

# XML 프로토콜의 효율적 메시지 전달방식

최 현동\*, 고찬\*\*

\* 서울산업대학교 컴퓨터공학  
139-743 서울시 노원구 공릉길 138(공릉2동 172번지)  
Tel:010-8373-3867,choihd1202@naver.com  
\*\* 서울산업대학교 컴퓨터공학  
139-743 서울시 노원구 공릉길 138(공릉2동 172번지)  
Tel: 010-9198-6705, chankoh@snut.ac.kr

## 제1장 서론

최근 공공부분에서 XML표준문서를 전송할 수 있는 XML기반의 메시지 전송 프로토콜(XML Message Protocol)에 대한 수요가 증가하고 있다. 이러한 수요를 충족하는 대표적 기술로 고전적인 형식의 XML-RPC(eXtended Markup Language Remote Procedure Call)와 SOAP(Simple Object Access Protocol)이 사용되고 있다

XML-RPC는 적용형식이 간결하고 이 기종간의 높은 이식성을 제공하며 또한 문서에 정의된 이름과 값이 매치된 형태로 에러 발생시 디버깅과 오류 발견이 용이하여 프로그램 수정이 쉽고 유지보수성이 탁월하다. 그러나 이 방법은 사용자 정의 데이터, 수신지정, 메시지에 따른 프로시저 제어 등이 복잡한 관계로 초급자가 사용하기에 부담이 많다. SOAP은 특정 플랫폼에 독립적이며 상호 운영성과 확장성이 높으나, UDDI, WSDL을 포함해야 하고 큰 XML문서를 전송 시 TAG정보도 함께 크므로 많은 정보자원 점유와 성능저하 문제가 발생한다. 본 연구에서는, 이러한 전송의 문제점들을 해결하도록 효율적인 XML메시지 전달방식을 제안하고, Web Service 구축의 예를 들어 실험을 하였다. 실험을 통하여 제안된 방법의 유효성을 입증 하였다.

## 제2장 XML의 개요

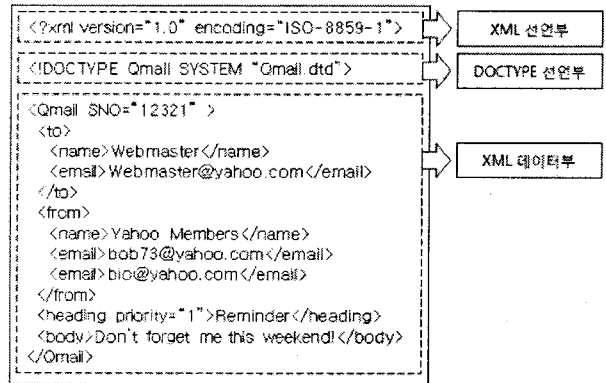
### 2.1 XML의 구성

1998년 W3C(World Wide Web Consortium)의 권고안으로 제안된 XML은 HTML과 달리 데이터 의미를 기술할 수 있는 메타언어의 특성을 가지고 있으며 SGML보다 매우 사용이 간단하고 편리하게 구성되었다. 이런 XML은 현재 많은 응용분야에서 사용되고 있으며 그 사용범위는 더욱더 확장될 것으로 예상된다.

### 2.2 XML 구성요소

XML문서의 구조는 XML선언부(XML declaration), DOCTYPE선언부, XML 데이터부로 구성되며 마크업(Markup)으로 표현되는 논리적 구조와 저장개체인 엔티티(entity)로 불리는 물리적 구조로 구성된다.

XML문서의 구성요소는 엘리먼트(Element), 속성(Attribute), 처리명령(Processing Instruction), 엔티티(Entity), 주석(Comment), CDATA 섹션(CDATA Section) 등으로 구성된다. 이를 요약하면 다음의 표와 같다.



구성요소	설명
엘리먼트(Element)	마크업 태그와 그 안에 포함된 내용
빈 엘리먼트 (Empty Element)	데이터와 같은 내용을 포함하지 않은 엘리먼트
루트 엘리먼트 (Root Element)	문서 내 모든 엘리먼트와 내용을 포함하고 있는 XML 문서의 엘리먼트
속성 (Attribute)	엘리먼트에 포함되어 추가적인 정보를 제공
엔티티 (Entity)	텍스트, Binary 데이터, 비 ASCII 문자를 저장하는데 사용
처리명령(Processing Instruction)	전체 문서나 문서의 일부를 처리하는 응용 프로그램과 연결해 주는 명령어
주석 (Comment)	XML 프로세서가 해석하지 않는 설명문
CDATA 섹션 (CDATA Section)	특수한 문자를 일반 텍스트로 인식하도록 하는 표기법

DOCTYPE 선언부에는 XML문서에 사용될 구성요소를 정의해 놓은 DTD(Document Type Definition)문서를 따로 링크를 통해 관리하고 있다. 이것은 데이터 베이스 테이블 스키마정의와 같은 것으로 XML내부의 엘리먼트는 DTD에 의해 유효성을 검사할 수 있어서 모든 요소가 유효성 검사를 통과한 문서에 대해 Well-Formed(문법에 적격인)한 문서라 하여 XML문서는 Well-Formed해야 이용할 수 있는 문서가 된다.

