
Fast 웹서비스를 위한 Fast XML 인코딩 시스템 구현

최봉규* · 조태범* · 정희경*

Implementation of the Fast XML Encoding System for Fast Web Services

Bong-Kyu Choi* · Tae-Beom Cho* · Hoe-Kyung Jung*

요 약

웹서비스는 서로 다른 플랫폼간의 통합을 가능하게 하였으나, 네트워크 환경이나 임베디드 시스템과 같이 상대적으로 느린 통신 매체에 자주 접속하거나 모바일과 같이 자원이 한정적인 소형 기기에서 사용 할 경우 전체 응용 프로그램의 성능을 저하시키는 문제가 발생하고 있다. 이에 ITU-T(International Telecommunication Union - Telecommunication)와 ISO(International Organization for Standardization)/ IEC(International Electrotechnical Commission)에서 공동으로 바이너리 XML 인코딩 표준을 제안하게 되었으며, 현재 진행 중인 바이너리 XML 인코딩 표준으로는 Fast Infoset과 Fast Schema가 있다.

본 논문에서는 웹서비스의 성능 향상을 위해 현재 표준화가 진행 중인 Fast Infoset 알고리즘과 Fast Schema 알고리즘을 도입하여 Fast XML 인코딩 시스템을 구현하였다. 또한, 구현된 Fast XML 인코더를 통하여 개발자나 사용자들이 Fast 웹서비스 시스템을 구축하는데 있어 참고 할 수 있는 테스트 베드를 제공한다.

ABSTRACT

Web Services came true a services integration, but the XML document decreases the efficiency of the whole application program with connects frequently in relatively slow communication media like network environment, embedded system or use the resource limited small-sized instrument like mobile. Thereupon, ITU-T and ISO/IEC suggested encoding standard of binary XML.

In this paper, implementation of Fast XML encoding system through introduction Fast Infoset algorithm and Fast Schema algorithm for web services increase performance. And, offered of test bed that build of Fast Web Services system through Fast XML Encoder.

키워드

XML, Web Services, Fast Web Services, Fast Infoset, Fast Schema

I. 서 론

서로 다른 환경에서 사용되는 어플리케이션들은 플랫폼과 프로그래밍 언어에 종속적이기 때문에 상호운용성이 결여되어 웹의 발전에 걸림돌로 작용하고 있다. 이에 웹에서의 상호운용성 확보를 위해 운영환경에 독

립적인 문서 형식을 정의할 수 있는 XML(Extensible Markup Language)을 기반으로 한 웹서비스(Web Services)가 등장하게 되었다.

웹서비스는 서로 다른 플랫폼간의 통합을 가능하게 하였으나, 텍스트 데이터로 XML 기반의 메시지를 주고 받기 때문에 네트워크 환경이나 임베디드 시스템과 같

이 느린 통신 매체에 자주 접속하거나 모바일과 같이 차원이 한정적인 소형 기기에서 사용 할 경우 전체 응용 프로그램의 성능을 저하시키는 요소로 작용할 수 있다.

현재까지 차원이 한정적인 환경에서의 XML 성능 문제는 Gzip과 같은 단순 압축 방식을 이용해 해결하고자 하였으나, XML에 적합하지 않은 압축 방식들은 XML 문서의 크기만 줄일 뿐 근본적인 성능 개선을 이루지 못하였다. 이에 ITU-T와 ISO/IEC에서 공동으로 바이너리 XML 인코딩 표준을 제안하게 되었으며, 바이너리 XML 인코딩 표준으로는 Fast Infoset과 Fast Schema가 있다 [1,2].

본 논문에서는 웹서비스의 성능 향상을 위해 Fast Infoset 알고리즘과 Fast Schema 알고리즘을 도입하여 Fast XML 인코딩 시스템을 구현하였다. 또한, 구현된 Fast XML 인코더를 통하여 개발자나 사용자들이 Fast 웹서비스 시스템을 구축하는데 있어 참고 할 수 있는 테스트 베드를 제공한다.

II. 관련연구

2.1 Fast 웹서비스 아키텍처

XML 기반 메시지는 현재 웹서비스 기술의 중심이고, SOAP(Simple Object Access Protocol) 메시지는 기존 프로토콜보다 큰 메시지를 가지며, 많은 계산을 필요로 한다[1,3,4].

