

2013학년도

경기과학고 기초 R&E

모바일 기기에서의 문서 스트리밍을 위한 기술 연구

Study about Technique for Document Streaming at Mobile Devices

2013. 11. 24

연구 참여자 : 박현민(525hm@naver.com)

공동 연구자 : 강한필(ho94949@gmail.com), 이재범(woqja125@naver.com)

연구 책임자 : 전현석 (what_is_computer@msn.com)

과학영재학교 경기과학고등학교

모바일 기기에서의 문서 스트리밍을 위한 기술 연구

Study about Technique for Document Streaming at Mobile Devices

초 록

스마트폰과 태블릿이 상용화됨에 따라 이를 이용하여 언제 어디서나 오피스 작업을 하는 시대가 되었다. 그러나 점점 커지는 문서 파일의 용량, 제한된 무선 인터넷 속도와 프로세서 성능, 저장 공간의 용량 등으로 인해 해당 문서 파일의 전체를 다루기에는 아직도 데스크탑 PC보다 어려운 점이 많다. 따라서 본 논문에서는 이러한 문서 파일의 전송과 처리 등을 쉽게 하기 위한 구체적 방안으로 문서 스트리밍 서비스를 제시하였다. 또한 랜덤 접근에 용이한 파일 포맷들을 탐색했고 문서 스트리밍 환경 구현을 위해 다양한 문서 파일에 대해 간단한 분석으로 목차를 만들어 미리보기 형태로 보여주는 발전된 형태의 미리보기를 제시하는 방법과 그로부터 연결되는 특정 부분의 스트리밍 시 다양한 미디어 파일 등의 전송 방법에 대해 연구하였다.

본 문

I. 서론

1.1 이론적 배경

1.1.1 문서 파일(Document File)

문서 파일은 일련의 문자들로 구성된 파일로 동영상, 소리 등의 미디어 파일과는 다르게 표, 그림 및 일부 동영상과 소리들도 그 내부에 포함하고 있다.

1.1.2 스트리밍 서비스(Streaming Service)

스트리밍 서비스는 미디어를 원하는 부분부터 다운로드 받아 와서 일부분만 받고도 재생할 수 있는 기술로 재생 시작까지 걸리는 시간을 효과적으로 줄일 수 있는 방법이다. 인터넷 상에 많은 음악 스트리밍 서비스와 Youtube 등의 동영상 스트리

밍 서비스가 많이 활성화 되어 있다.

1.1.3 바이트 순서 표식(Byte Order Mask, BOM)

TXT 파일의 시작 부분에서 그 파일의 인코딩 형식과 바이트 순서, 엔디언 타입 등을 나타내기 위해 사용된다. ANSI 인코딩의 경우에는 값이 없으며, UTF-8일 때는 EF BB BF, UTF-16LE(Little Endian)인 경우에는 FF FE, UTF-16BE(Big Endian) 경우에는 FE FF의 값을 갖는다.

1.1.4 엔디언(Endian)

엔디언은 다중 바이트의 값을 갖는 변수를 나타낼 때 저장 장소에 기록하는 순서를 나타내는 용어이다. 리틀 엔디언(Little Endian)은 바이트 순서를 뒤집어서 표현하여 더 작은 단위로의 변환에 유리한 특성을 가지며 인텔 계열 시스템에서 사용된다. 빅 엔디언(Big Endian)은 원래의 바이트 순서대로 저장 장소에 기억되며 모토롤라 계열 시스템과 인터넷 전송 규약 등에서 사용된다.

1.1.5 RFC(Request for Comments) 문서

RFC 문서는 비평을 기다리는 문서라는 의미로, 컴퓨터 네트워크 공학 등에서 인터넷 기술에 적용 가능한 새로운 연구, 혁신, 기법 등을 적어 놓은 문서들을 나타낸다. 인터넷국제표준화기구(IETF)는 일부 RFC들을 인터넷 표준으로 받아들이기도 한다.

1.2 연구의 필요성



그림1 iPad를 이용한 문서 작업

Fig.1 Document work using iPad

요즈음에는 스마트폰과 태블릿이 상용화됨에 따라 이를 이용하여 언제 어디서나 오피스 작업을 하는 시대가 되었다.