### 2.3 SOAP와 관련기술

#### 2.3.1 SOAP(Simple Object Access Protocol)

플랫폼과 상관없이 분산환경에서 XML과 HTTP를 사용하여 서비스, 객체 및 서버에 액세스 할 수

있는 통신 규약으로 HTTP과 SMTP와 같은 다양한 인터넷 표준 통신 프로토콜 이용하여 XML Message Protocol 이다, SOAP은 서버의 주소와 객체의 이름, 그리고 전달되는 파라미터들만 명기하면 클라이언트 운영체제나 환경에 상관없이 언제라도 객체를 사용할 수 있다는 장점이 있다. XML 이 애플리케이션 간에 전달되는 데이터를 표준화했다면 SOAP 은 서버의 객체를 호출하는 RPC 규약을 표준화 한 것이다

**2.3.2 WSDL(Web Service Description Language)**  
 웹 서비스를 기술하는 일종의 스크립트언어로 XML 포맷으로 구성되고 HTTP를 통하여 전달될 수 있으며 다른 개념으로 인터페이스를 정의하는 IDL(Interface Definition Language)에 해당한다. A라는 서비스가 가지는 메소드와 속성으로 어떤 인수를 호출하고 무슨 방식의 리턴 값을 제공하는 지를 알려주는 것이다. 정의된 요소 각각의 의미는 다음과 같다

**2.3.4 UDDI(Universal Description, Discovery and Integration)**  
 웹 서비스 관련 정보의 공개와 탐색을 위한 표준이다. 서비스 제공자는 UDDI라는 서비스 소비자에게 이미 알려진 온라인 저장소에 그들이 제공하는 서비스들을 저장하게 되고, 서비스 소비자들은 그 저장소에 접근함으로써 원하는 서비스들의 목록을 찾을 수 있게 된다.UDDI 비즈니스 등록은 다음과 같이 세 가지 구성요소를 갖는다.

- 화이트 페이지(White Pages):주소, 연락처 등의 알려져 있는 식별자
- 옐로페이지(Yellow Pages):표준 분류법을 기반으로 한 산업 분류
- 그린페이지(Green Pages):비즈니스를 통해 노출된 서비스에 대한 기술 정보

**2.4 XML-RPC**  
 RPC프로토콜의 일종으로서, 인코딩 형식은 XML을 채택하고, 전송 프로토콜은 HTTP을 활용하면서, 전송규약은 매우 단순하며, Data형식과 명령어가 다른 XML-Message 통신보다 매우 적고, 단순한 편이다. 가장 큰 특징은 다른RPC PROTOCOL 보다 많은 규격설정과 실제 프로그램 적용 시 많은 양의 코딩을 요구하지만 XML-RPC은 매우 적은 노력을 요구하고, 보통 SOAP 보다 단순하면서, 일정한 지식만 있다면 효율적인 Web Service에 적용하기 쉽다.

**제3장 문제점 분석**

**3.1 XML 문제점**

(1) SOAP는 Encoding Length size가 크다  
 문제점 하나 SOAP은 다른 유사한 원격객체(RMI,

CORBA 등)호출방식 보다 전송되는 데이터의 페이로드(payload)가 크다. 예를 들어 동일한 메소드를 호출 할 경우 RMI는 Encoding Length 40 bytes 이며 SOAP는 Encoding Length 368 bytes가 9배 정도 메시지의 사이즈가 커졌다, 그 이유는 서비스 메소드명, 서비스 파라미터 형식들과 이에 대한 입력 값들 그리고 메시지 내에 기술된 자원정보를 식별하기 위한 URI와 다양한 네임스페이스(Namespace)등을 기술하기 때문이다

**(2) SOAP는 대기시간이 길다**

Apache Axis, Microsoft SOAP, CORBA 그리고 RMI 의 특정 원격 메소드 호출 후 응답 메시지 도착까지의 대기시간(laytency)를 측정한 비교결과다. 이 결과에서 볼 수 있듯이 SOAP을 이용하는 Apache Axis 와 Microsoft SOAP가 CORBA 및 RMI 보다 훨씬 대기시간이 길게 나타나고 있다.