이는 기존 방식보다 SOAP 메시지의 데이터 처리가 비효율적인 것을 의미한다. 이에 웹서비스를 보다 빠르게 처리하기 위해 XML의 바이너리 인코딩이 필요하게 되었으며, 바이너리 인코딩 기반의 Fast 웹서비스 시스템 구축이 요구되고 있다. 그림 1은 Fast 웹서비스에서 XML SOAP 메시지의 전송과 처리 흐름을 보인다.

Fast 웹서비스의 요점은 웹서비스에서 사용되는 XML 데이터의 크기를 줄여 전송과 처리에 소요되는 시간을 줄이는 것이다. 이를 위해 웹서비스의 전송 프로토콜인 SOAP 메시지와 전송되는 서비스를 기술하는 언어인 WSDL(Web Services Description Language)에 대한 효율적인 압축이 중요하다.

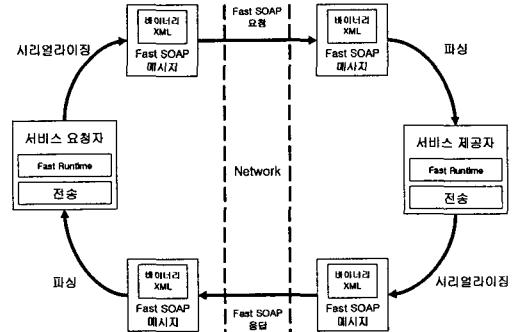


그림 1. Fast 웹서비스의 처리 흐름도
Fig. 1. Processing Pipeline of Fast Web Services

기존의 웹서비스에서는 서버와 클라이언트 간에 단순히 일반 XML 문서나 압축된 XML 문서로 메시지를 주고받았다. 그러나 XML 메시지는 전송량이 크다는 단점이 있고, 압축을 하더라도 압축과 해제 과정에서 발생하는 처리 시간으로 인해 웹서비스의 성능 향상에 걸림돌이 되었다. 하지만 Fast XML의 경우 문서의 크기는 줄어들면서 추가적인 압축과 해제의 과정을 필요로 하지 않으며, XML 문서와 마찬가지로 직접 파싱(Parsing) 및 시리얼라이징(Serializing)이 가능하다. 따라서 기존 웹서비스의 단점을 보완하고자 Fast 웹서비스에서는 Fast XML로 인코딩된 SOAP 메시지를 사용한다[5,6].

2.2 Fast Infoset

Fast Infoset은 XML Infoset에 기반한 인코딩 표준으로서 테이블과 인덱스를 사용하여 XML 데이터를 인코딩 한다[7,8]. Fast Infoset 알고리즘은 XML 정보셋에 따라 구분된 정보들을 테이블에 저장하여 인덱싱(Indexing) 한 후, XML 문서의 문자들을 인덱스로 대체함으로써 문서의 크기를 줄인다.

Fast Infoset 알고리즘을 이용한 바이너리 인코딩은 먼저 XML 문서를 파싱하여 XML 정보셋으로 변환하고, Fast Infoset 알고리즘에 따라 각각의 정보셋을 어휘(Vocabulary) 테이블에 저장하고 인덱스로 대체하여 Fast Infoset 테이터로 재구성한다[9,10].

2.3 Fast Schema

Fast Schema는 XML 문서의 Character 정보 항목을 제외한 모든 정보 항목을 Schema 문서로 처리하고 Character 정보 항목만을 인코딩하여 문서의 크기를 줄이는 방식이다. Fast Schema 알고리즘은 다음과 같은 특징을 가진다[11].

첫째, 높은 압축률로 Fast Schema 알고리즘은 XML 문서의 정보셋과 구조가 모두 Schema 문서에 포함되어 있어, Schema 문서의 특성상 미리 정의된 문서는 참조가 간편하고, 따로 전송할 필요가 없어 XML 문서의 압축에 효과적이다.

둘째, 구조 데이터의 최소화로 Schema 문서에는 Element 이름 등과 같이 대부분의 정보가 표현되어 있기 때문에 Fast Schema 문서에서 사용되는 구조 데이터는 몇몇 특수한 경우의 숫자 표현과 문자열의 길이 표현만으로 이루어져 있다.

