모바일 기반 미디어 다운로드 서버 설계 및 구현

백성호° 이은순 김태용
중앙대학교 첨단영상대학원 영상공학과
shbaek@gametech.cau.ac.kr°, eslee@kcachost.kcac.ac.kr, kimty@cau.ac.kr

Design and Implementation of Mobile Based Media Download Server

Seongho Baek° Eunsoon Lee Taeyong Kim
Dept. of Imaging Science, Graduate School of Advanced Imaging Science, Chung-Ang University

요 약

모바일 기반에서 콘텐츠 개발을 할 때는 단말기의 하드웨어 특성을 고려해야 한다. 즉, 단말기에 할당된 힙(Heap) 메모리 크기에 제한이 있기 때문에 많은 양의 미디어 데이터를 응용프로그램에 포함시킨다는 것은 한계가 있다. 본 논문에서는 이러한 문제점을 극복하기 위하여 이미지, 사운드, 문자열과 같은 미디어 데이터를 서버로부터 다운로드 받아 활용할 수 있는 미디어 다운로드 서버를 구현하고, 시스템 설계 시 고려해야 할 사항에 대해서 효과적인 접근 방법을 제시한다. 이로써 본 연구를 통하여 리소스 제약이 많은 모바일 환경에서 다양한 콘텐츠 개발을 가능하게 한다.

1. 서 론

최근 급속도로 발전하고 있는 이동통신 네트워크 기술과 포스트 PC시대에 부응하는 고성능 휴대형 단말기들의 출현으로 과거 PC상에서나 가능했던 콘텐츠들이 이제는 모바일 환경에서 구현되고 있으며, 고성능의 콘텐츠 서비스들도 충분히 가능해지고 있는 현실이다[1, 2]. 이러한 콘텐츠는 VM(Virtual Machine)기반의 미들웨어 플랫폼 위에서 실행되는 응용프로그램[3]을 말하며, 국내의 경우 이동통신 사업자 별로 다양한 플랫폼이 제공 되고 있다. 각 플랫폼 마다 각기 다른 특성과 장점들을 가지고 있어 콘텐츠 개발 시 많은 이점을 제공 하지만, 사실상 단말기의 하드웨어 특성을 고려하지 않을 수 없다. 즉, 단말기에 할당된 힙 메모리나 기타 작업 공간이 대단히 제한적이기 때문에 많은 양의 미디어 데이터를 한꺼번에 응용프로그램에 포함 시킨다는 것은 한계가 있다[4].

따라서 본 논문에서는 GNEX(General & Next multimedia player) 플랫폼 기반에서 이러한 문제점을 극복하기 위해서 응용프로그램 실행중 이미지, 사운드, 문자열과 같은 미디어 데이터를 서버로부터 다운로드 받아 활용할 수 있는 미디어 다운로드 서버 시스템의 설계와 구현을 목적으로 하고 있다. 또한 서버 모듈에서 단말기마다 채택하고 있는 트리비얼(Trivial), 슬라이딩 윈도우(Sliding Window), 매직 슬라이딩 윈도우(Magic Sliding Window)방식의 3가지 프로토콜을 처리하는 루틴에 중점을 두고 있으며, 미디어 데이터 다운로드 시 고려해야 할 사항들과 문제가 될 수 있는 부분에 대한 해결방안을 기술한다. 이로써 본 연구를 통하여 서로 상이한 미디어 다운로드 프로토콜 상에서 단말기의 데이터 요청 시 미디어 데이터를 제공함으로써 단말기의 열악한 환경을 극복하고 다양한 형태의 콘텐츠 개발을 가능하게 한다[5, 6].

본 논문의 구성은 다음과 같다. 2장에서는 미디어 다운로드 시스템의 구조와 설계 시 고려해야 할 사항 및 문제점에 대하여 제시하고, 3장에서는 제안한 방법을 설명한다. 4장에서 제안된 시스템의 구현 및 이를 검증하기 위한 실험 결과를 제시한다. 마지막으로 5장에서 본 논문의 결론을 맺는다.

2. 미디어 다운로드 시스템

2.1 시스템 개요

본 논문에서 제안하는 시스템은 응용프로그램 실행중 이미지, 사운드, 문자열과 같은 미디어 데이터를 서버로부터 다운로드 받아 활용할 수 있는 미디어 다운로드 시스템이다[8].

