 시스템으로 전송	ST_ACTIVE
	FD_READ	영상 획득 시스템으로부터 데이터를 수신하고 프레임을 만들고, 완전한 프레임이 구성되면 RxQ에 삽입 한다. 압축 중인 프레임이 없으면 RxQ로부터 한 프레임을 삭제하고 압축 시작	ST_ACTIVE
	IDM_STOP	영상 획득 시스템으로 정지 제어 메시지를 송신, 네트워크 송수신 동작 모두 정지	ST_PAUSE
	IDM_END	영상 획득 시스템, 다중전송시스템과 연결 해제	ST_END_VCM
ST_PAUSE	IDM_CONTINUE	압축 드라이버 재 동작/송수신 네트워크 동작 계속, 영상 획득 시스템으로 계속(IDM_CONTINUE) 메시지 전송	ST_ACTIVE
ST_END_VCM	none	영상 압축 드라이버 동작 정지, 큐의 모든 프레임을 삭제, 시스템 종료	ST_END_VCM

ST_PAUSE상태는 압축 동작과 데이터 송신 동작 모두 중지된 상태이다. 다중 전송 시스템으로부터 IDM_CONTINUE가 올 때까지 현재 상태를 계속 유지한다. IDM_CONTINUE가 오면 다시 ST_ACTIVE상태로 천이하고 위 동작을 반복하게 된다. ST_ACTIVE, ST_NOREAD 두 상태에서는 모두 사용자 메시지로 정의된 IDM_END를 받으면 ST_END상태로 천이하고 시스템 동작을 종료한다. 상태 천이에 따른 자세한 동작을 (그림 5), 〈표 2〉와 같이 요약하였다.

4.3 영상 다중 전송 시스템

4.3.1 영상 다중 전송 시스템 구조

영상 다중 전송 시스템은 영상 압축 시스템으로부터 전달받은 영상을 클라이언트 시스템으로 전달하기 위한 시스템으로 (그림 7)과 같은 오브젝트 모듈로 구성되어 있다.

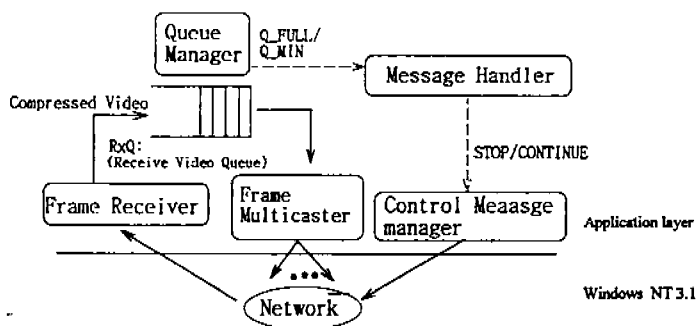
영상 수신 모듈(Frame receive module)은 압축 시스템으로부터 전달받은 데이터를 수신 큐(RxQ)에 삽입

하며 큐 관리 모듈은 흐름 제어를 위하여 큐의 길이를 감시하고, 최대 큐 길이(15 프레임)를 넘으면 압축 시스템으로 IDM_STOP을 보낸다. 그리고 RxQ 길이가 5프레임 이상 남을 때까지 클라이언트 시스템으로 데이터를 전송하고 RxQ가 줄게 되면 압축 시스템에 IDM_CONTINUE를 보내어 압축 영상을 전송받는다. 다중 전송 모듈(Frame multicast module)은 RxQ로부터 프레임 단위로 압축된 영상을 Modified XTP 프로토콜의 다중 전송기능을 이용하여 클라이언트 시스템으로 전달한다.

4.3.2 영상 다중 전송 시스템 상태 흐름 및 동작

영상 다중 전송 시스템은 ST_START, ST_ACTIVE, ST_MULTICAST_ONLY, ST_PAUSE, ST_END상태들을 가진다. ST_START상태에서 시스템 초기화와 영상 압축 시스템 그리고 지정된 클라이언트 시스템들과 연결 설립을 하고 ST_ACTIVE상태로 천이한다.

ST_ACTIVE상태에서는 영상 압축 시스템으로부터



(그림 7) 영상 다중 전송 시스템 구조
(Fig. 7) Video multicast system structure

압축 영상 데이터를 전송 받아 RxQ에 삽입하는 일과 이 큐로부터 프레임 단위로 영상 데이터를 각 클라이언트로 전송시키는 일을 각각 비동기적으로 수행한다. 큐 관리 모듈은 RxQ의 길이를 감시하고 큐가 정의된 최대 프레임 수에 이르면 ST_MULTICAST_ONLY 상태로 천이한다.

