

## J2EE 1.4를 이용한 XML-Aware Network 구현 방안

여창훈<sup>0</sup>

고려대학교 컴퓨터학과  
duckdgn5<sup>0</sup>@hotmail.com

### Implementation methodology of XML-Aware Network used J2EE 1.4

Chang-hoon Yeo<sup>0</sup>

Dept. of Computer Science and Engineering, Korea University

#### 요 약

오늘날의 웹 환경이 웹 서비스 혹은 분산 컴퓨팅 환경으로 변화하여 가고 있다. 이 웹 서비스 환경을 위해 구조화된 문서로 사용할 수 있는 XML을 기반으로 활발히 개발이 이루어 지고 있고 또한 관련 기술들이 소개 되고 있다. 그러나 XML은 오늘날의 웹 환경과 같은 Content-aware network 방식이 아니라 새로운 encoding-data 방식이기 때문에 보안에 아주 취약하다는 문제점을 낳고 있다. 특히 XML은 Network의 전송을 위해 SOAP 메시지를 사용하게 되는데 SOAP 메시지는 방화벽을 그대로 통과하기 때문에 해커들의 주 사용 도구가 될 수가 있다. 본 논문에서는 XML의 보안 측면을 보다 강화하기 위해 최근 화두로 떠오르고 있는 XML-Aware Network에 대해서 소개하고 J2EE 1.4 에 포함된 API를 사용하여 XML Aware Networking을 위한 구현 방안을 제시 하는데 초점을 맞추고 있다.

#### 1. 서 론

XML[1](wXtensible Markup Language)은 1996년 W3C (World Wide Web Consortitium)에서 처음 제안한 것으로써, 웹상에서 구조화 된 문서를 활용 할 수 있도록 설계된 표준화 된 텍스트 형식이다. 1998년 2월에 W3C에서 표준으로 확정된 상태이며, 많은 곳에서 XML을 기반으로 활발한 개발이 이루어지고 있고 또한, 관련 기술들이 소개 되고 있다.[1] XML은 문서의 내용과 디자인이 완전히 분리 되어 있고 XML문서를 통해서 트리구조의 구조적인 형태로 다양한 방식의 링크 기능을 제공한다. 또한 표준화된 DTD를 통해서 모든 분야에서 사용이 가능하다. 요즘 많이 사용하는 WML, SMIL, SVG등이 모두 XML에 기반을 둔 언어이다. 이것은 XML의 DTD를 선언하고 모든 사용자가 그 규칙을 맞추어 사용하면 컴퓨터나 사람이 모두 해석 할 수 있는 메타언어가 될 수 있다. 초기의 웹 방식 그리고 현재의 웹 방식은 Content-aware network 방식 즉 Http aware network라고 한다. 페이지의 요청을 개별적인 웹 서버로 로드 발란싱(Load balancing)을 해주고 이때 로드 발란싱 기법(URL switch, cookie, redirection, etc..)에 대한 방식도 점점 다양해지고 있다. 그러나 차세대 인터넷 망에서는 웹 서비스가 활발해 지고 기존의 HTTP에 SOAP과 XML-RPC가 적용되기 때문에 더 이상 기존의 환경을 사용 하기에는 보안 측면에서 많은 문제점들을 낳고 있다. XML자체가 Contents가 아닌 새로운 encoding-data이며 새로운 프로토콜이기 때문이다. XML의 SOAP 메시지는 모든 firewall을 통과 한다는 단점을 가지고 있다. 그렇기 때문에 XML-Aware Network가 최근

큰 화두가 되기 시작하였다. 본 논문의 구성은 2장에서 차세대에서 거의 모든 컴퓨터 시스템에서 사용하게 되는 XML의 문제점을 알아보고 3장에서 XML-Aware Networking[3]을 간략히 소개하고 4장에서는 J2EE 1.4에 포함된 API를 사용한 Aware Networking구현 방안을 제시 하였고 5장에서 향후 및 결론을 논하였다.

#### 2. XML의 문제점

##### 2.1. XML Performance Challenge

XML은 text-based self-tagging format라고 불리어 진다. 이는 같은 메시지라고 바이너리 형태에 비해서 20배 정도가 커지기 때문이다.