셋째, 사용 제한으로 Fast Schema 알고리즘은 Schema 문서를 사용해 문서 구조를 처리하므로 Schema 구조가 미리 정의되어 있는 XML 문서만이 사용 가능하다.

III. 인코딩 시스템의 설계

Fast XML 인코딩 시스템은 사용되는 알고리즘에 따라 Fast Infoset 변환부와 Fast Schema 변환부로 분류하였으며, 각각의 알고리즘 처리를 위한 데이터를 제공하는 XML 파싱부를 두어 처리 과정을 통합하여 구성하였다. 그림 2는 Fast XML 인코딩 시스템 전체 구성도이다.

XML 파싱부의 모듈들은 XML 파싱 처리기, XML Infoset 처리기, XML Schema 처리기로 나누어지며, 원본 XML 문서의 파싱을 통해 각각의 변환부로 필요한 데이터를 넘겨주는 역할을 담당한다. Fast Infoset 변환부에는 XML 파싱 처리기와 XML Infoset 처리기를 이용해 XML 정보 항목과 프로퍼티를 전달하며, Fast Schema 변환부에는 XML 파싱 처리기와 XML Schema 처리기를 이용해 XML 문서 구조 정보를 제공한다.

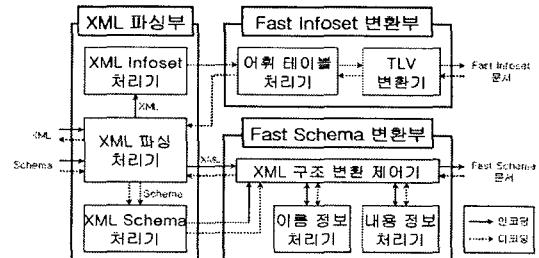


그림 2. 인코더 전체 구성도

Fig. 2. Diagram of Encoder

3.1 XML 파싱부

3.1.1 XML 파싱 처리기

XML 파싱 처리기는 문서를 통해 XML 문서를 파싱 및 시리얼라이징 하는 부분을 담당한다. 인코딩은 Fast Infoset 알고리즘과 Fast Schema 알고리즘에 맞는 데이터를 선별하는 XML Infoset 처리기와 XML Schema 처리기에 파싱된 XML 및 Schema 문서를 전달한다. 디코딩은 Fast Infoset 변환부와 Fast Schema 변환부로부터 파싱된 XML 데이터를 받아 XML 문서로 시리얼라이징 한다.

3.1.2 XML Infoset 처리기

XML Infoset 처리기는 XML Infoset 데이터를 분류하여 Fast Infoset 변환부의 어휘 테이블 처리기와 TLV(Tag, Length, Value) 변환기에서 사용할 수 있도록 제공한다. 인코딩은 파싱된 XML 데이터를 저장하는 XML Infoset을 생성하는 모듈로서 어휘 테이블 처리기와 TLV 변환기의 변환 흐름에 맞추어 Document 정보 항목, Entity 정보 항목, Notation 정보 항목, Element 정보 항목, Attribute 정보 항목, Namespace 정보 항목, Character 정보 항목, 기타 정보 항목 순으로 분류된다.

3.1.3 XML Schema 처리기

XML Schema 처리기는 파싱된 Schema 문서에서 구조 정보를 추출하여 Fast Schema 변환부의 XML 문서 구조 제어기에 전달한다. XML Schema 처리기는 직접 변환되는 데이터가 아닌 변환을 돋는 Schema 데이터의 처리를 담당하므로 인코딩, 디코딩 시 같은 역할을 수행한다.

3.2 Fast Infoset 변환부

Fast Infoset 변환부는 XML Infoset 데이터를 테이블에 저장하여 인덱스를 부여하는 어휘 테이블 처리기와 인덱싱 처리된 XML Infoset 데이터를 TLV 형식으로 바이너리 인코딩하여 Fast Infoset 문서를 만드는 TLV 변환기로 이루어져 있다. 인코딩은 XML 파싱 처리기, XML Infoset 처리기, 어휘 테이블 처리기, TLV 변환기 순으로 진행된다. 디코딩의 경우 인코딩과 반대의 순서로 진행되며, XML Infoset 처리기는 거치지 않는다.