교사지도결과보고서.pdf	2012-03-12 오후...	Adobe Acrobat D...	50,839KB
교사지도결과보고서-완료.hwp	2012-03-12 오후...	한컴오피스 한글 ...	311,509KB
교수지도결과보고서-완료.hwp	2012-03-12 오후...	한컴오피스 한글 ...	148,213KB
교수지도결과보고서-완료.pdf	2012-03-12 오후...	Adobe Acrobat D...	49,765KB
기초R&E.hwp	2013-03-14 오전...	한컴오피스 한글 ...	515,048KB
기초R&E.pdf	2013-03-14 오전...	Adobe Acrobat D...	53,454KB
심화R&E.hwp	2013-03-14 오전...	한컴오피스 한글 ...	147,160KB
심화R&E.pdf	2013-03-14 오전...	Adobe Acrobat D...	42,204KB

그림2 용량이 큰 문서 파일들

Fig.2 Large document files

한편 스마트폰 카메라처럼 카메라 관련 주변 기기의 발전에 따라 그림2처럼 많은 사진 등을 포함하여 용량이 큰 문서 파일들도 많이 생성되고 있다. 위와 같은 파일들은 데스크탑 PC의 하드 디스크로부터 읽어 오는데도 상당한 시간이 소요된다. 이런 파일들을 하드 디스크보다 느린 플래시 메모리를 사용하는 스마트폰과 태블

랏에서 읽는 것에 더 많은 시간이 걸린다는 것은 당연한 일이다.



그림3 3G와 LTE의 인터넷 속도 테스트

Fig.3 3G and LTE's internet speed test

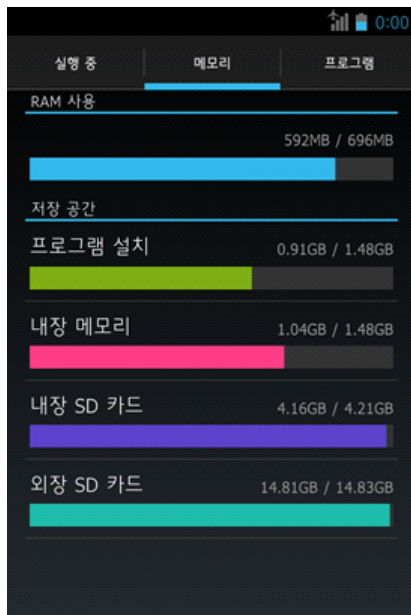


그림4 스마트폰의 제한된 RAM과 저장 공간

Fig.4 Limited RAM and storage space on smartphone

또한 그림3과 그림4에서 알 수 있듯이 휴대용 기기는 무선 인터넷 속도 및 RAM, 저

장 공간 등이 상당히 제한되어 있어 문서 파일을 전부 저장하고 RAM 상으로 불러들이는 일도 한계가 있다. 이런 환경에서 점점 용량이 커지는 문서 파일의 스트리밍 서비스는 앞으로 점점 필요성이 증가할 것이다.

1.3 연구 현황

동영상이나 소리처럼 흘러가는 데이터를 인터넷을 전송하는 서비스는 현재 인터넷 상에도 많이 개발되어져 있고 서비스되고 있다. 그러나 그 이외의 데이터들, 특히 문서 파일에 대해서는 스트리밍이라는 기술이 접목된 적이 많지 않다.

이광영(2006)은 "스트리밍 기술을 이용한 전자문서의 서비스 비용 개선에 대한 연구"에서 전자문서의 스트리밍 기술을 연구했고, 문서에 대한 스트리밍을 처리하기 위한 지금보다 더 최적화된 프로토콜의 개발과 모든 종류의 문서파일을 처리할 수 있는 일관화된 viewer의 개발 및 네트워크 자원의 활용을 위해 변화된 부분에 대한 데이터만 처리가 될 수 있도록 하는 기술이 필요하다고 했다.[1]

앞에서 언급했던 것처럼 스트리밍 서비스는 동영상과 소리 부분에는 많이 개발되어 있지만 문서 부분에서 개발된 것은 많지 않고 이미 개발된 것도 한계점이 있다. 엄밀하게 말하자면 스트리밍까지는 아니지만 비슷한 예를 들자면, 네이버의 N드라이브 같은 어떤 클라우드 서비스들은 문서를 페이지 단위 미리보기 서비스를 제공하는 하지만 편집 기능은 없고, 구글의 Google Docs 서비스는 웹상에서의 편집을 지원하지만 웹 브라우저에서 구동된다는 특징의 한계로 특정 단축키나 워드 프로세서에서는 가능한 특정 동작이 제한되어 있으며 기존 파일과의 호환 문제도 존재한다. 한편 실제 스트리밍 방식으로 설계되었더라도 그 특징을 적절히 사용하지 못하고 보안성을 중시한 제품으로 출시된 경우도 있다. Logicplant사의 uZard Viewer은 스트리밍 방식을 통한 빠른 열람, 유출 원천 차단 등을 특징으로 가지고 출시된 '뷰어'인데[2], 이는 문서 스트리밍 시 단순히 문서 전체를 하나의 스트림으로 보고 스트리밍해 주는 것이기 때문에 사진과 글이 복합되어 있는 요즘의 문서들은 효율이 떨어지며, 뷰어이기에 양방향 통신도 불가능하다는 문제점이 있다.