예) 0xA13FF51301 [5byte]  
 <?xml version="1.0"?><invoice><no><1001></no><product><sku>150591501</sku></product></invoice> [95byte]

[그림1] XML byte 예제

뿐만 아니라 XPath, XSLT, XML Security, SOAP 등의 복잡한 XML 처리 모델이 요구 된다. 또한 XML 성능을 높이기 위해 XML/SOAP을 자제하거나 직접 개발자가 특정 환경의 상황에 맞게 응용 프로그램의 Hand-Tune을 해야 할 뿐 아니라 결과에 비해 시간이 많이 걸린다.[2]

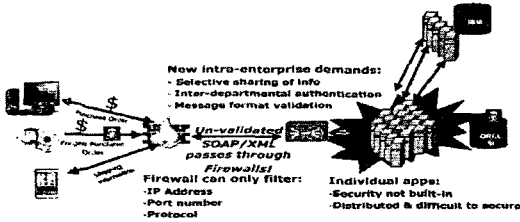
Benchmark Run	TPS	Latency (mSec)
Direct XML to browser	2060	23
JSP, no XML	1250	97
Server-Servlet, XALAN	197	493
Server-Servlet, XSLTC	274	182

[그림2] 간단한 XML-HTML benchmark result

##### 2.2. XML Security Challenge

XML에서 자주 사용하게 되는 SOAP 메시지는 기존의 네

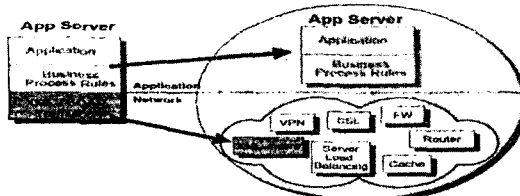
트위크의 보완 구조와 관계없이 네트워크를 통과하게 된다.



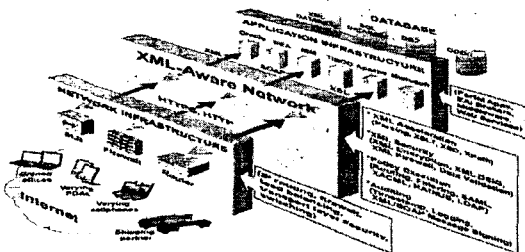
[그림3] SOAP 메시지를 이용한 해킹 시나리오  
 차세대 인터넷은 앞으로 XML을 사용한 웹 서비스가 활발히 이루어지고 있기 때문에 해커들이 SOAP 메시지를 이용하여 해당 DB로 들어와서 자료를 마음껏 빼가는 것이 가능하게 된다. 뿐만 아니라 방화벽의 정책이나 각 서버의 Application을 다운 시키거나 xDOS 공격, Snooping 공격에 대한 대책이 없는 상황이다.

### 3. XML Aware Network

기존의 웹 서비스 혹은 XML의 SOAP 메시지에 대한 보안이 강력히 대두 되면서 기존의 환경에 대한 새로운 접근 방식이 필요하다. 새로운 환경에서는 네트워크 상에서 XML을 인식 가능한 하드웨어 혹은 소프트웨어가 필요할 것으로 보이며 방화벽을 통하여 접속한 메시지들 체크하여 보다 안정화된 네트워크 환경이 되어야 한다.



[그림4] XML Aware Network 프로토콜

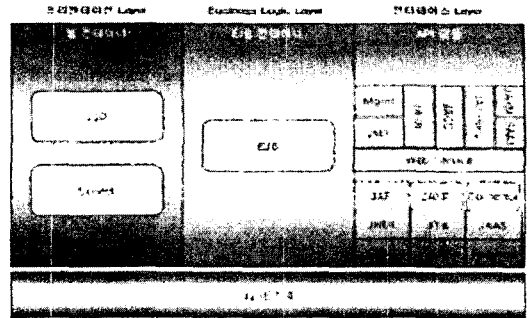


[그림5] XML-Aware Network 아키텍처

그림 5와 같이 XML Aware Network은 일종의 방화벽 역할을 하는 디바이스를 두게 되면 XSLT, XSD, XPath 등을 파싱 하는 역할 뿐 아니라 방화벽, Date 유효성 검증 및 라우팅, Filtering, SAML 정책 실행 등의 역할을 맡는 하드웨어 혹은 소프트웨어 적인 역할과 XML Acceleration 역할까지 가능하게 된다.