3.2.1 어휘 테이블 처리기

어휘 테이블 처리기는 Fast Infoset에 정의된 10가지의 어휘 테이블을 관리하는 모듈로서 XML 문서의 중복되는 정보 항목들을 테이블에 저장하여 인덱스로 대체하는 역할을 담당한다. 그림 3은 Fast Infoset 알고리즘의 어휘 테이블이 긴 문자열 테이블, 짧은 문자열 테이블, 복합 테이블로 구성됨을 보여주고 있다.

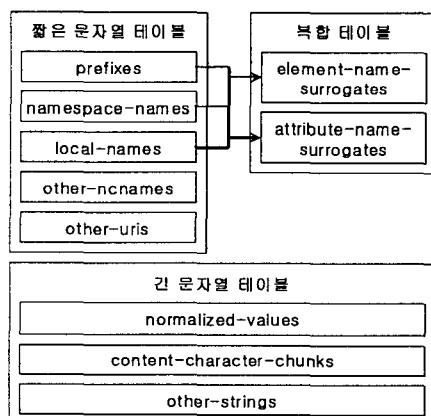


그림 3. 어휘 테이블 구성도
Fig. 3. Diagram of Vocabulary Table

3.2.2 TLV 변환기

Fast Infoset 알고리즘을 통해 변환된 XML 정보 항목은 태그, 길이, 내용의 순서로 구조화 된다. 그림 4는 XML Infoset이 Fast Infoset으로 변환되는 흐름을 보이며, XML Infoset의 변환 흐름은 XML 문서의 파싱 흐름과 일치하므로 XML 문서 구조를 따른다.

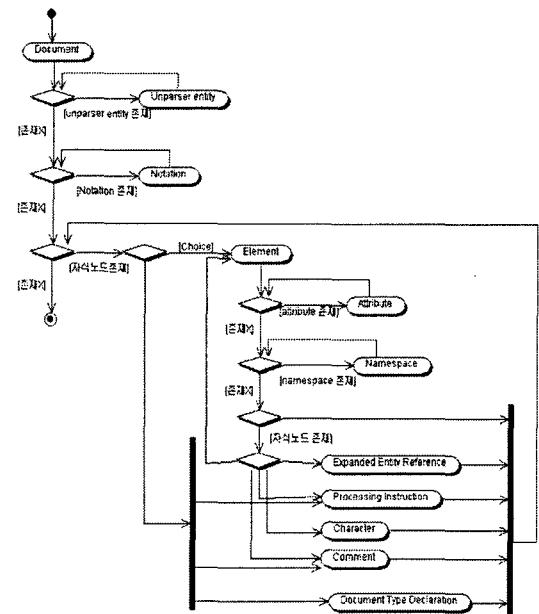


그림 4. XML 정보 항목 변환 흐름도
Fig. 4. Activity Diagram of XML Information Item

3.3 Fast Schema 변환부

Fast Schema 변환부는 파싱된 XML 데이터를 받아들여 문서 변환 처리하는 XML 구조 변환 제어기를 중심으로 이름 정보 처리기와 내용 정보 처리기가 이름 정보와 내용 정보 처리를 담당하는 형태로 구성된다.

Fast Schema 알고리즘은 Schema 문서를 파싱하여 얻어낸 XML 문서 구조를 기반으로 XML 문서를 인코딩 하므로 처리 흐름이 XML과 Schema로 나누어진다. 디코딩의 경우 Schema 문서의 파싱을 통해 XML 문서 구조를 얻어내는 과정은 동일하며, XML 문서의 변환과정은 인코딩과 반대로 Fast Schema 문서를 파싱하여 XML 문서를 생성한다.

3.3.1 XML 구조 변환 제어기

XML 구조 변환 제어기는 Fast Schema 변환부의 중심으로 이름 정보 처리기와 내용 정보 처리기를 호출하여 변환을 제어한다.

