

이동형 운영체제를 위한 적응적 스트림 처리기 설계 및 구현

박성근*, 오미경**, 임영환***

송실대학교 컴퓨터과

e-mail:pmystory@media.ssu.ac.kr

A Design and Implementation of an Adaptive Stream Controller for Mobile Operating System

Seung-Keun Park*, Mi-Kyong Oh**, Young-Hwan Lim***

*Dept of Computing, Soong-Sil University

요약

본 논문에서는 이동 단말기 중 하나인 Windows CE 기반 PDA를 위한 적응적 멀티미디어 스트림 처리기를 설계 및 구현하였다. 제한된 시스템 환경을 갖고 있는 PDA 상에서 멀티미디어 데이터를 실시간으로 전송 및 저장할 때 생기는 메모리 문제를 효율적으로 해결하기 위하여 적응적 버퍼 관리 기법을 사용하였다. 단말기에 적합한 형태로 멀티미디어 데이터를 변환하여 전송하기 위하여 필터를 사용하였고, 필터 파이프라인을 구성하였다. 또한 다양한 형태의 미디어와 필터의 입출력을 일관성 있게 처리하기 위해서 DLM(Dynamic Linking Module) 구조를 사용하였다.

1. 서론

최근 인터넷 인구가 급속하게 증가하고, 초고속 정보통신망이 빠르게 확산됨에 따라서 인터넷을 통하여 영화를 감상하고, TV를 시청하고, 음악을 듣는 일들이 일상 생활이 되고 있다.

스트리밍(Streaming) 기술은 실시간으로 동영상이나 오디오를 보고 듣는데 있어서 필수적인 기술로서, 대용량의 멀티미디어 데이터를 다운로드 받지 않고, 파일을 다운로드 하면서 동시에 재생해 준다. 이 기술은 멀티미디어 데이터의 실시간 전송도 가능하도록 한다.

최근 PDA나 Handheld PC 등 포스트 PC(Post PC)의 수요가 늘어나고 있다. 이러한 단말기들은 이동이 간편하기 때문에, 무선 통신망과 연결한다면 이동 중에도 인터넷 등 여러 가지 기능을 사용할 수 있다. 그리고 이러한 단말기들은 Desktop PC에 비해서 메모리가 부족하고, 하드디스크도 없는 등 컴퓨팅 환경이 제한적이기 때문에, 대용량의 멀티미디어 데이터를 저장해서 가지고 다니기가 곤란하다. 따라서 이동 단말기 상에서 멀티미디어 스트리밍 서비스가 요구된다.

본 논문에서는 이동형 운영체제를 위한 스트림(Stream) 처리기를 제안하고 Windows CE 기반의 PDA 상에서 실험하고자 한다.

2. 이동형 운영 체제를 위한 스트림 처리기 문제점

- 시스템 환경

Windows CE 기반의 PDA는 일반 PC(Personal Computer)에 비해 성능(CPU 처리속도, 메모리 등)이 낮다. 컴퓨팅 파워가 떨어지는 시스템에서 멀티미디어 데이터를 처리하기 위해서는 최소한의 기능

을 수행할 수 있는 라이트웨이트(Light Weight) 적응적 스트림 처리기를 개발해야 한다.

멀티미디어 자료 전송 연출 시 데이터의 입력 속도와 입력된 데이터의 처리속도 차로 인해 버퍼에 데이터가 증가할 수 있고, 이런 경우 메모리의 사용량이 증가한다. 이는 제한된 메모리를 갖는 PDA 상에서 문제가 되기 때문에, 이 문제를 해결하기 위해 적응적 버퍼 관리 기법을 사용한다.

- 다양한 단말기에 적합한 멀티미디어 데이터 전송 동영상을 포함한 멀티미디어 데이터를 시스템 사양이 낮은 단말기로 실시간 전송하거나 재생 시키기 위해서는 단말기의 사양에 맞도록 데이터를 조절할 수 있어야 한다.