### 4. J2EE API를 이용한 XML-Aware Network 방안

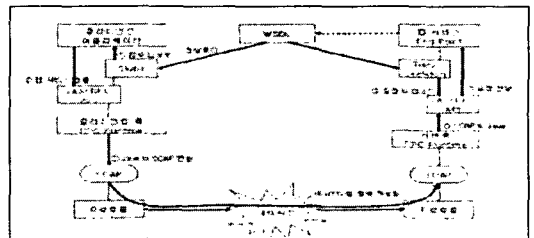
### 4.1. J2EE 플랫폼 아키텍처



[그림6] J2EE 플랫폼 아키텍처

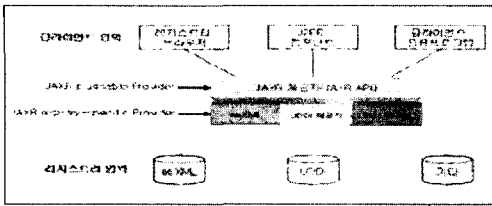
그림6은 J2EE 1.4를 기반으로 한 IBM WebSphere Application Server 5.1의 플랫폼이다. 가장 눈에 띄는 변화는 웹 서비스의 통합에 있다. 관리 모듈인 Java Management, JMX(Java Management eXtension)가 있고 XML 처리를 위한 JAXP(Java API for XML Registries)과 웹 서비스 프로토콜을 지원을 위한 JAX-RPC(Java API for XML-based RPC), SAAJ(SOAP with Attachments API for Java), JAXM(Java API for XML Messaging), 그리고 보안 모듈인 JACC(Java Authorization Contract for Container)등이 있다. 또한 모듈을 통합하기 위해 웹 서비스 모듈이 제공되고 있다.

### 4.2. JAX-RPC 아키텍처 및 JAXR 구조도



[그림7] JAX-RPC 아키텍처

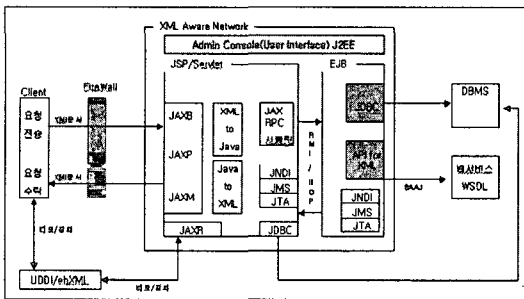
JAX-RPC[4] 기반 웹 서비스의 자세한 구조도를 그림7에서 보여주고 있다. WSDL을 통해 만들어진 Stub를 임포트 하여 웹 서비스 호출을 위한 준비를 한다. 클라이언트는 JAX-RPC를 이용하여 임포트 한 웹 서비스 메소드를 호출 한다. JAX-RPC를 통한 서비스 요청은 RPC Runtime을 거치면서 XML 문서로 변환되고 이는 다시 SOAP 메시지로 구성되어 진다. SOAP 메시지는 일반 표준 프로토콜인 HTTP를 통해 서버 측에 전달되며 서버 측 RPC Runtime은 SOAP 메시지를 받아 Java로 매핑 한다. JAX-RPC는 웹 서비스 skeleton을 참조하여 자바 매핑을 해당 메소드로 매핑 한다.



[그림8] JAXR 구조도

JAXR[4]은 XML기반 레지스트인 UDDI, ebXML등과 소통하기 위한 표준 API이다. JAXR을 통해 개발자는 모든 XML기반 레지스트리에 접근 할 수 있으며 코드 변경 없이 투명하게 레지스트리 제공자를 교체 할 수 있다. 이제 어플리케이션 서버들은 자체 내부에 JAXR과 XML 레지트m트리를 지니게 되고 이는 개발자들이 어플리케이션 서버의 레지스트리에 웹 서비스를 저장 할 수 있고 이를 외부 사용자에게 서비스 할 수 있게 해준다. 또한 J2EE 1.4는 JSR이라는 규약을 만들어 J2EE 프로그램 모델과 융합을 시키고 있기도 하다.

