

# 인터넷상에서 실시간과 비 실시간의 동시 원격 모니터링을 위한 Directshow 기반의 다채널 감시 시스템 구조 연구

박용대<sup>o</sup>, 노병희

아주대학교 정보통신 전문대학원  
{neverdai, bhroh}@ajou.ac.kr

## Directshow-based multi-channel surveillance system architecture for simultaneous real time and non-real time remote monitoring on the internet

Yong-dae Park<sup>o</sup>, Byeong-hee Roh

Ajou university, Graduate school of Information and Communication

### 요 약

영상의 전송에 있어서 단일 채널인 경우 지금까지 많은 연구가 진행되어왔다. 그러나 다채널의 영상을 전송할 경우에, 여러 채널을 하나로 묶어서 전송하게 되는데, 이 방법에 대한 논의는 그리 활발하지 못하였다. 우리는 이 연구를 통해서 다채널의 영상을 하나로 묶어서 전송하는 시스템과 수신하는 시스템을 설계하고 구현하였다. 이번 연구의 최종 목표는 H.263인코더<sup>[1][2]</sup>를 사용하고 마이크로소프트 다이렉트쇼(Directshow<sup>TM</sup>) 기술을 이용하여 4 채널의 감시카메라 시스템을 구현하는 것이다.

### 1. 서 론

local 건물 내 또는 네트워크를 통한 원격 감시 시스템들의 개발은 많이 진행되어 왔다. 감시 카메라 시스템은 주로 실시간의 영상을 보여주거나, 비 실시간의 영상, 즉 과거에 녹화된 영상을 동시에 보여줄 수 있어야 한다. 따라서 카메라로 캡처된 영상은 원격지로 전송이 되는 동시에 저장 장치에 저장되는 형태를 가진다. 그런데 기존의 원격 감시 시스템들은 주로 JPEG, wavelet과 같은 정지 영상 기법을 주로 사용하기 때문에 이러한 시스템들은 아주 큰 용량의 저장장치가 필요할 뿐만 아니라, 네트워크의 대역폭도 많이 잡아먹는 문제점을 안고 있다. 또한 만약 이러한 시스템을 인터넷에서 이용할 경우, 인터넷에서의 대역폭이라는 자원은 시시각각 변화하기 때문에, 네트워크에 적응적인 방법이 필요하지만, 기존의 시스템들은 네트워크에 적응적인 방법을 사용하지 않는 단점을 가지고 있다.

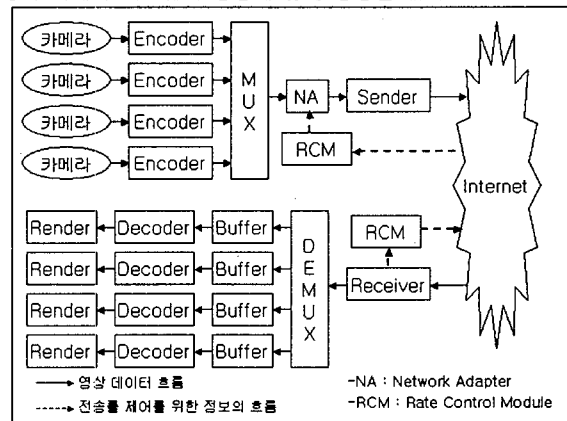
이 논문에서는 인터넷을 통해 실시간의 원격 모니터링과 비 실시간의 원격 검색이 가능한 다채널 감시 시스템에 대한 구조를 제안할 것이다. 우리는 프로그램의 구현을 위해서 마이크로소프트사의 Directshow를 사용했고, 전지 영상 기법의 단점을 보완하기 위해서, ITU-T의 H.263 인코더를 이용했다. 우리는 또한 실시간 모니터링을 위해 네트워크의 상황에 따라 전송률을 조절하는 방법을 적용할 것이다.

논문의 구성은 다음과 같다. 시스템 기본 구조에서는 캡처 시스템과 모니터링 시스템의 구조를 설명하고, Directshow 기반의 시스템 구조에서는 directshow를 이용하여 어떻게 프로그램을 구성하였는지를 설명한다. 통신 프로토콜에서는 캡처 시스템과 모니터링 시스템의 통신 프로토콜과 데이터 PDU에 대해 이야기 하고, 시스템의 구현 결과에서는 시나리오를 따른 프로그램의 동작과 결과, 그리고 개선 사항에 대해서 이야기를 한다.