이렇기에 다양한 포맷의 파일들에 대해 스트리밍 및 쌍방향 통신을 지원하는 서비스가 필요하고 이를 위해 간단한 처리를 통해서 많은 종류의 문서들의 목차와 각 문단의 도입부 정도를 알아낼 수 있도록 하는 알고리즘 및 스트리밍시 글과 미디어 파일의 전송 형태를 다르게 하는 것이 필요하다.

1.4 연구 목표

1.2에서 소개한 것처럼 현재는 문서 파일을 보거나 편집하려면 처음부터 끝까지 모두 다운로드 해야 하기에 네트워크에 상당한 부하가 걸릴 수 있으며 이에 따라 모바일 기기의 경우 간단하게 할 작업에 비해 시간이 낭비될 수 있다. 따라서 본 논문에서는 보고 싶은 부분부터, 또한 모두 다운로드 받지 않고도 문서를 볼 수 있도록 랜덤 접근에 용이한 파일 포맷들을 탐색하였으며, 이를 전송해주는 스트리밍 알고리즘 및 그 프로토콜 등에 대해 연구하였다. 또한 문서 스트리밍 환경 구현을 위해 문서 파일에 대한 간단한 구조적 분석 등으로 목차 및 각 문단의 개요 정도를 미리보기 형태로 보여주는 발전된 형태의 미리보기 및 그로부터 연결되는 특정 부분의 스트리밍 시 미디어 파일의 전송 방법에 대해 연구하였다.

II. 연구과정 및 방법

2.1 파일 포맷 연구

2.1.1 TXT - Notepad 등

ANSI, UTF-8, UTF-16 방식으로 텍스트 파일들을 직접 저장해서 hexa 에디터 등으로 내부 구조를 분석하고 각 인코딩에 관련된 RFC 문서들을 참고해 정리하였다.

2.1.2 CSV - Microsoft Excel 등

엑셀로 표 형태의 데이터를 저장해 텍스트 편집기로 열어 분석하고 해당 인터넷 표준 문서인 RFC 4180을 참고해 정리하였다.

2.1.3 HWP - 한글과컴퓨터 한글 97~2010

한글과컴퓨터에서 2010년 6월 30일에 공개한[2] "한글 문서 파일 구조" 문서[3]를 통해 한글 97까지 사용된 HWP 버전 3.X 파일, 그 후 워디안 및 한글 2002부터

2010까지 사용된 HWP 버전 5.0 파일의 구조에 대해 조사하였다.

2.1.4 HML - 한글과컴퓨터 한글 97~2014

한글과컴퓨터에서 2010년 6월 30일에 공개한[3] "한글 문서 파일 구조" 문서[4]를 통해 W3C XML 구조에 기반을 두고 설계된 HWPML 문서의 구조 및 특징에 대해 조사하였다.

2.1.5 HWPX - 한글과컴퓨터 한글 2014

한글과컴퓨터에서 2013년 10월 10일에 출시한 한글 2014를 이용해 HWPX 문서를 생성해 보고, 2011년에 제정된 KS X 6101 OWPML 표준[5]을 통해 HWPX 문서의 구조 및 특징을 분석하였다.

2.1.6 DOC, XLS, PPT - Microsoft Office 97~2003

Microsoft에서 2009년에 공개한 Office 파일 포맷 문서[6]를 통해 문서 작업에 많이 사용되었던 Microsoft Office 97~2003의 저장 형식인 DOC, XLS, PPT 파일의 특징을 분석하였다.

2.1.7 DOCX, XLSX, PPTX - Microsoft Office 2007~2013

Microsoft에서 2009년에 공개한 Word, Excel, PowerPoint 표준 지원 문서[7]를 통해 문서 작업에 많이 사용되는 Microsoft Office 2007~2013의 저장 형식인 DOCX, XLSX, PPTX 파일의 구조와 특징을 분석하였다.

2.1.8 PDF - Adobe Acrobat

Adobe에서 1993년부터 2011년까지[8] 지속적으로 포맷을 발전시키며 공개한 PDF Reference 문서[9]를 통해 장기 보존 문서로 사용되는 PDF 파일의 특징을 분석하였다.