Fast XML 인코딩 시스템에서 테스트하고자 하는 XML 문서는 웹 서비스의 메시지 전송에 사용되는 SOAP과 WSDL 문서이다[12,13]. 그림 5는 XML 구조 변환 제어기의 SOAP 문서 변환 흐름도이다.

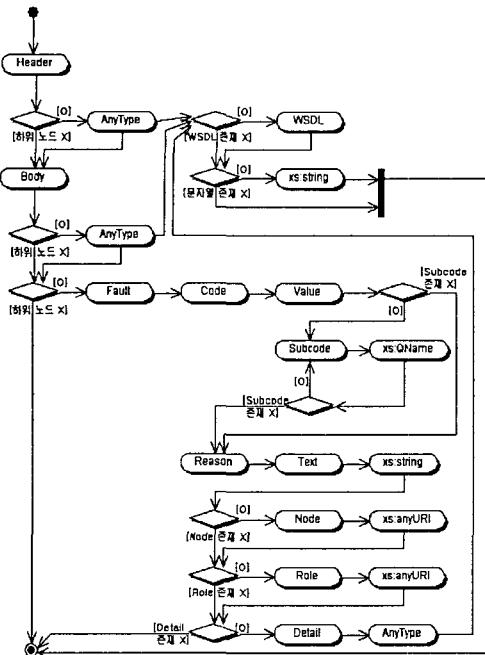


그림 5. SOAP 문서 변환 흐름도
Fig. 5. Activity Diagram of SOAP Document

3.3.2 이름 정보 처리기

이름 정보 처리기의 역할은 크게 이름 정보의 제거와 이름 정보를 대신할 구조 데이터의 생성으로 나눌 수 있다. XML 문서의 제공자와 요청자가 모두 Schema 문서를 가지고 있다면 XML의 데이터 구조를 표현하는 Element와 Attribute의 이름 정보는 필요가 없으며, 이름 정보를 제거함으로써 직접적인 내용 정보만을 전송할 수 있게 된다. 그러나 XML 문서에는 Schema 문서만으로는 알 수 없는 Sequence, Sequence-of, Choice, Enumeration 구조 정보가 있으며, 이 경우 Fast Schema 문서에 구조 데이터를 삽입하여 문서 구조를 유지한다.

3.3.3 내용 정보 처리기

내용 정보 처리기는 Character, Attribute Value 정보 항목 등 문서에서 담고자 하는 내용 정보를 처리한다. Fast Schema 알고리즘을 통해 변환된 XML 문서는 태그가 지워지고 Element, Attribute 이름이 제거되었으므로, 어느 곳이 내용의 시작이고 끝인지 알 수 없다. 그래서 Fast Schema 알고리즘에서는 내용 정보의 바로 앞에 길이 정보를 붙여 내용의 시작과 끝을 알 수 있도록 하였다.

IV. 인코딩 시스템의 구현

구현 환경은 Windows XP SP2 환경에서 JDK(Java Development Kit) 1.5를 사용하여 Java 언어로 구현하였다. 또한 Fast XML 인코딩 알고리즘들의 성능 분석 및 테스트를 위해 구현되었으며, XML 문서, Fast Infoset 문서, Fast Schema 문서들의 구조 및 내용 파악이 용이하도록 기능을 구현하는데 초점을 맞추었다.

4.1 메인 화면 구성

Fast XML 인코더는 파일 확장자에 따라 XML 문서, Fast Infoset 문서, Fast Schema 문서에 맞는 인코딩이 자동으로 진행된다. XML 문서일 경우 Fast Infoset 알고리즘과 Fast Schema 알고리즘에 따라 각각 변환되며, Fast Infoset 문서 또는 Fast Schema 문서일 경우 XML 문서로 디코딩 된 후 나머지 알고리즘에 따라 인코딩된다. 그럼 6은 변환을 위한 인터페이스를 구성하고 있는 메인 화면을 보인다. 메인 화면은 문서 구조를 파악하기 위한 XML Tree를 화면 왼쪽에 배치하였으며, XML 문서, Fast Infoset 문서, Fast Schema 문서 파일을 보고 저장할 수 있는 버튼들과 인코딩 성능을 비교하기 위한 결과표 부분, 그리고 변환 메시지를 보여주는 메시지 표시 부분으로 구성되어 있다.