#### 4.3. J2EE 1.4를 이용한 XML-Aware Network 구현 방안



[그림9] XML-Aware Network 구현방안

그림 9는 J2EE 1.4 기반인 IBM WebSphere Application 5.1을 응용하여 XML-Aware Network을 구현하였다. Client가 최초 request를 보내면 UDDI/ebXML에서 해당 요청의 질의 문을 JAXR로 보내지게 된다. JAXR은 그림 8에서와 같이 모든 레지스트리에 접근 가능하며 코드 변경 없이 해당 질의에 대한 응답을 Client에게 다시 보내 request를 Server측에 전송을 하게 된다. 물론 해당 질의가 Http이면 FireWall의 정책을 통해 해당 JSP/Servlet을 통해 결과를 가지고 오게 된다. 그러나 XML/SOAP 문서일 경우에는 JAXB(Java Architecture for XML Binding)을 통하여 JAXM((Java API for XML Mesaging)과 JAXP(Java API for Precessing)을 통하여 메세징 및 프로세싱 처리가 가능하게 된다. 또한 XML Filtering을 위한 JSTL(Java Standard Tag Library)를 통하여 파싱 및 프로세싱이 가능하며 SAX[7]를 통하여 XFilter구현

또한 가능 하게 된다. 이렇게 XML로 처리된 문서에 대해 바인딩, 프로세싱, 필터를 구현하여 해당 XML/SOAP에 대한 방화벽의 정책 구성이 가능하게 되며 이는 Java로 변환하여 해당 요청을 처리 할 수 있게 되고 응답을 XML 문서로 다시 변환하여 보낼 수 가 있게 된다. 또한 2차적으로 EJB Container를 통하여 SAAJ를 이용하기 때문에 추가적인 보안 역시 가능 하게 된다. 또한 WAS 제품별로 Performance Mornitoring을 지원해 주기 때문에 해당 JSP/Servlet 등의 Performance등의 측정이 모두 가능하며 실시간으로 Clustering 및 HA기능도 가능할 뿐 아니라 일반 유저가 사용할 때 어떠한 사용자가 현재 접속 중이며 어떠한 자원에 접근 하였는지도 Turning이 가능하다.

#### 5. 결론

인터넷 및 분산 컴퓨팅 환경의 혁신으로 나온 XML과 웹 서비스가 활성화되기 위해서 해결해야 할 주요 난제로 지목되고 있는 보안 부문을 해결하기 위해 국제 표준화 기구가 XML 관련 표준화 작업을 활발하게 해왔으며 이를 기반으로 산업계에서 웹 서비스 보안을 위한 표준화 작업에 적극적으로 참여하고 있다. XML/SOAP에 대한 보안인 XML-Aware Networking은 앞으로도 중요 이슈가 될 것이며 이를 가능하게 하기 위해 Filtering 및 핸들링에 대한 많은 연구가 있을 것으로 예상된다. 그렇기 때문에 이에 관한 연구도 지속적으로 하여 실제 구현을 위한 연구도 해 나갈 것이다.

#### 참 고 문 헌

- [1]The World Wide Web Consortium(W3C)'s XML web page, <http://www.w3.org/XML/>
- [2]Rick McGuir., XML Acceleration Appliances. Emerging Internet Technologies IBM Software Group. November 5, 2003
- [3]DataPower web page, <http://www.datapower.com>
- [4]최성운, 진화하는 J2EE 플랫폼, 명지대학교 컴퓨터공학과 교수
- [5]IBM Redbooks, IBM WebSphere Application Server V5.1 System Management and Configuration WebSphere Handbook Series
- [6]The Java XML Downloads web page, <http://java.sun.com/xml/download.html>
- [7]The Java Web Services Tutorial web page, <http://java.sun.com/webservices/docs/ea2/tutorial/doc/JAXP-Intro4.html>