ST_MULTICAST_ONLY상태에서는 영상 압축 시스템으로 IDM_STOP을 송신하고 영상 압축 시스템으로부터 수신한 압축 영상 데이터를 프레임 단위로 클라이언트들로 각각 전송을 수행하고 RxQ가 정의된 최소 큐 길이로 줄면 다시 ST_ACTIVE 상태로 천이한다.

ST_PAUSE상태에서는 사용자 인터페이스로부터 메시지를 수신하게 되는데, 이때 다중 전송 시스템은 압축 시스템으로 IDM_STOP을 송신한다. 이때 압축 데이터 수신 동작 모듈과 송신 동작 모듈 모두 정지한 상태이다. 사용자 인터페이스로부터 IDM_CONTINUE가 올 때까지 이 상태를 유지한다. IDM_CONTINUE

가 오면 다시 ST_ACTIVE상태로 천이하고 위 동작을 반복하게 된다.

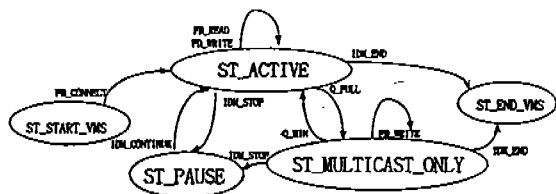
ST_ACTIVE, ST_MULTICAST_ONLY 두 상태에서는 모두 사용자 메시지로 정의된 IDM_END 메시지를 받으면 ST_END상태로 천이하고 시스템 동작을 종료한다. 상태 천이에 따른 자세한 동작을 (그림 8), <표 3>과 같이 요약하였다.

5. 시스템 성능 테스트

시스템의 성능 평가를 위하여 각 서버 시스템의 처리 지연을 측정하고 다중 전송 클라이언트 시스템에서의 초당 영상 프레임 수를 측정하였다.

영상 획득 보드는 압축 기능이 없는 Video Blaster SE를 사용하고 비디오 크기를 80×60에서 320×240까지 변화시켜 영상을 획득하는데 드는 지연 속도와 초당 획득할 수 있는 프레임 수 (표 4)를 측정하였다. 이 결과는 비디오의 크기가 커질수록 영상획득에 드는 시간이 많이 걸림을 보여주고 있다.

영상 압축 시스템의 압축 유틸리티는 VFWDK(Video for Windows Develop Kit)에서 제공하는 라이브러리를 사용하였다. 그리고 압축 프로그램을 선택적으로 사용할 수 있도록 구현하였으며 각 압축 프로그램을 선택할 때 압축율을 조정할 수 있도록 하였다. 본 논문에서 사용된 압축 프로그램은 Microsoft Video 1과 Intel Indeo Video이다. 그리고 Microsoft Video 1은 요구 화질 100%, 75%, 50%일 때 각각 성능을 측정하였으며 Intel Indeo Video 경우는 요구 화질



(그림 8) 영상 다중 전송 시스템 상태천이도
(Fig. 8) Video multicast system state transit diagram

〈표 3〉 영상 다중 전송 시스템 상태 천이 테이블
 〈Table 3〉 Video multicast system state transit table

Current State	Event	Action	Next State
ST_START_VMS	FD_CONNECT	영상 압축 시스템, 멀티캐스트 클라이언트 시스템과 연결 설립	ST_ACTIVE
ST_ACTIVE	FD_READ	영상 압축 시스템으로부터 데이터를 수신하고 완전한 프레임을 만들고, 완전한 프레임이 구성되면 RxQ에 삽입	ST_ACTIVE
	FD_WRITE	RxQ로부터 한 프레임을 삭제하고 연결 중인 클라이언트 시스템으로 프레임 전송	ST_ACTIVE
	Q_FULL	영상 압축 시스템으로 정지 제어(IDM_STOP) 메시지를 송신하고 수신 네트워크 동작 중지시킴	ST_MULTICAST_ONLY
	IDM_STOP	영상 압축 시스템으로 정지 제어(IDM_STOP) 메시지를 송신, 네트워크 송수신 동작 모두 정지 시킴	ST_PAUSE
	IDM_END	영상 압축 시스템, 클라이언트 시스템들과 연결 해제	ST_END_VMS
ST_MULTICAST_ONLY	FD_WRITE	RxQ로부터 한 프레임을 삭제하고 연결 중인 클라이언트 시스템으로 프레임 전송	ST_MULTICAST_ONLY
	Q_MIN	영상 압축 시스템으로 정지 제어(IDM_STOP) 메시지를 송신, 네트워크 수신 재동작시킴	ST_ACTIVE
	IDM_STOP	영상 압축 시스템으로 정지 제어 메시지를 송신, 네트워크 송수신 동작 모두 정지	ST_PAUSE
	IDM_END	영상 압축 시스템, 클라이언트 시스템과 연결 해제	ST_END_VMS
ST_PAUSE	IDM_CONTINUE	영상 압축 시스템으로 계속(IDM_CONTINUE) 메시지 전송, 송수신 네트워크 동작 계속	ST_ACTIVE
ST_END_VMS	none	큐의 모든 프레임을 삭제, 시스템 종료	ST_END_VMS