2.2 스트리밍 프로토콜 연구

2.2.1 TCP(Transmission Control Protocol)

인터넷 상의 두 기기가 정보를 주고받기 위해 사용하는 Transport Layer 프로토콜 중의 하나인 TCP의 특징 및 장단점을 RFC 문서 등을 통해 분석하였다.

2.2.2 UDP(User Datagram Protocol)

인터넷 상의 두 기기가 정보를 주고받기 위해 사용하는 Transport Layer 프로토콜 중의 하나인 UDP의 특징 및 장단점을 RFC 문서 등을 통해 분석했고 앞에서 언급한 TCP와 비교하였다.

2.3 문서 파일에의 스트리밍 적용 방법 연구

2.3.1 목차 및 각 문단 개요의 빠른 추출 및 미리보기 지원

문서 파일을 스트리밍해서 빠르게 열었다고 해도 사실 원하는 내용을 찾아가기에는 해당 문서에서 목차가 있는 경우 그 곳까지 봐야 하고, 없는 경우에는 직접 하나 하나 읽어보면서 찾아야 하는 불편함이 있다. 이를 해결하기 위해서 목차 및 각 문단의 개요를 빠르게 추출하고 이를 먼저 보내줘서 미리보기를 할 수 있도록 하는 방법에 대해 연구하였다.

2.3.2 글과 미디어 파일과의 전송 방법 차별화

글과 미디어가 모두 포함된 파일을 통째로 한 스트림으로 보고 스트리밍하는 것은 효율성 측면에서 원래의 방법보다 그렇게 큰 장점이 없다. 이를 발전시키기 위해 글과 미디어 파일의 전송 방법을 차별화하는 것을 제시하고 이를 적용했을 때의 장점을 예상해 보았다.

III. 결과

3.1 파일 포맷 연구

3.1.1 TXT - Notepad 등

ANSI 방식은 BOM(Byte Order Mask)이 없으며 단지 ANSI 부호로 인코딩된 데이터를 선형으로 기록하는 방식이다.

UTF-8(UCS Transformation Format - 8-bit) 방식은 U+FEFF를 이 형식으로 인

코딩한 EF BB BF의 BOM을 가지며 각 문자는 아래 표 I 와 같이 인코딩된다.[10]

표 I 유니코드 범위에 따른 UTF-8 표현

Table I UTF-8 expression by unicode range

범위 시작	범위 끝	UTF-8 표현
U+ 0000	U+ 007F	0xxxxxxx
U+ 0080	U+ 07FF	110xxxxx 10xxxxxx
U+ 0800	U+ FFFF	1110xxxx 10xxxxxx 10xxxxxx
U+ 10000	U+ 1FFFFF	11110xxx 10xxxxxx 10xxxxxx 10xxxxxx
U+ 200000	U+ 3FFFFFFF	111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx
U+ 4000000	U+ 7FFFFFFF	1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx

* 회색으로 처리된 부분은 RFC 3629에서 UTF-16 사용 최대 범위인 4바이트에 맞추기 위해 사용이 제한되었다.[11]

이 방식은 널(NULL, U+ 0000) 문자를 제외하고는 0x00이라는 값이 나오지 않아 쉬운 처리를 할 수 있고, 각 문자의 첫 바이트에 존재하는 비트 수에 따라 뒤에 연결되는 바이트 수를 알 수 있으며 스트리밍을 위한 랜덤 접근을 하더라도 상위 두 개의 비트가 10일 경우 그 바이트부터 인코딩할 수 없다는 판단을 쉽게 할 수 있다.

UTF-16(UCS Transformation Format - 16-bit) 방식은 엔디언에 따라 UTF-16LE(UTF-16 Little Endian), UTF-16BE(UTF-16 Big Endian)으로 구분된다. 각각의 BOM은 FF FE, FE FF이다. UTF-16BE에 따른 각 문자의 표현 방식은 아래 표 II와 같다.[12]

표 II 유니코드 범위에 따른 UTF-16BE 표현

Table II UTF-16BE expression by unicode range

범위 시작	범위 끝	UTF-16BE 표현
U+ 0000	U+ FFFF	xxxxxxxx xxxxxxxx
U+ 10000	U+ 10FFFF	110110XX XXxxxxxxxx 110111xx xxxxxxxx

* XXXX : 상위 5비트에서 1을 뺀 값. 상위 5비트 값의 범위가 0x01~0x10이므로 1을 빼면 모두 4비트로 표현할 수 있다.