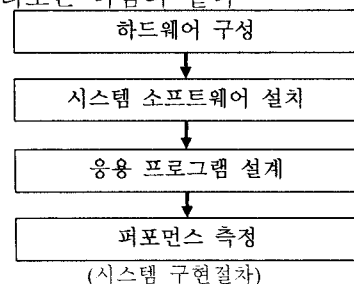
(출처) Davis, et.al., Latency Performance of SOAP Implementations, IEEE Cluster Computing and the Grid 2002

**3.2 문제점 원인 분석**

처음으로 CORBA 나 RMI 는 상호간의 미리 보안정책, 접속방법, 통신방식과 같은 세부사항들이 직접 협의가 되어 특정 사용자만이 통신을 하기 때문에 서비스 대기시간이 상대적으로 짧고, SOAP은 웹 상의 불특정 다수를 타겟을 대상으로 서비스가 운용이 되기 때문에 상대적으로 대기시간이 길다.두 번째로, CORBA 및 RMI는 서비스 클라이언트와 서버간의 4계층인 전송 레벨에서 접속되지만, SOAP는 7계층인 응용레벨에서 상호간의 연결이 되기 때문이며, 이것 이외 많은 단점이 있다.

**제4장 XML기반의 메시지 전송 프로토콜 퍼포먼스 측정**

현재 SOAP과 XML-RPC 기반의 Web Service 퍼포먼스 측정이 진행되고 있으며 실험에 대한 테스트 시나리오는 다음과 같다

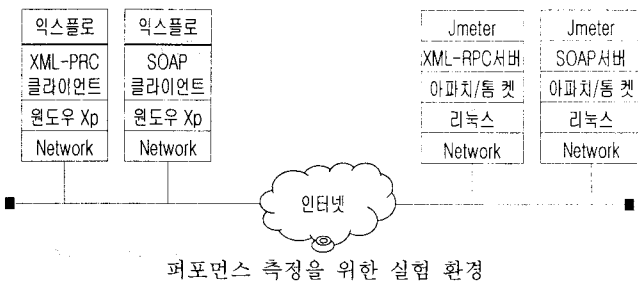


#### 4.1 하드웨어 구성 및 시험측정 소프트웨어 설치

##### 4.1.1 전제조건

구분	내용	
네트워크	네트워크 통신지연이 실험 결과에 영향을 미칠 수 있으므로 모든 컴퓨터는 동일한 네트워크 세그먼트(Segment)안에 구성한다	
하드웨어	실험 대상 컴퓨터 자원(CPU, Memory, RAM, LAN Card 등)등의 성능에 의하여 실험결과가 달라 질 수 있으므로, 모든 실험 대상 컴퓨터는 동일 사양의 컴퓨터를 사용한다	
소프트웨어	운영체제	운영체제는 종류에 따라 실험결과가 달라 질 수 있으므로, 모든 실험 대상 운영체제는 동일 사양의 운영체제를 사용한다
	테스트 툴	Web Service기반의 부하와 성능을 테스트하고 결과를 리포트 할 수 있는 Apache JMeter Tool을 이용예정임
	개발언어	Java
기타	실험 대상 컴퓨터에서 실험에 필요하지 않은 모든 프로그램 및 프로세스는 제거 함	

##### 4.1.2 시험측정 시스템 구성도

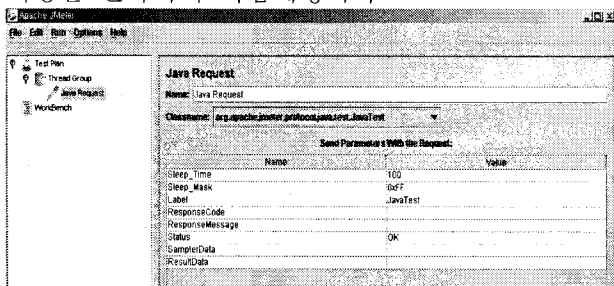


#### 4.2 SOAP & XML-RPC 퍼포먼스 측정용 프로그램 설계

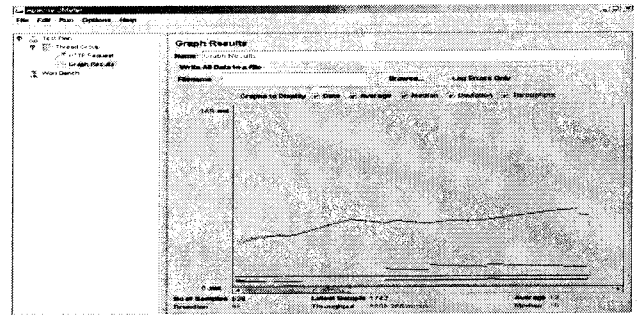
JAVA 타입	SOAP 타입	XML-RPC 타입	비고
Int	xsd:int	Int	정수 4 byte
double	xsd:double	double	실수 (8 byte)
String	xsd:string	String	'1234567890'
Date	xsd:dateTime	dateTime.iso8601	Date Type