2.2 시스템 설계 고려 사항

2.2.1 미디어 다운로드 프로토콜 처리

미디어 다운로드 프로토콜은 그림1처럼 3가지 방식이 있으며 서버 모듈을 개발할 때에는 단말기마다 채택한 프로토콜이 서로 상이하기 때문에 3가지 방식을 모두 구현해야 한다. 각 방식의 핸드셰이킹(Handshaking) 과정은 다음과 같다.

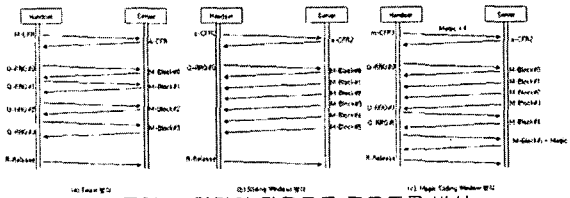

그림 1. 미디어 다운로드 프로토콜 방식

2.2.1.1 트리비얼 방식-그림1(a)

단말기는 CFR 패킷으로 미디어 데이터에 대한 다운로드를 요구하며 서버에서는 요청한 미디어 데이터를 찾으면, CFA 패킷으로 응답한다. 단말은 수신한 CFA 패킷이 정상적인 경우에는 RRQ0을 송신하며 서버는 RRQ#i 을 수신하면 i번째 블록 데이터를 단말기에 송신한다. 단말기는 요청한 i번째 블록 데이터를 수신하면 RRQ#i+1을 송신한다. 단말기는 수신이 완료되면 릴리즈 패킷을 송신한다.

2.2.1.2 슬라이딩 윈도우 방식 -그림1(b)

단말기는 CFR2 패킷으로 미디어 다운로드를 요구하며 서버에서는 요청한 미디어 데이터를 찾으면, CFA2 패킷으로 응답한다. 단말기는 수신한 CFA2 패킷이 정상적인 경우에는 RRQ0 을 송신하여 서버는 RRQi 을 수신하면 i번째 블록을 전송한다. 단말기는 수신이 완료되면 릴리즈 패킷을 송신한다.

2.2.1.3 매직 슬라이딩 윈도우 방식-그림1(c)

단말기는 m-CFR3 패킷으로 미디어 다운로드를 요구하며 서버에서는 요청한 미디어 데이터를 찾으면, a-CFR2 패킷으로 응답한다. 단말기는 수신한 a-CFR2 패킷이 정상적인 경우에는 Q-RRQ0을 송신하며 서버가 Q-RRQ#i 를 수신하면 #i+Magic 번째 블록 데이터를 전송한다. 단말기는 수신한 블록데이터를 확인한 후, 처리가 완료되었음을 알리는 Q-RRQ#+1을 송신한다. 수신이 완료되면 릴리즈 패킷을 송신한다.

2.2.2 미디어 다운로드 패킷 처리

단말기에서 서버로 송신하는 데이터는 내부적으로 송신할 데이터를 GNEX 모듈이 자동 패키지(Package) 처리하여 송신한다. 즉, 그림2처럼 사용자와 데이터의 앞과 뒷부분에 헤더(Header)와 트레일러(Trailer)정보가 추가되어 전송된다. 따라서 서버에서는 단말로부터 수신 받은 데이터 패킷을 분석하여 헤더와 트레일러를 제외한 순수한 사용자 데이터를 추출하여 사용하도록 구현해야 한다[7].

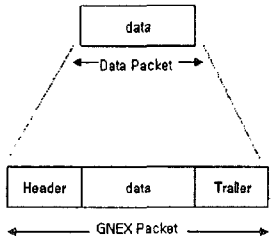

그림2. GNEX 프로토콜 패킷 구조

2.3 시스템 구조 및 서비스 방법

본 논문에서 설계 구현된 미디어 다운로드 시스템은 그림3과 같은 구조를 가진다. 단말기는 먼저 미디어 데이터를 송수신하는 클라이언트-서버 형식의 응용프로그램을 다운로드 받기 위해서 무선인터넷서비스에 접속한다. 무선 인터넷 서비스에 접속이 되면 PPP(Point-to-Point Protocol)연결이 설정되며, WAP(Wireless Application Protocol)서버는 응용프로그램을 다운로드 받을 수 있는 GTC(GNEX Technology Center)의 다운로드 서버로 연결할 수 있는 메뉴를 제공한다. 사용자가 제공된 메뉴에서 다운로드 할 응용프로그램을 선택하게 된다. Game Run WML 스크립트에 의하여 단말기와 다운로드 서버 간의 핸드셰이킹이 시작되고 실제로 다운로드가 수행된다. 다운로드가 완료된 후 응용프로그램에서는 필요에 따라 미디어 데이터를 요청하게 된다. 이때 미디어 다운로드 서버 측에서는 이미지, 사운드, 문자열과 같은 미디어 데이터를 모바일 환경에 적합한 바이너리 포맷(Binary Format)형태로 엔 코딩(Encoding)한 후 단말기에 제공한다.