이 방식은 모든 글자가 2의 배수 바이트로 존재하므로 비순차적 접근을 할 때 2의 배수 단위로 접근하여 읽으면 BMP 이외의 글자를 읽지 않는 이상 전혀 문제가 되지 않는다. 사실 거의 모든 UTF-16 방식 문서에는 BMP 외부 글자가 존재하지도 않기에 비순차적 접근에 있어서는 안전하다고 볼 수 있다.

앞에서 살펴본 모든 인코딩들에 대해 TXT 파일들은 첫 부분에 BOM만 몇 바이트가 존재할 뿐 별다른 헤더가 없고 압축도 전혀 되어 있지 않다. 따라서 랜덤 접근에 매우 용이하지만 에디터 등의 한계로 상당히 기능이 부족하다는 결론을 내렸다.

3.1.2 CSV - Microsoft Excel 등

CSV는 Comma-separated Values의 약자로 그 파일에서 표의 각 열은 , (콤마)로, 각 행은 ↵ (엔터)로 분리되어 있다. 각 열 도중 콤마나 엔터가 들어갈 경우 "" (쌍따옴표)로 감싸주기도 한다.[13]

이 파일 형식 역시도 TXT와 마찬가지로 헤더와 압축이 존재하지 않아서 비순차적 접근이 쉽기 때문에 스트리밍에는 적합하나 표 형태의 데이터 전용으로 설계된 포맷이기에 기능이 부실하다는 결론을 내렸다.

3.1.3 HWP - 한글과컴퓨터 한글 97~2010

들어가기에 앞서 해당 파일 포맷 공개 문서의 저작권 조항에 의해 본 내용은 한글과 컴퓨터의 한글 문서 파일(.hwp) 공개 문서[4]를 참고하여 작성하였음을 밝힌다.

한글 97까지 사용했던 한글 3.x 문서 파일 구조는 일반적인 이진 파일 구조이며, 고

정 길이 블록인 파일 헤더, 문서 정보, 문서 요약 블록과 가변 길이 블록인 정보 블록, 폰트 이름, 스타일, 문단 리스트, 추가 정보 블록으로 구성되어 있다. 한 번의 순차적 접근으로 모든 내용을 파악할 수 있으며, 비순차적 접근이 필요하지 않다. 그러나 임의의 위치에 있을 때 현재 어느 블록에 존재하는지를 판별할 수 없고 각 블록의 위치를 찾기도 힘들다. 따라서 스트리밍 서비스에는 그다지 적합하지 않다.

그 이후로 제작되어 현재까지 사용되고 있는 한글 5.0 문서 파일 구조는 윈도우즈 복합 파일(Windows Compound File) 구조를 가진다. 이는 그 내부에 레코드와 스트림을 갖고 있는 트리 구조의 파일이어서 확장성과 상위 호환성이 매우 좋다. 따라서 스트리밍에 상당히 적합하다.