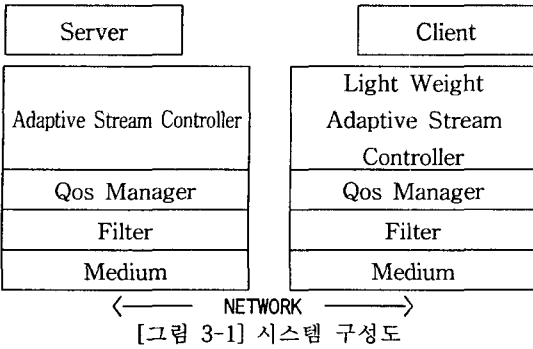
본 시스템에서는 필터 파이프라인(Filter Pipeline)을 구성하여 이 문제를 조절할 수 있다.

- 다양한 형태의 미디어와 필터

스트림 처리기를 위해서는 다양한 미디어(Medium)와 필터(Filter)를 사용한다. 미디어의 입출력 장치를 통해 데이터의 입출력이 처리되고, 필터를 통해서 데이터의 형변환이 이루어진다. 여기서 데이터의 입출력을 일관성 있게 처리할 필요가 있다. 디바이스(Device) 종속적인 미디어(카메라, 모니터, 마이크, 스피커 등)를 DLM(Dynamic Linking Module) 형태로 작성하여 이 문제를 해결할 수 있다.

3. 스트림 처리기 모델

- 클라이언트/서버 모델



스트림 처리기는 클라이언트(Client)-서버(Server) 모델이다. 스트림 처리기를 사용하고자 하는 응용 프로그램 즉, 클라이언트 프로세스(Client Process)는 먼저 통신망 상에 존재하는 스트림 서버와 접속하여야 한다. 이를 위하여 스트림 처리기의 클라이언트 라이브러리(Client Library)는 객체 지향형 API를 제공한다. (Essence *pEssence : 객체생성)

- 가상 객체 인터페이스

API는 가상 객체 인터페이스(Virtual Object Interface)를 사용하여 스트림 처리기 서버 내에서 멀티미디어 데이터를 처리한다. 가상 객체 인터페이스는 서버와 클라이언트간에 객체의 생성과 소멸, 그리고 소속 함수의 호출 등에 관한 특수한 프로토콜(Protocol)을 통하여 서버 내에 실제적인 객체가 존재하도록 하고, 클라이언트에는 가상적인 객체를 생성하게 하여 클라이언트로 하여금 마치 실제 객체처럼 사용할 수 있도록 한다. 따라서 API로 설명되는 모든 객체는 가상 객체라 할 수 있으며 실제 객체는 연결되어진 스트림 처리기 서버 내에 존재한다.

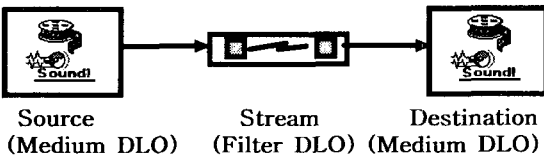
- 멀티미디어 데이터 흐름 모델 > 스트림

스트림은 단일 미디어의 흐름으로서 미디어 데이터 입·출력의 가장 기본적인 구조이다. 스트림은 미디어의 흐름을 원활하게 해주는데, 성능은 연속 미디어에 대해서는 연속성을 보장하고 이벤트형 미디어에 대해서는 실시간성을 보장한다. 스트림이 미디어의 흐름을 실현하기 위하여 세부적으로 다음과 같은 기능을 제공한다.

- 파일, 디바이스, 통신, IPC와의 입·출력 서비스
- 스트림 필터링(Filtering) 서비스

- 스트림 처리 요소

스트림 계층은 멀티미디어 데이터 입력 장치로부터 데이터를 캡처한 후 자료 사용매체로 데이터를 전송하여 재생하는 기능을 수행하는 계층이다.



[그림 3-2] 스트림 개념도

- 소스객체
소스 객체는 입력용 미디어를 소유하고 있는 객체로서 스트림에서 프레임(Frame) 단위로 데이터를 전송한다.