### 2. 감시 카메라 시스템 구조

#### 2.1 시스템 기본 구조

감시 카메라 시스템은 크게, 영상을 카메라로 캡처하는 부분과, 캡처된 영상을 모니터링하는 두 부분으로 나누어진다. 전자는 캡처 시스템이고 후자는 모니터링 시스템인데, 캡처 시스템은 여러 카메라에서 캡처된 영상을 받아서 가공하여 저장하고, 모니터링 시스템은 사용자에게 캡처된 데이터를 보여주는 기능을 한다. 이 때 모니터링을 하게 되는 영상은 크게 두 종류로 구분이 되는데, 실시간 영상과 비 실시간 영상, 즉 과거에 캡처되어 저장된 영상이다. 감시 카메라 시스템은 이 두 종류의 영상을 보여줄 수 있어야 한다. 만약 감시 시스템이 다채널일 경우, Muxing/Demuxing의 방법이 감시 카메라 시스템의 성능에 많은 영향을 준다.



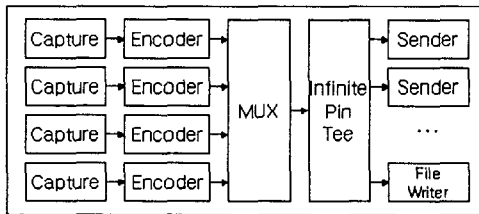
<그림 1> 감시 시스템 기본 구조

2.2 Directshow<sup>[3]</sup> 기반의 시스템 구조

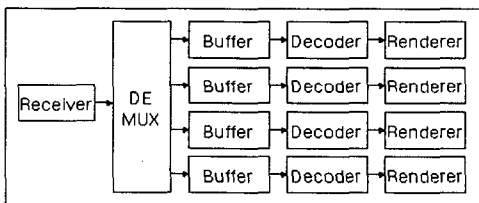
2.2.1 Directshow

마이크로소프트 다이렉트 쇼(Directshow<sup>TM</sup>)는 윈도우즈 시스템에서 멀티미디어의 효율적인 제어를 위해 사용되는 DirectX의 하부 기술이다. 멀티미디어의 처리를 위한 하나의 프로그램은 각각의 기능을 하는 필터들로 구성이 된다. 필터의 종류는 크게 세 가지인데, 소스필터와 변환필터, 그리고 랜더필터로 나눌 수 있다. 소스필터는 영상이나 음성을 캡처하는 필터나 파일에 저장된 데이터를 읽어오는 필터이고, 변환필터는 입력으로 들어온 데이터를 가공하여 다른 형태의 데이터로 바꾸어 저장한다. 마지막으로 랜더필터는 입력으로 들어온 데이터를 랜더링, 저장하거나, 네트워크로 전송을 하는 필터를 말한다. 멀티미디어 데이터는 이러한 필터들을 거치면서 새롭게 가공되어서, 요구되는 결과물로 변경되어 처리되는 형태를 가진다.

우리는 감시카메라 시스템의 설계를 위해서 Directshow 기술을 사용하였다. 그림 2,3은 Directshow를 이용한 감시카메라 시스템의 구조를 나타낸다. 아래 그림에서 각각의 네모 상자는 Directshow를 이용해서 만들어진 필터를 나타낸다. 그림 1에서의 Infinite Pin Tee 필터는 한 입력 핀으로 들어온 데이터를 그대로 여러 개의 출력 핀으로 전달해주는 기능을 가지는 필터로, 여러 모니터링 시스템에서 요청이 들어올 경우 여러 개의 sender가 있어야 하기 때문에 사용되었다.



<그림 2> 캡처 시스템 필터



<그림 3> 모니터링 시스템 필터

2.2.2 어플리케이션

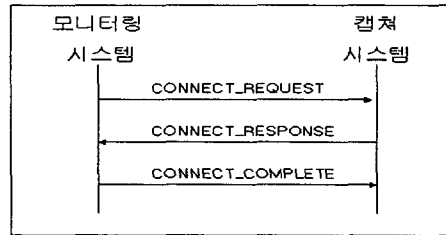
어플리케이션은 사용자의 입력을 받아서, 사용자가 원하는 기능을 실행시켜준다. 어플리케이션이 수행하는 작업은 크게 사용자의 입력을 받아들이는 기능, 필터를 생성, 수정하는 작업, 그리고 캡처 시스템과 모니터링 시스템의 통신 기능 등으로 나뉘볼 수 있다. 사용자의 입력을 받아들이는 기능은 사용자가 원하는 기능을 눌렀을 경우, 이에 적절한 기능을 수행하는 과정이 된다. 캡처 시스템은 서버 시작/서버 종료와 같은 사용자의 요청이 있을 수 있고, 모니터링 시스템은 실시간 데이터의 모니터링/이 전 데이터의 모니터링 등과 같은 요청이 있을 수 있다. 필터의 생성, 수정 기능은 사용자의 요청을 수행하기 위해 그림 2이나 그림 3과 같은 필터를 생성하거나 수정하는 기능을 나타낸다. 그리고 캡처 시스템과 모니터링 시스템간의 통신 기능은 모니터링 시

스템이 캡처 시스템에게 원하는 기능을 요청하기 위한 기능이다.