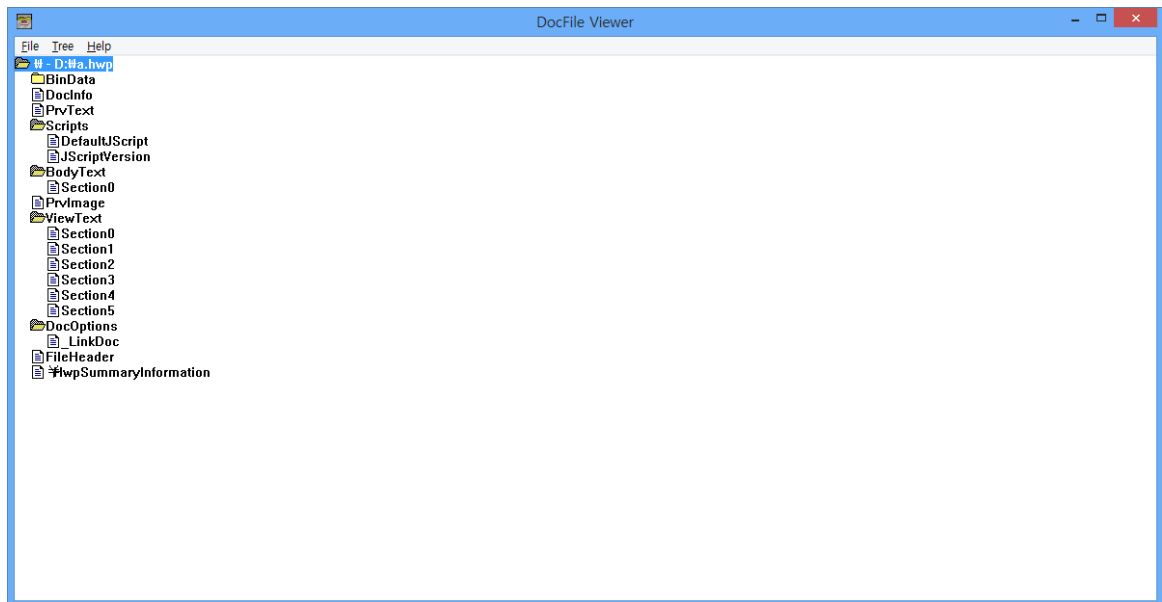


그림5 DocFile Viewer로 본 HWP 문서 파일의 레코드 및 스트림 구성 형태

Fig.5 HWP document file's record and stream structure using DocFile Viewer

3.1.4 HML - 한글과컴퓨터 한글 97~2014

HML을 확장자명으로 가지며 W3C XML에 기초를 둔 HWPML 파일은 일반 XML 파서로도 읽을 수 있을 정도로 HWP 파일보다 더 많은 확장성과 상위 호환성을 가지며 XML의 특성상 그 자체로 이미 트리 형태가 구성되어 있다. 따라서 특정 데

이터를 찾기가 매우 쉽고, 이에 따라 스트리밍에 매우 적합하다고 판단했다.

3.1.5 HWPX - 한글과컴퓨터 한글 2014

HWP 형식의 문서 파일에 대한 개방성을 확보하기 위해 제정된 HWPX(OWPML) 표준(KS X 6101)은 ZIP 파일로 압축된 다중 XML 파일 구조를 갖는다. 간단한 압축 해제 소프트웨어로 XML 파일들을 얻을 수 있고 이를 트리 구조로 구성하여 특정 노드 부분의 데이터를 보내줄 수 있기에 스트리밍에 적합하다.

3.1.6 DOC, XLS, PPT - Microsoft Office 97~2003

Microsoft Office 97~2003에서 기본 포맷으로 사용하는 DOC(Word), XLS(Excel), PPT(PowerPoint) 파일들은 모두 자체적 바이너리 파일로 되어 있다. 이는 한글 3.x 문서 파일 구조와 비슷하여 임의의 위치에 있을 때 현재 어느 블록에 존재하는지를 판별할 수 없고 각 블록의 위치를 찾기도 힘들다. 따라서 이 포맷 역시도 스트리밍 서비스에는 그다지 적합하지 않다.

3.1.7 DOCX, XLSX, PPTX - Microsoft Office 2007~2013

Microsoft Office 2007~2013에서 기본 포맷으로 사용하는 DOCX(Word), XLSX(Excel), PPTX(PowerPoint) 파일들 중 암호가 걸려 있지 않은 파일들은 모두 ZIP 형태의 파일로 확장자를 ZIP으로 바꾸면 일반 압축 프로그램에서 모두 열어볼 수 있다.

각 파일의 내용은 그림6처럼 DOCX 파일의 경우 모든 내용이 word/document.xml에, XLSX 파일의 경우 모든 내용이 xl/workbook.xml에 있으며, PPTX 파일의 경우에는 슬라이드별로 구분된 내용이 ppt/slides/slideN.xml에 저장된다.

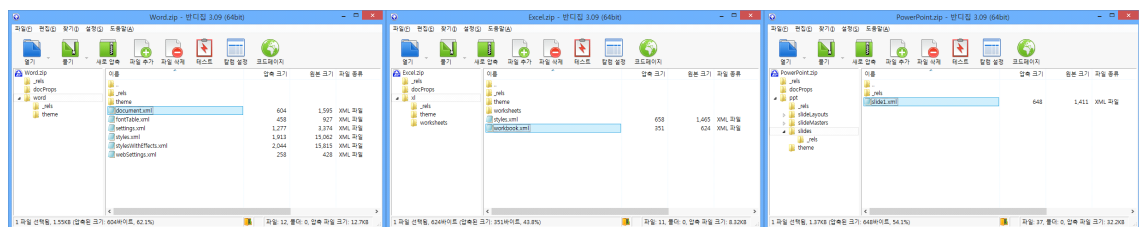


그림6 DOCX, XLSX, PPTX 파일에서 문서 내용이 위치한 XML 파일

Fig.6 XML file which has document content in DOCX, XLSX, PPTX file

이 세 파일들은 HWPX 파일과 비슷한 특징을 가지고 있어 간단한 처리로 문서의 구조를 모두 파악할 수 있고 해당 데이터를 전송하기에도 적합할 것으로 보인다.