〈표 4〉 테스트에 사용된 영상 획득 보드의 성능
 〈Table 4〉 Performance of video capture board

측정 내용	비디오 크기	80×60	120×90	160×120	200×150	320×240
영상 획득 지연(sec)		0.160	0.163	0.193	0.30	0.31
영상 획득율(Frames/sec)		6.25	6.13	5.18	3.33	3.18

〈표 5〉 영상 압축 지연(단위 : μs)
 〈Table 5〉 Video compression delay(μs)

비디오 크기 압축 프로그램 종류	비디오 크기				
	80×60	120×90	160×120	200×150	320×240
MS Video 100%	533	4348	8360	8000	13089
MS Video 75%	267	2989	6752	6500	12565
MS Video 50%	267	543	5788	5000	12042
Indeo 100%	533	2446	7395	7000	13089
Indeo 75%	533	2989	7717	4500	13089
Indeo 50%	533	2989	7717	4500	13089
Indeo 30%	533	4076	8360	8000	13089
Indeo 10%	533	3804	8360	8000	13613

〈표 6〉 압축 프로그램 압축율(%)
 〈Table 6〉 Video compression ratio(%)

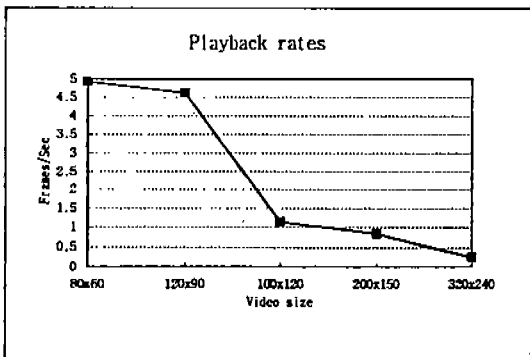
비디오 크기 압축 프로그램 종류	비디오 크기				
	80×60	120×90	160×120	200×150	320×240
MS Video 100%	40	41	39	40	42
MS Video 75%	59	64	59	65	68
MS Video 50%	80	83	82	84	85
Indeo 100%	66	70	63	69	69
Indeo 75%	66	70	63	69	69
Indeo 50%	71	75	68	73	75
Indeo 30%	75	78	73	77	78
Indeo 10%	75	78	73	77	78

100%, 75%, 50%, 30%, 10%로 나누어 측정하였다. Microsoft Video 1은 압축율이 Intel Indeo Video보다 우수하지만 요구 화질이 50%이하일 경우는 영상의 손실이 크다. <표 5>는 측정된 영상의 크기 변화에 따른 압축 지연을 나타내었고 <표 6>은 압축 프로그램의 종류와 요구 화질의 변화에 따른 압축을 측정 결과를 나타내었다.

압축 지연은 영상 획득에 드는 시간에 비하여 작은 것을 보여준다. 그리고 압축알고리즘의 종류보다는 비디오의 크기 및 요구 화질의 수준에 따라 시간 지연의 변화가 큰것을 보여주고 있다. 결국 전체시스템 중에서 압축 시스템은 병목점으로는 작용하고 있지 않다.

클라이언트 시스템에서 영상 재생은 영상을 받은 후 수신 영상을 큐에 대기시켰다가 프레임간의 지연 시간을 일정하게 적용하여 플레이 백(Play back)시켰다. (그림 9)에서 영상 크기에 따른 프레임 재생율을 나타내었다. 그리고 이 결과에 적용된 압축 프로그램은 Microsoft Video 1 화질 50%일 때의 결과이다.