4.2 Fast Infoset 화면 구성

Fast XML 인코더의 Fast Infoset 탭에서는 인코딩 된 "finf" 파일을 저장할 수 있으며, 트리를 통해 정보 항목

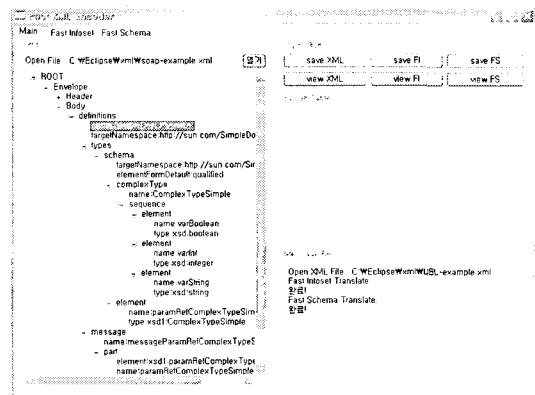


그림 6. 메인 인터페이스
Fig. 6. Main Interface

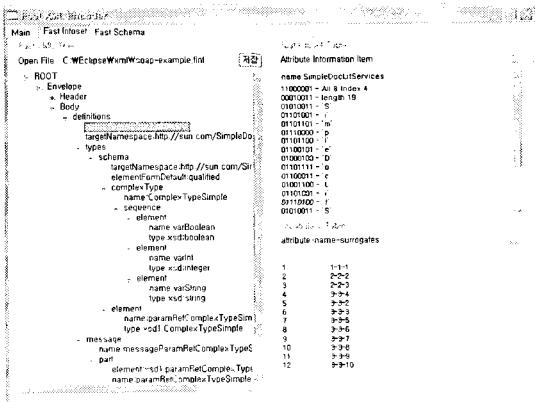


그림 7. Fast Infoset 인터페이스
Fig. 7. Fast Infoset Interface

을 선택하면 Fast Infoset 테이블 부분과 어휘 테이블 부분에 해당 정보 항목의 상세 정보를 보여주도록 구현하였다. 그림 7은 Fast Infoset 변환부 탭을 보여주고 있다.

Fast Infoset 테이블 부분은 변환된 Fast Infoset 문서의 바이너리 데이터를 비트 단위로 잘라서 정보 항목과 비교 설명한다. 어휘 테이블 부분은 정보 항목이 저장된 어휘 테이블을 보여주며, 콤보 박스를 통해 어휘 테이블 종류를 선택할 수 있다.

4.3 Fast Schema 화면 구성

Fast XML 인코더의 Fast Schema 탭에서는 인코딩 된 “mfs” 파일을 저장할 수 있으며, Fast Schema 테이블과 Schema로 변환 정보를 표시한다. 그림 8은 Fast Schema 변환부 탭을 보여주고 있다.

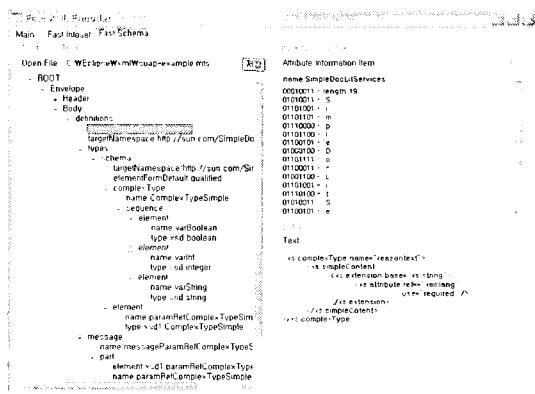


그림 8. Fast Schema 인터페이스
Fig. 8. Fast Schema Interface

XML Schema 테이블 부분은 변환된 Fast Schema 문서의 바이너리 데이터를 비트 단위로 정보 항목과 비교 설명한다. Schema 부분은 Schema 문서에서 선택된 정보 항목의 구조를 정의하고 있는 태입 부분을 텍스트로 나타낸다.

V. 결과 분석 및 고찰

테스트 내용은 MS .NET 기반의 웹서비스 제공자와 Sun JWSDP를 이용한 웹서비스 요청자간 Fast XML 문서의 상호운용 가능성 평가와 Fast XML 인코더의 성능 평가이다. 그림 9는 Fast 웹서비스의 구성도를 보인다 [14,15].