- 데스티네이션 객체
데스티네이션 객체는 출력용 미디어 객체를 소유하고 있는 객체로서 소스에서 생성된 프레임의 종착점이 된다. 따라서 프레임 구조체를 위하여 할당된 메모리를 돌려준다.

- 스트림 객체
스트림 객체는 [그림3-2]와 같이 소스 객체와 데스티네이션 객체의 연결로서, 소스 객체가 미디어 객체로부터 생성한 프레임은 데스티네이션 객체가 소유하고 있는 출력용 미디어 객체로 전달하는 기능을 수행한다. 특히 스트림 객체는 이러한 전달의 과정에 있어서 자신에게 등록된 필터 객체에 이전 미디어나 필터로부터 전달되어온 데이터를 적용함으로써 스트림 객체를 통과하는 데이터에 적절한 변화를 준다.

- 미디어 객체
미디어 객체는 각종 멀티미디어 디바이스, 파일 포맷(File Format), 통신 프로토콜과의 입·출력을 일관성 있게 하기 위하여 사용되는 객체로서 DLO 구조이다. 예로 카메라 DLO, 모니터 DLO, 스피커 DLO, TCP 통신용 DLO등이 있다.

- 필터 객체
필터 객체는 스트림 상에 위치하며, 입력되는 데이터를 가지고 이벤트를 발생하거나 데이터 변환 작업을 수행한다. 필터의 종류에는 이벤트 발생 필터와 데이터 변화 필터가 있다. 필터로는 H.263 인터더, H.263 디코더, ADPCM 인터더, ADPCM 디코더, 트랜스코더(Transcoder) 등이 있다.

4. 스트림 처리기 버퍼 관리 및 필터 파이프라인

-적응적 버퍼 관리 기능

PC상에서 카메라 미디어용이 YUV411 포맷(Format)의 QCIF(176×144) 크기로 이미지를 초당 15 프레임 캡처하고, 인코딩(Encoding)하여 PDA로 전송하는데, PDA에서는 초당 3 프레임을 디코딩한다고 가정하면, 디코더의 버퍼에는 초당 12 프레임만큼 데이터가 증가한다. 이는 제한된 메모리를 갖는 PDA 상에서 심각한 문제가 된다. 그래서 이 문제를 해결하기 위해 적응적 버퍼 관리 기법을 사용한다.

적응적 버퍼 관리 기법이란, 임계값(Threshold)을 설정하여, 이 값에 따라서 데이터의 입출력을 적응적으로 조절하여 버퍼의 크기가 증가하지 않고 일정한 상태를 유지하도록 하는 방법이다.

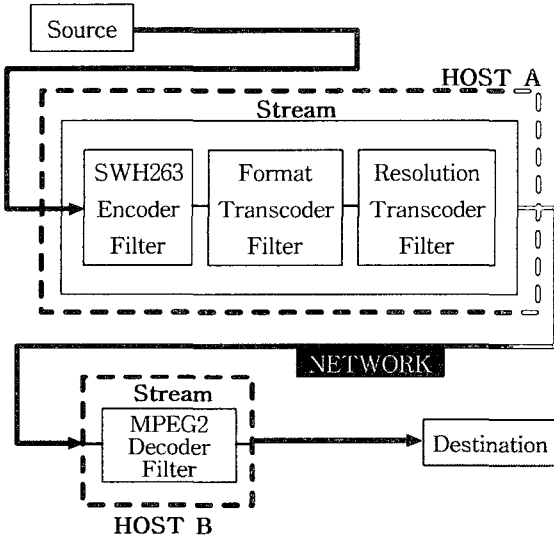
적응적 버퍼 관리 기법의 동작 알고리즘은 다음과 같다.