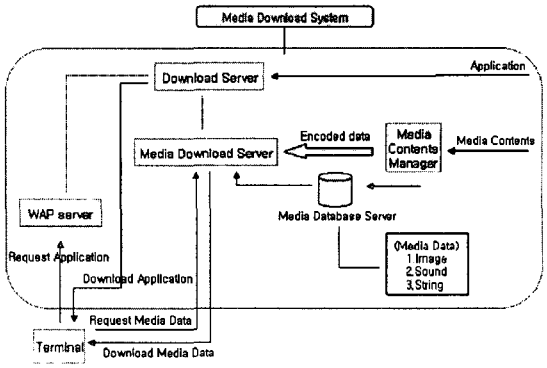

그림3. 미디어 다운로드 시스템 전체 구조

3.제안한 방법

3.1 미디어 다운로드 클라이언트-서버 모델 기법

그림4는 미디어 다운로드 클라이언트-서버 처리 모듈이다. 단말기의 응용프로그램에서 미디어 다운로드 서버에 미디어 데이터를 요청, 다운로드 받는 방법은 LoadMedia()함수를 이용한다. 이때 응용프로그램에서 다운로드 받을 미디어 데이터가 여러 개일 경우 이를 구별하기 위해서 서버 프로그램과 정해진 약속에 의해 키(Key)값을 정하고 그 값에 해당하는 미디어 데이터를 다운로드 하도록 한다. 서버 모듈에서는 구현된 3가지 미디어 다운로드 프로토콜 환경에서 접속한 단말기들의 채택한 프로토콜에 맞게 goCFR(), goRRQ()함수를 이용하여 미디어 데이터를 제공한다.

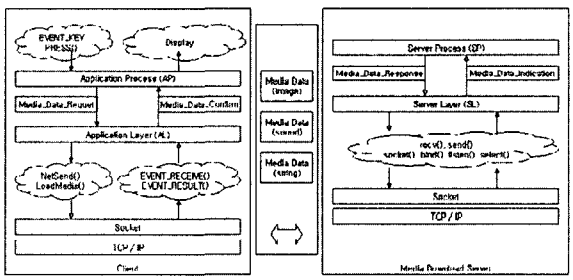

그림4. 미디어 다운로드 클라이언트-서버 알고리즘 모듈

3.1.1 다운로드 프로토콜 처리 루틴

다운로드 프로토콜 처리 루틴은 단말기가 미디어 다운로드 서버에 접속하여 미디어 데이터를 요청했을 경우 규정된 패킷의 오피 코드(operation code)표를 이용해 트리비얼, 슬라이딩, 매직 슬라이딩의 3가지 프로토콜 방식을 구별하고, CFR, RRQ, RELEASE상태를 확인한다.

3.1.2 CFR 처리 루틴

CFR처리 루틴은 단말기의 미디어 데이터에 대한 요청이 들어왔을 때 그림5와 같이 제공할 미디어 데이터의 정보를 응답한다.

```
00: ret = findChecksum(imagePath, &size, &checksum) ;
01: memset(packet, 0, SBUF_LEN);
02: packet[0] = 'S'
```

```
03: packet[1] = 'G'
04: packet[2] = 'G'
05: packet[3] = 'A or a'
06: packet[4] = HIBYTE(sDataLen);
07: packet[5] = LOBYTE(sDataLen);
08: packet[6] = HIBYTE(size);
09: packet[7] = LOBYTE(size);
10: packet[8] = HIBYTE(checksum);
11: packet[9] = LOBYTE(checksum);
12: packet[10]=addPacketSum(&packet[6], sDataLen);
13: totLen = sDataLen + 7 ;
14: ret = send(clientSocket, packet, totLen, 0);
```

그림5. CFR처리 루틴

3.1.3 RRQ 처리 루틴

RRQ처리 루틴은 CFA 패킷이 정상적인 경우에 트리비얼, 슬라이딩, 매직 슬라이딩의 3가지 프로토콜 방식에 따라 블록 데이터를 송신한다.

4. 구현 및 실험 결과

본 논문에서 제안한 시스템의 서버와 클라이언트는 Windows-XP Professional 운영체제를 갖는 펜티엄 4 1.8GHz PC를 사용하여 구현하였으며, MS VC++ 6.0컴파일러, GNEX SDK 1.24 Version을 사용해 에뮬레이터(Emulator)에서 작업을 마친 뒤 실제 단말기에서 동작하도록 하였다.