웹서비스 요청자가 웹서비스 제공자에게 SOAP 메시지로 원하는 정보를 요청하게 되면, 요청자 측에 위치한 Fast XML 인코더에서 XML 문서를 Fast XML 문서로 인코딩한다. 인코딩된 Fast XML 문서는 제공자에게 전송되어 제공자 측 Fast XML 인코더에 의해 XML 문서로 디코딩 한다.

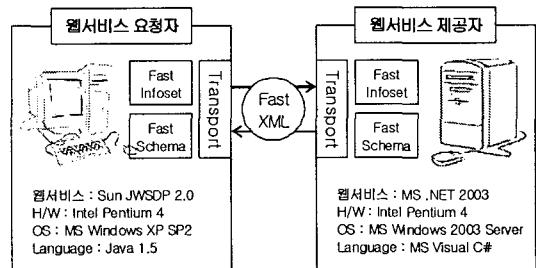


그림 9. Fast 웹서비스 구성도
Fig. 9. Diagram of Fast Web Services

5.2 고찰

본 논문에서는 신뢰할 수 있는 성능 평가를 위해 문서 크기와 정보 항목의 수를 고려하여 총 10개의 SOAP 문서를 테스트 하였다. 표 1은 SOAP 문서들의 인코딩 테스트 결과이다.

표 1. 정보 항목 수에 따른 인코딩 결과
Table 1. Encoding Result of Information Item Number

	정보 항목 수	XML 문서	Fast Infoset	Fast Schema
1	15	342	306	254
2	20	411	297	242
3	31	702	498	296
4	65	1,656	1,041	595
5	77	2,147	1,331	721
6	151	5,851	3,395	1,270
7	174	8,972	4,916	1,647
8	184	8,983	4,925	1,640
9	337	47,007	23,274	7,192
10	386	49,498	23,158	6,831

Fast XML 인코딩 테스트에서는 Fast 웹서비스를 위해 SOAP 문서만을 사용하므로 정보 항목의 종류는 거의 일정하다. 이는 SOAP 문서의 정보 항목수가 많아서 문서의 크기가 커지면, 정보 항목의 중복이 늘어나게 된다는 것을 의미한다. 그림 10은 표 1에서 정리한 테스트 결과의 비교 분석을 위한 그래프이다.

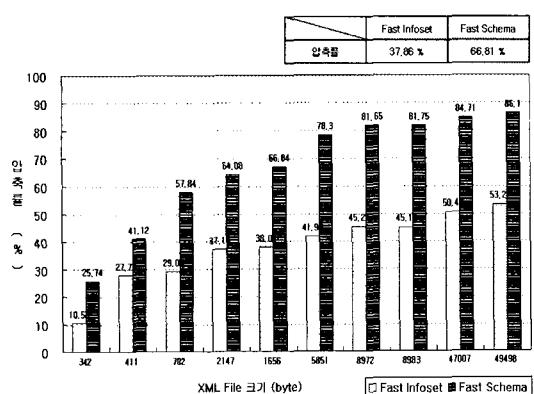


그림 10. 테스트 결과 그래프
Fig. 10. Grape of Test Result

테스트 결과를 보면 Fast XML 인코딩에서 Fast Infoset 방식과 Fast Schema 방식의 경우 평균적으로 각각 37.86%와 66.81%의 압축률을 보인다. 이는 Fast 웹서비-

스 구축을 위해 필요한 Fast XML 인코딩이 SOAP Schema에 의존적이므로 모든 XML 문서를 인코딩 할 수 있는 Fast Infoset 방식보다는 SOAP Schema에 국한시킨 Fast Schema 방식이 뛰어난 것으로 판단된다.

VI. 결 론

모바일이나 대용량 서버 등 모든 플랫폼의 어플리케이션을 통합하여 진정한 서비스 통합을 이루기 위해서는 웹서비스의 성능을 개선한 Fast 웹서비스의 구축이 필요하다. 그러나 사용되는 메시지인 XML 문서는 표현하고자 하는 부분의 자료가 너무 장황하기 때문에 부하가 큰 문제점을 가지며, 다양한 플랫폼 환경에서 Fast 웹서비스를 구축하기 위해서는 상호운용성이 확보되어야 하나 개발자들이 참조할 수 있는 가이드라인이 없어 큰 혼란이 예상된다.