1. 버퍼의 임계값을 초기화한다.
2. 버퍼의 크기를 체크(Check)한다.
3. 버퍼의 크기가 임계값 보다 크면, 데이터의 흐름을 조절한다.
 - 3.1 데이터를 누락시킨다.
 - a. 데이터를 버퍼에 저장하지 않고 누락시킨다.
 - b. 데이터를 버퍼에 저장한 후, 데이터를 처리하기 전에 데이터를 누락시킨다.

- 3.2 데이터의 생성을 조절한다.(인코딩을 조절)
- 3.3 3.1과 3.2의 방법을 합한다.

- 필터 파이프라인 구성

필터 파이프라인이란 단말기의 QoS 정보를 반영하여 스트림 상에 필요한 데이터 변환 필터들을 연결시킬 때, 스트림 상에 존재하는 필터 리스트를 말한다. 필터 파이프라인을 사용하면 멀티미디어 데이터 전송 시, 단말기의 QoS 정보를 반영한 적절한 데이터 형 변환 필터들을 사용하여 실시간으로 데이터 변환하고 전송한다. 이를 통해 적응적 스트림 처리기로서의 역할을 수행한다.



[그림 4-1] 필터 파이프라인 구성 예

호스트 A에서 CIF(352×288) 크기의 비디오 데이터를 카메라 미디어를 통해서 캡처하여 인코딩한 후 Windows CE 기반의 PDA인 호스트 B에서 연출하기 위해서 H.263 포맷에서 MPEG-2 포맷으로 변환시키고, 변환된 데이터를 QCIF로 크기를 변환하여 네트워크를 통해서 전송한다. 호스트 B에서는 전송 받은 데이터를 MPEG-2 디코더 필터를 통해 디코딩한 후 PDA에서 데이터를 재생한다.

- DLM(Dynamic Linking Module) 구조

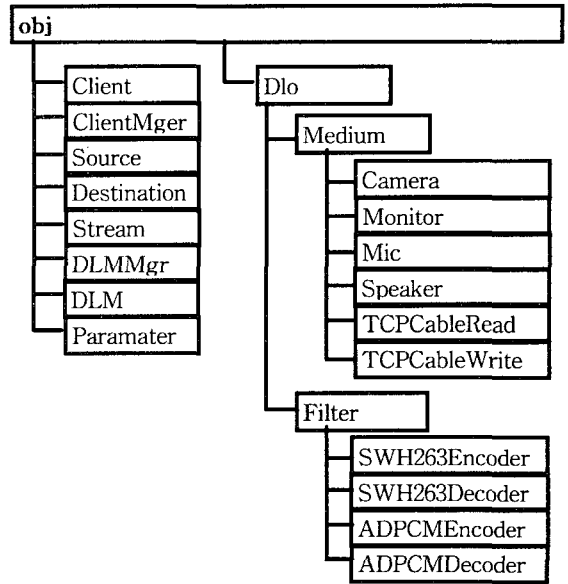
본 시스템은 각종 멀티미디어 디바이스를 손쉽게 지원할 수 있도록 DLO 객체는 DLL 기능을 사용하여 구현해야 한다. 새로운 디바이스를 시스템에서 지원하기 위한 DLL은 다음과 같은 부분으로 구성된다.

- DLL 인터페이스 함수 : DLL 정보, 새로운 인스턴스 생성
- DLO 베이스 클래스(Base Class) : 입출력 기능 기본 객체
- 미디어/필터 베이스 클래스
- 디바이스 스펙시픽(Specific) 클래스 : 특정 디바이스 기능을 위한 사용