비디오의 크기가 80×60 및 120×90에서는 영상 손실이 거의 없이 압축을 수행하여 다중 전송을 하고 있음을 보여주고 있다. 하지만 크기가 160×120이상의 비디오에서는 현저히 줄어듦을 보여주고 있는데, 이는 영상 획득 시스템에서 압축 시스템으로 전송시 비디오 크기의 증가로 네트워크 병목현상이 발생했기 때문이다. 이를 해결하기 위해서는 각 서버 시스템을 보다 빠른 네트워크로 연결함으로써 해결할 수 있을 것이다.



(그림 9) 다중 전송 클라이언트에서의 재생율 (Fig. 9) Playback rates on a video multicast client system

6. 결 론

본 논문에서는 네트워크 상의 영상 다중 전송을 위한 분산 병렬 처리 서버를 구현하였고, 이를 위하여 서버 시스템간의 효율적인 데이터 전송과 비동기적 처리를 위한 메커니즘을 제시하였다. 본 연구에서 구현된 결과는 저 비용으로 원격지 영상 강의, VOD(Video on demand)와 같은 응용 프로그램에 필요한 실시간 영상을 다중 전송하는데 활용할 수 있을 것이다. 향후 고속망 환경에서의 테스트 및 음성, 문자 등과 같은 다양한 미디어와 함께 사용할 수 있는 완전한 멀티미디어 서버 시스템으로 발전시켜 나아갈 것이다.

참 고 문 헌

- [1] A. N. Netravali, W.D. Roomf, K. Sabnani, "Design and Implementation of a High Speed Transport Protocol," *IEEE Trans. On Communication*, Vol. 38, No. 11, Nov. 1990.
- [2] Christos Papadopoulos, Gurudatta M. Parukar, "Experimental Evaluation of SUNOS IPC and TCP/IP Protocol Implementation," *IEEE/ACM transaction on networking*, Vol. 1, No. 2, April, 1993.
- [3] Arthur Dumas, "Programming WinSock," SAMS Publishing, SAMS Publishing, 1995.
- [4] "Multimedia Programmer's Guide, Microsoft Windows Software Development Kit," Microsoft Corporation, 1992.
- [5] "Multimedia Programmer's Refrence, Microsoft Windows Software Development Kit," Microsoft Corporation, 1992.
- [6] Douglas E. Comer & David L. Stevens, "Internetworking with TCP/IP Volume I, II," Prentice Hall, 1991.
- [7] "XTP protocol definition revision 3.6," Protocol Engines Incorporated, Nov. 1991.
- [8] Helen Custer, "Inside WINDOWSNT," Microsoft press, 1993.
- [9] Ralph Davis, "Windows NT Network Programming," Addison Wesley Publishing, 1994.

[10] Loeb. S, "Delivering Interactive Multimedia Documents over Networks," *IEEE Communications Magazine*, Vol. 30, No. 5, 1992.



김 철 우

1994년 영남대학교 전산공학과 졸업(학사)
1996년 경북대학교 컴퓨터공학과(석사)
1996년~현재 한국통신 통신망연구소 전임연구원
관심분야: 분산처리, 전화망 트래픽 분석 등



이 선 우

1995년 경북대학교 컴퓨터공학과 졸업(학사)
1995년~현재 경북대학교 컴퓨터공학과 석사과정
관심분야: 고속 통신 프로토콜, ATM, 멀티미디어 통신 등

한 기 준

1979년 서울대학교 전기공학과 졸업(학사)
1981년 한국과학기술원 전기공학과 졸업(석사)
1985년 University of Arizona 전기 및 전산공학과 졸업(석사)
1987년 University of Arizona 전기 및 전산공학과 졸업(박사)
1981년~1984년 국방과학연구소 연구원
1988년~현재 경북대학교 컴퓨터공학과 부교수
관심분야: 전산망 프로토콜, B-ISDN, MAN/LAN, 분산처리



이 완 직

1992년 경북대학교 통계학과 졸업(학사)
1994년 경북대학교 컴퓨터공학과(석사)
1994년~현재 경북대학교 컴퓨터공학과 박사과정
관심분야: 고속 통신 프로토콜, ATM, 멀티미디어 통신 등



이 경 호

1995년 경북대학교 컴퓨터공학과 졸업(학사)
1995년~현재 경북대학교 컴퓨터공학과 석사과정
관심분야: 고속 통신 프로토콜, ATM, PCS, 멀티미디어 통신 등