미디어 다운로드 시스템의 서버 프로그램을 하기 위하여 마이크로소프트에서 제공하는 윈도우즈 소켓을 사용했다. 본 논문에서 구현한 미디어 다운로드 시스템의 서버와 클라이언트(단말기)의 실행 결과 화면은 그림6과 그림7에서 보인다. 그림6에서는 서버는 초기 서버소켓(Server Socket)을 생성하고 바인드(Bind)를 한다. 리슨(Listen)상태에서 단말기의 접속요청을 기다리다 접속이 이루어지면 해당 단말기의 접속 아이피와 포트번호를 보여주고, 단말기에서 이미지 데이터 요청 시 해당하는 키 값에 대응하는 미디어 데이터를 단말기의 프로토콜 전송 방식에 맞게 제공한다.

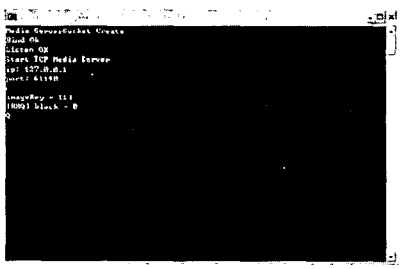

그림6. 미디어 다운로드 서버

단말기 응용프로그램을 실행시키면 그림7의 좌측화면처럼 1~3번까지 목록이 나온다. 1~3번의 키 값을 선택하게 되면 미디어 다운로드 서버로 각 키 값에 대한 미디어 데이터를 요청하게 되고 서버는 키 값에 대응하는 이미지 데이터를 제공한다. 우측은 서버로부터 다운로드 받은 이미지 결과 화면이다. 이미지의 크기는 128*128픽셀(pixel)의 약 11.2KB용량의 이미지이며 실제 폰에서 시뮬레이션 결과 동일한 화면을 보여줬다.

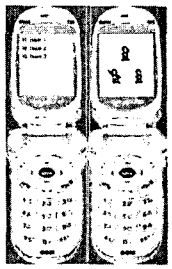

그림7. 미디어 다운로드 결과

5. 결론

본 논문에서는 모바일 기반에서 미디어 데이터를 다운로드 받을 수 있는 서버 시스템을 구현하였다. 제안된 프로토콜 처리 알고리즘을 이용하여 서로 상이한 프로토콜을 가지고 있는 단말기의 접속 시 해당하는 프로토콜에 맞게 다운로드를 적용시켰으며, CFR, RRQ 처리 알고리즘을 이용하여 미디어 다운로드 프로토콜 패킷규격에 맞는 데이터를 단말기에게 제공했다. 시뮬레이션 한 결과 서버로부터 128*128픽셀의 약11.2KB 용량의 미디어 데이터 다운로드 가능함을 증명하였다. 이를 통하여 리소스 제약이 많은 모바일 환경에서 다양한 형태의 콘텐츠 개발을 가능하게 할 수 있다.

향후 연구 방향은 미디어 데이터 압축 연구를 진행시켜 최적화된 알고리즘에 의한 데이터 용량의 최소화로 데이터 전송의 최적화를 제공해야 할 것이다.

참고문헌

[1]Wetherall, D., Legedza, D. and Guttag, J., "Introducing new Internet services: why and how", Network. IEEE, Volume: 12, Issue: 3, pp. 12-19, May-June 1998.

[2]김기천, "모바일 서비스 기술 동향", 정보처리학회지, 제9권, 제2호, 2002.

[3]J. Bates, D. Halls and J.Bacon, "Middleware support for mobile multimedia applications", ICL Systems Journal, pp. 289-314, November 1997.

[4]황병연, 오영석, "모바일 게임을 위한 압축 기술", 정보처리학회지, 제9권, 제3호, 2002.

[5]Lei Huang, Uwe Horn, Frank Hartung and Markus Kampmann, "Proxy-based TCP-friendly streaming over mobile networks", SIGMOBILE. ACM, pp.17-24, 2002.

[6]Sumit Roy, Bo Shen, Vijay Sundaram and Raj Kumar, "Application Level Hand-off Support for Mobile Media Transcoding Sessions", NOSSDAV. ACM, pp.95-104, 2002.

[7]Peng Ge, McKinley, P.K, "Comparisons of error control techniques for wireless video multicasting", Performance ,computing and Communications Conference. 21st IEEE , pp.93-102, 2002.