3.1.8 PDF - Adobe Acrobat

PDF는 수많은 obj(오브젝트)로 구성된 문서 파일이다. 파일의 끝 부분에 xref 테이블을 갖고 있어 임의의 obj로 접근하기 쉽고 각 obj는 모두 독립적이어서 각 단위 별로 전송하는 데에도 유리할 것으로 보이기에 스트리밍에 적합하다고 판단했다.

3.2 프로토콜 연구

3.2.1 TCP(Transmission Control Protocol)

TCP는 RFC 793으로 정의되어 있으며[14] 인터넷 표준이다. 이 프로토콜은 안전하고 정확한 전송을 목표로 하는 프로토콜이다. 각 패킷에 대해 3번의 handshake 과정이 필요하고 각 패킷이 수신되었는지 아닌지를 확인하여 되지 않았을 시 재전송까지 하기에 중간에 정보가 변형될 가능성을 최소화한다. 하지만 그런 확인 과정을 모두 거치기에 오버헤드는 큰 편이고 이에 따라 이후 설명할 UDP보다 느린 특징을 보인다.

3.2.2 UDP(User Datagram Protocol)

UDP(User Datagram Protocol)은 RFC 768으로 정의되어 있으며[15] 인터넷 표준이다. 이 프로토콜은 TCP와는 달리 빠른 데이터의 전송을 목표로 하는 프로토콜이다. 클라이언트로부터 긍정 응답 메시지가 오지 않는 경우에도 특정 시간이 지나면 그 데이터에 대한 전송을 포기하며 헤더도 큰 편이 아니다. 그래서 TCP에 비해 빠른 전송을 할 수 있지만 데이터의 정확성을 보장할 수 없다는 것이 단점이다.

3.3 문서 파일에의 스트리밍 적용 방법 연구

3.3.1 목차 및 각 문단 개요의 빠른 추출 및 미리보기 지원

문서 파일들은 내부에 특정한 목차를 갖고 있기 마련이다. 이 알고리즘은 그 목차가 있는 경우 목차를 읽어오며, 없는 경우에는 문서를 분석하여 그 목차를 만들고 이를 서버가 클라이언트로 문서보다 먼저 전송하여 사용자가 원하는 곳으로 바로 이동할 수 있도록 하는 것을 목표로 한다.

문서 파일에서 트리를 구성해 목차를 만드는 기술은 이미 구현되어 있다. 이는 Microsoft Word에서 이미 사용되는 기술인데, 여기에 각 목차에 대한 개요를 보여줄 수 있도록 하여 사용자가 정확히 원하는 곳을 찾아갈 수 있도록 개선하였다. 그림7은 해당 기술을 윈도 탐색기에 적용했을 때 모습을 예상하여 나타낸 것이다.

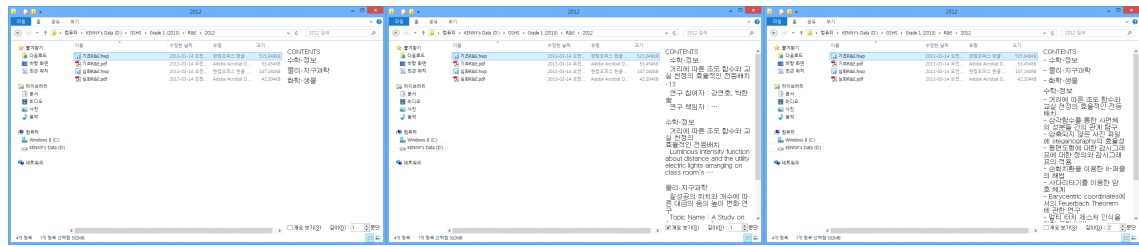


그림7 윈도 탐색기에 목차 및 개요 자동 제작을 사용했을 때

Fig.7 Using index and summary auto making at windows explorer

3.3.2 글과 미디어 파일과의 전송 방법 차별화

글과 미디어 파일을 통합한 문서 파일을 하나의 스트림으로 보고 스트리밍하는 경우에는 속도상에서의 그렇게 큰 장점이 없다. 그래서 제안하는 방법이 글과 미디어 파일과의 전송 방법을 다르게 하는 것인데, 문서 파일의 편집 시 중점 사항은 대부분 글이므로 글은 TCP 프로토콜 하에서 미리 모두 전송해 놓고, 각 미디어 파일들은 해당 페이지에 편집하려고 도달하기 직전부터 스트리밍을 통해 클라이언트로 전송하는 것이다.

이 방법을 사용하면 편집시 안 보는 부분에 대해서는 사진을 불러올 필요가 없으므로 사용하지 않았을 때에 비해 상당한 트래픽 절감이 있을 것으로 보인다.