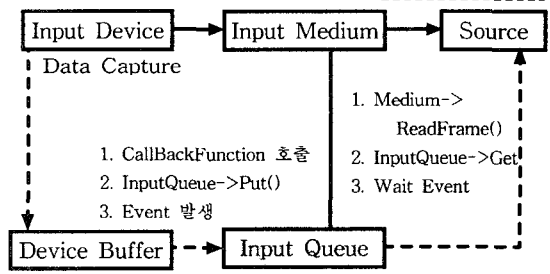
5. 스트림 처리기 설계 및 구현

- 모듈별 설계

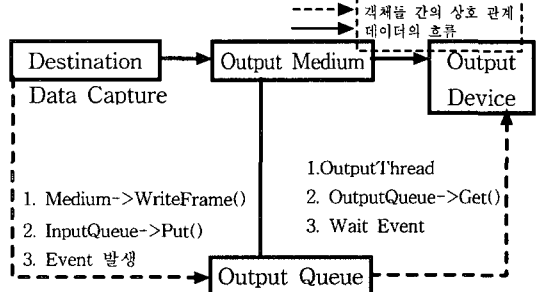
현재 구현된 스트림 처리기의 전체 클래스 계층도



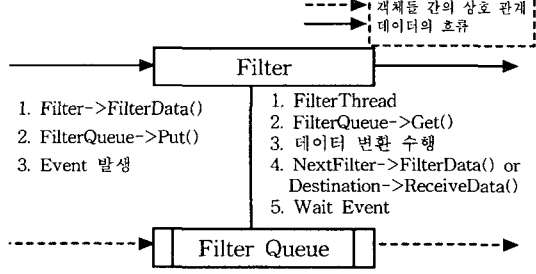
- 미디어 동작 모델 (입력) --- 객체들 간의 상호 관계
--- 데이터의 흐름



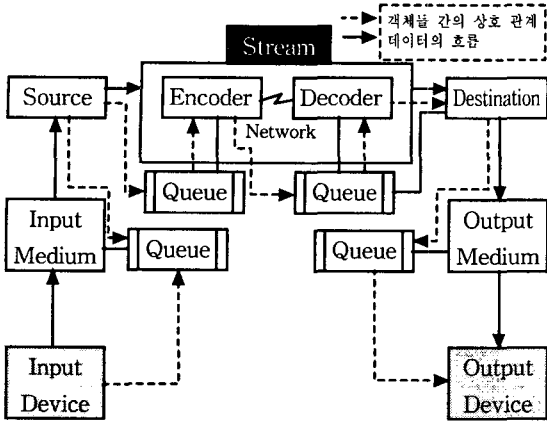
- 출력 미디어의 동작 모델



- 필터 동작 모델



- 스트림 전체 구성요소 및 동작의 예



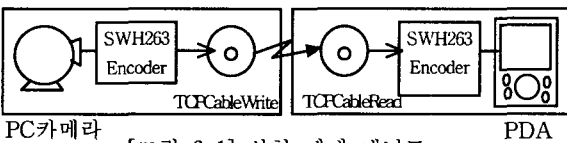
6. 실험 및 결과 분석

본 실험 환경은 다음과 같다.

- 로컬 호스트(Local Host) - PC
- 운영체제 : Windows 2000 Professional
- 하드웨어 : CPU - 450 MHz, 메모리 - 256M
- 스트림 처리기 : Windows를 위한 적응적 스트림 처리기 (Baby Essence for Windows)
- 미디어 : 카메라 미디어, TCPCable 읽기 미디어
- 필터 : H.263 인코딩 필터
- 리모트 호스트(Remote Host) - PDA
- 운영체제 : Windows CE 3.0
- 하드웨어 : CPU - 131MHz, 메모리 - 32M
- 스트림 처리기 : 이동형 운영체제를 위한 스트림 처리기 (Baby Essence for Windows CE)
- 미디어 : 모니터 미디어, TCPCable 쓰기 미디어
- 필터 : H.263 디코딩 필터

- 실험 예제

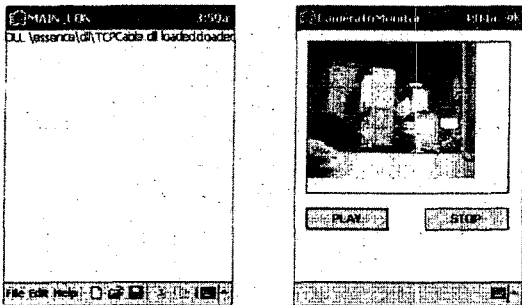
PC에서 카메라를 통해서 동영상 멀티미디어 데이터를 캡처한 후, H.263으로 데이터를 인코딩하여 TCPCable을 통하여 PDA로 데이터를 전송한다. PDA에서는 데이터를 전송 받아 H.263으로 디코딩한 후, PDA의 화면에 동영상 데이터를 출력한다.