이에 본 논문에서는 기존의 웹서비스에서 사용되던 XML 문서의 과부하 문제를 해결하기 위해 Fast Infoset 알고리즘과 Fast Schema 알고리즘을 도입하여 Fast XML 인코딩 시스템을 설계 및 구현하였다. 또한, 구현된 시스템을 이용한 테스트 베드를 구축하여 Fast 웹서비스의 상호운용 가능성을 보였다.

향후 연구과제로는 현재 구현된 Fast Schema 알고리즘을 SOAP과 WSDL 뿐만 아니라 Schema를 가지는 모든 XML 문서를 인코딩 가능하도록 확장시킬 필요가 있다.

참고문헌

- [1] 최일선 배재대학교 박사논문, “Fast Infoset 기반 Fast 웹서비스에 관한 연구”, 2004.
- [2] 이강찬, “웹서비스 기술 표준화 및 도입 이슈”, 2003.
- [3] 박종근, “[표준언어]Binary XML 표준화 이슈 및 동향”, 2005.5.
- [4] 변일수, 안창원, 박종근, 조희만, Desmons Frederic, 김성운, “바이너리 XML 기술 동향” 2005.8.
- [5] Patrick Cauldwell, Professional XML 웹서비스, Wrox, 2002.
- [6] XML Binary Characterization Working Group, <http://www.w3.org/XML/Binary/>

- [7] Fast Web Service, <http://java.sun.com/developer/technicalArticles/WebServices/fastWS/>
- [8] ITU-T Rec. X.891 (2005) | ISO/IEC 24824-1, Fast InfoSet
- [9] Fast InfoSet@java.net, <http://www.idealliance.org/proceedings/xtech05/papers/04-01-01/>
- [10] Paul Sasndoz, Alessandro Triglia, Santiego, Pericas-Geertsen, Fast InfoSet, <http://java.sun.com/developer/technicalArticles/xml/fastInfoSet/>
- [11] ITU-T Rec. X.892 (2005) | ISO/IEC 24824-2, Fast Web Services
- [12] SOAP Version 1.2 Part 1: Messaging Framework, <http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>
- [13] Web Services Description Language(WSDL), <http://www.w3.org/2002/ws/desc/>
- [14] 정희경외 5인, “웹서비스 상호 운용성을 위한 테스트 프레임워크 및 시뮬레이션” 한국해양정보통신학회, Vol.8 No.2 pp.941~944, 2004.10.
- [15] 유윤식 배재대학교 석사논문, “다중 어플리케이션을 위한 웹서비스 상호 운용성에 관한 연구”, 2004.

저자소개**최 봉 규(Bong-Kyu Choi)**

1989년 영남대학교 전자공학과
(공학사)
2001년 고려대학교 경영정보학과
(MBA)

2007년~현재 배재대학교 컴퓨터공학과 박사과정
현재 삼성전기 상품설계 부장
※ 관심분야 : 홈네트워크 용 Mobile 제어 프로그램,
Digital Electrical Simulation, Ubiquitous Communication,
CAE

**조 태 범(Tae-Beom Cho)**

1999년 배재대학교 정보통신공학과
(공학사)
2003년 배재대학교 정보통신공학과
(공학석사)

2005년~현재 배재대학교 컴퓨터공학과 박사과정
2003년~현재 배재대학교 IT교육센터 전임강사
※ 관심분야 : XML, 웹서비스, 멀티미디어, 데이터베이스

**정희경(Hoe-Kyung Jung)**

1985년 광운대학교 컴퓨터공학과
(공학사)
1987년 광운대학교 컴퓨터공학과
(공학석사)

1993년 광운대학교 컴퓨터공학과(공학박사)
1994년~현재 배재대학교 컴퓨터공학과 교수
※ 관심분야 : 멀티미디어 문서정보처리, XML, SVG, Web Services, Semantic Web, MPEG-21, Ubiquitous Computing, USN