2.3 통신 프로토콜

모니터링 시스템이 캡처 시스템에게 원하는 기능을 요청하기 위해서는 양 시스템간의 통신을 위한 프로토콜이 필요하다. 통신 프로토콜은 크게 실시간 데이터의 요청과, 비 실시간 데이터의 요청으로 나뉘볼 수 있다.

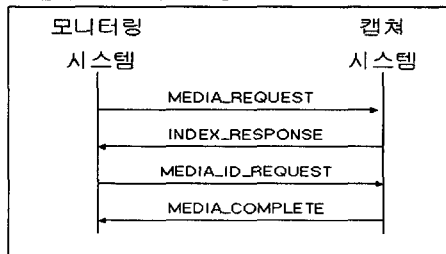
2.3.1 실시간 데이터의 요청



<그림 4> 실시간 데이터 요청 프로토콜

모니터링 시스템에서 사용자가 실시간 데이터를 보기 원하면 모니터링 시스템은 CONNECT\_REQUEST 메시지를 캡처 시스템으로 보낸다. 캡처 시스템이 이 메시지를 받으면, CONNECT\_REQUEST 메시지를 모니터링 시스템에 돌려 보낸다. 마지막으로 CONNECT\_COMPLETE 메시지를 모니터링 시스템이 캡처 시스템에 보내면 이 때부터 캡처 시스템은 모니터링 시스템으로 실시간 데이터를 전송하게 된다. 감시 카메라의 보안에 대한 특성 상, 만약 인증 기능이 필요하다면 CONNECT\_REQUEST에 아이디와 암호를 실어서 보내고, CONNECT\_REQUEST 메시지를 통해 인증을 확인하는 절차를 추가할 수 있다.

2.3.2 비 실시간 데이터의 요청



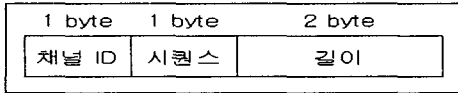
<그림 5> 비 실시간 데이터 요청 프로토콜

사용자로부터 비 실시간 데이터, 즉 이 전 데이터의 요청이 있으면, 모니터링 시스템은 MEDIA\_REQUEST 메시지를 캡처 시스템에 전송한다. 이 메시지를 수신한 캡처 시스템은 INDEX\_RESPONSE 메시지에 자신이 가지고 있는 이 전 데이터의 목록을 실어 보내게 되고, 모니터링 시스템은 이 중 하나를 선택해서 MEDIA\_ID\_REQUESTS 메시지에 실어 캡처 시스템에 전송한다. 캡처 시스템은 MEDIA\_COMPLETE 메시지를 모니터링 시스템에 전송한 후, 비 실시간 데이터를 캡처 시스템으로 전송한다.

2.3.3 다중화

다채널의 소스를 하나의 채널로 묶어 전송하기 위해서 다중화 작업을 거치는데 어떻게 다중화를 하느냐에 따라

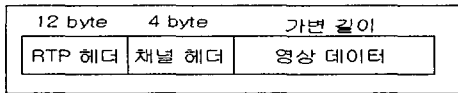
시스템의 성능에 영향을 줄 수 있다. 그러나 이번 연구에서는 비교적 간단한 방법으로 구현하였다. 즉, FIFO의 방법으로, 여러 소스 채널 중에서 데이터가 가장 먼저 들어온 순서대로 배열한 후 한 채널로 만드는 방법을 사용하였다. 그리고 수신측에서는 한 채널로 들어오는 데이터를 받아서 원래의 여러 채널로 되돌려야 하기 때문에, 다중화를 할 때 각각의 데이터에 채널에 대한 정보를 가지는 헤더를 붙여야 한다. 그림 6은 채널의 정보를 갖고 있는 헤더를 나타낸다. 채널 ID는 데이터의 소스 채널이 어떤 것인지 나타내고, 시퀀스는 소실된 데이터의 검출을 위해서 사용된다. 또한 길이는 영상 데이터의 길이를 나타낸다.