IV. 결론

스마트폰과 다양한 태블릿으로 열린 모바일 오피스 시대에서 가장 중요한 것은 시간이다. 따라서 작성 및 편집까지 걸리는 시간을 최소화하기 위해 두 가지에 대해서 연구하였다. 첫 번째는 많은 문서 포맷들에 대해 서버 단에서 XML 등의 트리 분석을 통해 클라이언트에게 미리 목차와 개요를 포함한 미리보기를 제공하여 사용자가 빠르게 편집 부분까지 도달할 수 있도록 하는 것이고, 두 번째는 스트리밍시 글과 미디어 파일의 전송 방법을 각각 TCP를 통해서와 UDP를 통해서 전송하는 식으로 차별화를 두어 편집할 수 있는 글이 최대한 빨리 보일 수 있도록 하는 것이다. 이 기술들을 적용한 모바일 문서 편집기가 나온다면 언제 어디서나 문서 편집을 더욱 쉽게 할 수 있는, 즉 현재 사회가 추구하고 있는 유비쿼터스 시대에 한 발짝 더 다가갈 수 있을 것이다.

V. 참고문헌

- [1] 이광영 (2006) "스트리밍 기술을 이용한 전자문서의 서비스 비용 개선에 대한 연구", 공학석사 학위 청구논문, 서강대학교.
- [2] Logicplant (2012) "uZard Viewer", Retrieved November 20 2013 from Gyeonggi Science High School, Web site: <http://www.logicplant.com/ko/product/genericviewer.asp>
- [3] 한글과컴퓨터 (2010) "[공지] 한글 문서 파일형식 (hwp) 공개", Retrieved May 22 2013 from Gyeonggi Science High School, Web site: http://www.hancom.co.kr/notice.noticeView.do?targetRow=1¬ice_seqno=33
- [4] 한글과컴퓨터, 한글 문서 파일 구조 Hwp Document File Formats, 한글과컴퓨터, 2010, pp.3-100
- [5] 지식경제부 기술표준원 (2011) "KS X 6101", Retrieved November 10 2013 from Gyeonggi Science High School, Web site: http://standard.go.kr/code02/user/0B/03/SerKS_View.asp?ks_no=KSX6101
- [6] Microsoft (2009) "Microsoft Office File Format Documents", Retrieved May 16 2013 from Gyeonggi Science High School, Web site: <http://msdn.microsoft.com/en-us/library/cc313105.aspx>

- [7] Microsoft (2009) "Word, Excel, and PowerPoint Standards Support", Retrieved June 7 13 from Gyeonggi Science High School, Web site: [http://msdn.microsoft.com/en-us/library/cc313070\(v=office.12\).aspx](http://msdn.microsoft.com/en-us/library/cc313070(v=office.12).aspx)
- [8] Wikipedia contributors (2013) "Portable Document Format", Retrieved September 11 2013 from Gyeonggi Science High School, Web site: http://en.wikipedia.org/w/index.php?title=Portable_Document_Format&oldid=572181441
- [9] Adobe (2007) "Adobe PDF Reference Archives", Retrieved September 11 2013 from Gyeonggi Science High School, Web site: http://www.adobe.com/devnet/pdf/pdf_reference_archive.html
- [10] F. Yergeau (1998) "RFC 2279 - UTF-8, a transformation format of ISO 10646", Retrieved September 4 2013 from Gyeonggi Science High School, Web site: <http://datatracker.ietf.org/doc/rfc2279>
- [11] F. Yergeau (2003) "RFC 3629 - UTF-8, a transformation format of ISO 10646", Retrieved September 4 2013 from Gyeonggi Science High School, Web site: <http://datatracker.ietf.org/doc/rfc3629>
- [12] P. Hoffman (2000) "RFC 2781 - UTF-16, an encoding of ISO 10646", Retrieved September 4 2013 from Gyeonggi Science High School, Web site: <http://datatracker.ietf.org/doc/rfc2781>
- [13] Y. Shafranovich (2005) "RFC 4180 - Common Format and MIME Type for Comma-Separated Values (CSV) Files", Retrieved September 4 2013 from Gyeonggi Science High School, Web site: <http://datatracker.ietf.org/doc/rfc4180>
- [14] Marina del Rey (1981) "RFC 793 - Transmission Control Protocol", Retrieved September 18 2013 from Gyeonggi Science High School, Web site: <http://datatracker.ietf.org/doc/rfc793>
- [15] J. Postel (1980) "RFC 768 - User Datagram Protocol", Retrieved September 18 2013 from Gyeonggi Science High School, Web site: <http://datatracker.ietf.org/doc/rfc768>