#### 4.3 측정 대상 프로그램 구현

JMeter툴를 활용하여 다음과 같이 SOAP와 XML-RPC를 단위테스터를 할 예정이며 최소 10회 이상을 반복하여 시험예정이다

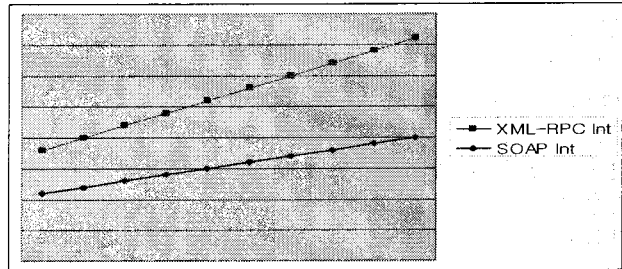


다음과 같은 JMeter화면에서 그래픽과 Data를 추출하여 결과분석 자료로 활용할 예정이다

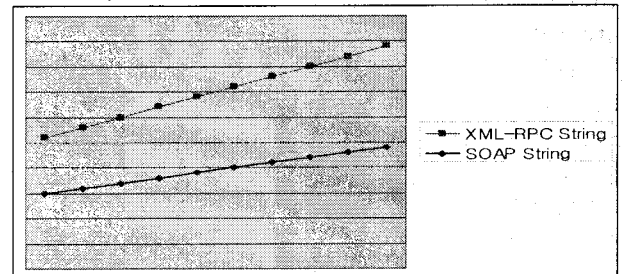


#### 4.3.1 SOAP와 XML-RPC의 정성적인 결과비교(실험후 작성)

##### (1) int Data type 측정결과



##### (2) String Data type 측정결과



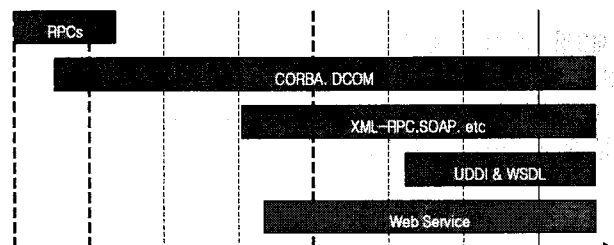
#### 제5장 시험 결과 분석

##### 5.1 정성적인 결과비교

현재 시험 중 관계로 결과를 예측하기는 어렵지만 경험적인 측면에서 예상을 한다면 Web Service구축 및 운영에서 많은 문제가 존재하지만 그 중에 하나가 퍼포먼스에서 많은 장애요인으로 인식하고 있고, 많은 운용자와 관리자는 이문제 해결을 위하여 노력하고 있다

##### 5.2 경험적인 측면의 비교

SOAP(Simple Object Access Protocol)이 web Service에 대표적으로 사용되고 있다, 그러나 XML-RPC는 많은 장점에도 불구하고 시대에 뒤떨어진다는 이유로 외면을 당하면서 환영을 받지 못하고 있으며, Web Service는 SOAP의 단점을 극복하지 못 하였으며, Web Service는 SOAP라는 관념을 극복하지 못하고 있다,



그림에서 같이 XML-RPC가 SOAP보다 뒤떨  
어지진 않았다고 생각한다. SOAP 서비스가 web  
Service를 위한 유일한 대안이라고 생각해서는  
안되며, 분명 WSDL 파일을 사용해 상호작용을  
표현하며 SOAP아 아닌 XML-RPC방법을  
사용하여 클라이언트와 상호작용하는 프로그램을  
개발할 수도 있다.

[참고문헌]

1. 자바와 XML, 2005
2. 자바 툴을 이용한 eXtreme Programming
3. XML-RPC 프로그래밍(자바, 펄, PHP, ASP를  
이용한)
4. XML-RPC(Programming Web Services with), 2001
5. Professional XML Web Services, 2002
6. Professional XML Database, 2001
7. Professional Java XML, 2002
8. 분산 환경에서 메시지 교환을 위한 SOAP운영  
플랫폼간의 성능비교, 서울사업대, 마경근