<그림 6> 채널 헤더

2.3.4 영상 데이터의 전송 PDU

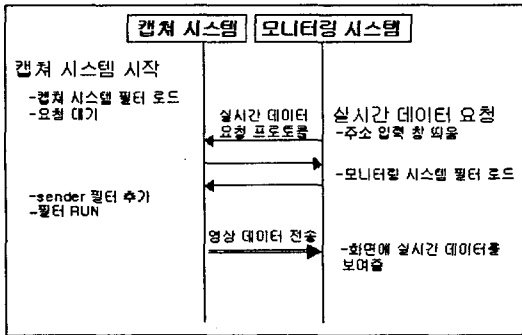
영상 데이터의 전송은 UDP기반의 RTP/RTCP 프로토콜이 사용되었다. RTP 헤더에 들어가는 정보는 4바이트의 SSRC, 4바이트의 Timestamp, 2바이트의 sequence 등이 된다. 그런데 운영체제 윈도우 시스템에서 사용하는 시간 정보는 100 나노 초까지 표현할 수 있는 8바이트를 사용한다. 따라서 8바이트의 시간 정보를 4바이트의 시간 정보로 변경하는 작업을 수행해야 한다. 그림 7은 각 한 영상 데이터를 보내기 위해 붙여지는 헤더와 영상 데이터를 나타낸다.



<그림 7> 영상 데이터 전송 PDU

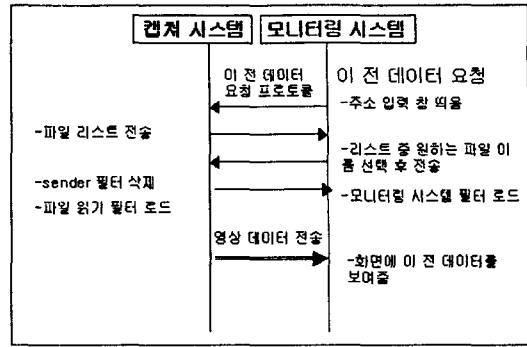
3. 시스템 구현 결과

캡처 시스템이 동작 중이고 하나의 모니터링 시스템이 캡처 시스템에게 실시간 데이터를 요청한 다음, 일정 시간이 지난 후에 이 전 데이터를 요청할 경우에 캡처 시스템과 실시간 시스템은 어떠한 변화가 일어나는지 알아보면 그림 8, 9와 같다.



<그림 8> 시나리오(1/2)

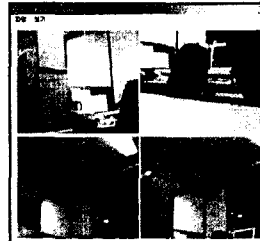
모니터링 시스템에서 캡처 시스템에게 실시간 데이터의 요청을 하면, 그림 3과 같은 메시지를 주고 받는다. 이 과정에서 캡처 시스템이나 모니터링 시스템은 각각 필요한 필터들을 로드한 후, 캡처 시스템에서 모니터링 시스템에 영상 데이터를 전송하기 시작하고, 모니터링 시스템은 이 영상 데이터를 수신하여 화면에 보여준다.



<그림 9> 시나리오(2/2)

모니터링 시스템이 실시간 영상을 보여주는 상황에서 비 실시간 영상을 요청하면, 캡처 시스템은 소유하고 있는 영상 리스트를 전송하고, 모니터링 시스템은 이 중에서 보기를 원하는 영상을 선택해서 캡처 시스템에 전송한다. 캡처 시스템은 실시간 데이터를 전송하던 sender 필터를 제거하고, 파일 읽기 필터를 로드한 다음, 비 실시간 영상 데이터를 전송한다. 모니터링 시스템은 이 영상 데이터를 수신하여 화면에 보여주게 된다.

그림 10과 그림 11은 캡처시스템과 모니터링 시스템에서 화면에 뿌려지는 실행 결과이다.



<그림 10> 캡처 시스템



<그림 11> 모니터링 시스템

4. 결론

본 연구를 통해서 다채널의 경우의 감시카메라 시스템을 구현하였다. 그러나 이 시스템은 아직 네트워크의 상황에 맞게 전송률을 조절하는 방법을 적용시키지 않았다. 전송률의 조절은 캡처 시스템과 모니터링 시스템에서 RCM(Rate control module)을 이용해서 네트워크의 상황에 대한 정보를 수집한 다음, 이를 바탕으로 NA(Network adapter)에게 전송률을 명시하는 방법이 있다. 이 때 NA에서의 전송률 조절은 인코딩된 데이터의 Motion Vector를 변경하는 기법<sup>[4]</sup>을 사용해서, 프레임을 drop시키는 방법이 효율적이다. 앞으로의 연구에서는 RCM과 NA 모듈을 제작한 후, 이 시스템에 올려서 네트워크의 상황에 맞게 전송률을 조절하는 기능을 추가하고 성능을 분석할 계획이다.

참조

[1] Recommendation H.263, "Video coding for low bit rate communication", ITU-T  
 [2] 김재균, "영상 통신 시스템", 영지 문화사  
 [3] Microsoft DirectX 8.1 SDK Documentation  
 [4] J. Youn, M. Sun, and C. Lin, "Motion Vector Refinement for High-Performance Transcoding," IEEE Trans. Multimedia, vol. 1, no. 1, pp. 30-40, March