PC카메라 [그림 6-1] 실험 예제 개념도

위 [그림 6-1]의 개념도에 따라 스트림 처리기에서 제공하는 API를 사용하여 프로그램화함.

- 예제 프로그램 실행 결과



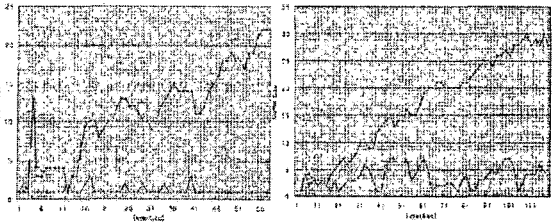
[그림 6-2] PDA 상에서 스트림 처리기 실행 결과

[그림 6-3] PDA 상에서 예제 프로그램 실행 결과

- 결과 분석

다음의 [그림 6-4]와 [그림 6-5]는 6.2의 실험 예제에서 4.1에서 제안한 적응적 버퍼 관리 기법을 적용하기 전과 적용한 후의 버퍼 크기 변화를 비교한 것이다. 버퍼 관리 기법을 적용하여 버퍼 크기를 일정하게 유지시켜 줌으로써 시스템 사양이 낮은 PDA에서 메모리가 사용량이 증가하는 문제를 해결할 수 있다.

7. 결론



[그림 6-4] 디코더 버퍼의 크기 변화

[그림 6-5] 출력 버퍼의 크기 변화

본 논문에서는 이동 단말기 중 하나인 Windows CE 기반 PDA를 위한 적응적 멀티미디어 스트림 처리기를 설계 및 구현하였다. 제한된 시스템 환경을 갖고 있는 PDA 상에서 멀티미디어 데이터를 실시간으로 전송 및 저장할 때 생기는 문제를 효율적으로 해결하기 위하여 적응적 버퍼 관리 기법을 사용하였다. 단말기에 적합한 형태로 멀티미디어 데이터를 변환하여 전송하기 위하여 필터를 사용하였고, 필터 파이프라인을 구성하였다. 또한 다양한 형태의 미디어와 필터의 입출력을 일관성 있게 처리하기 위해서 DLM(Dynamic Linking Module) 구조를 사용하였다.

PDA 상에서 멀티미디어 스트림 서비스가 가능해졌기 때문에, PDA를 무선 이동통신망과 연결한다면 이제 스트림 서비스 이용자는 언제 어디서나 원하는 미디어를 손쉽게 접할 수 있다.

참고문헌

- [1] Jeonghun Kim, Daewon Park, Mikyong Oh, Wonhee Choe, Younghwan Lim, "An Adaptive Stream Control Mechanism for Presentations with Guaranteed QoS on Wireless Internet," INF, France, 2001.
- [2] <http://www.microsoft.com/korea/msdn/embedded/default.asp>
- [3] Seunghyeon Lee, Jeonghun Kim, Mikyong Oh, Younghwan Lim "A PULL-based Data Transfer Method for HyperPresentation Mail On Internet," EALPIIT2000, pp.357-362, 2000.
- [4] 윤성규, 이승현 서광민, 임영환, "PUSH/PULL 방식을 이용한 프리젠테이션 QoS 향상 기법," 한국정보과학회 학술발표 논문집, Vol.26, No.1, pp.435-437, 1999.
- [5] Nick Grattan, Marshall Brain, Windows CE 3.0 : Application Programming; Prentice Hall, 2001.
- [6] <http://www.codeguru.com>
- [7] 임영환, "Combistation : 분산 멀티미디어 컴퓨팅 환경을 위한 컴퓨터 플랫폼," 정보과학회논문지, 제2권, 제1호, pp.160-181, 1996.