XML 기반 이더넷 프로토콜 분석

이재종*, 황재민*, 정인환*
*한성대학교 컴퓨터공학과
e-mail : {ejejo, jeminad, ihjung}@hansung.ac.kr

Ethernet Protocol Analyzing based on XML

Jae-Jong Lee*, Jae-Min Hwang*, In-hwan Jung*
*Dept of Computer Science, Hansung University

요 약

이더넷 프로토콜을 분석하는 것은 다양한 프로토콜 사양을 정확하게 해석하고 프로토콜의 확장성이 고려되어야 한다. 본 논문에서는 XML의 DOM 기술을 이용하여 이더넷 프로토콜을 분석할 수 있는 파서를 제안한다. 제안된 파서는 이더넷을 이용하는 각종 프로토콜 사양을 쉽게 기술할 수 있는 틀을 제공하고, 새로운 프로토콜이 추가되면 XML 언어의 확장성을 이용하여 유지 및 보수에 드는 비용을 최소화 할 수 있는 장점이 있다. 본 논문에서는 XML을 이용한 이더넷 파서를 PDA상에서 무선 LAN 패킷을 분석할 수 있도록 Win CE 기반 개발에 사용되는 EVC++ 언어로 구현되었다.

1. 서론

이더넷 프로토콜 분석이란 이더넷(Ethernet)의 특성인 동보 기능(broadcasting)을 이용하여 LAN에 흘러 다니는 모든 패킷들을 실시간으로 수집하여 OSI 7 Layer별로 분석하는 것을 의미한다. 프로토콜 분석을 이용한 응용이 일반 PC에서만 사용이 가능한 형태로 개발되어 있는 실정이다. 하지만 PDA 상에서 무선 LAN을 대상으로한 응용은 찾아보기 힘들다. 더블어 새로운 프로토콜에 대해서 확장성을 고려하기 어렵고 또한 유지 및 보수에 드는 비용을 고려하여야 한다.

XML(eXtensible Markup Language)[1]은 인터넷 상에서 문서나 데이터를 교환 또는 배포할 때 표준이 될 수 있는 마크업 언어(markup language)이고, 데이터를 위한 표준 포멧이라고 정의하고 있다[2]. 기존의 HTML과는 달리 태그를 새롭게 만들 수 있기 때문에 웹페이지 뿐만 아니라 그 외의 데이터도 확실히 표현할 수 있다는 점이 기존의 HTML의 한계를 뛰어 넘는다. XML을 범용성을 지닌 데이터 표현 언어라고 할 수 있는 또 다른 이유는 XML이 점점 확산되고 있다는 점을 들 수 있다.

본 논문에서는 XML 기술을 이용하여 무선 LAN 상에서 PDA를 이용하여 프로토콜을 분석할 수 있는 프로토콜 분석 기법을 제안한다. 또한 프로토콜 분석에서 사용할 프로토콜 정의를 XML 포맷으로 설계한다. 이는 새로운 프로토콜이 추가될 경우, XML 언어를 이용하여 새로운 프로토콜에 대해서 추가, 삭제 및 수정을 보다 용이하게 하기 위함이다.

제안한 프로토콜 분석 기법을 구현하기 위해서는 XML 문서를 그대로 취급하는 것보다는 프로그램적으로 처리할 수 있는 방법이 필요하다. 즉, 프로그램과 연결해서 XML 문서를 처리할 수 있는 API(Application Programming Interface)와 파서 기능이 필요한데, 이 방법으로는 크게 DOM(Document

Object Model)[3]과 SAX(Simple API for XML) 방식이 있다. DOM 방식은 XML 문서의 전체적인 표상을 트리 구조 형태로 메모리에 상주시켜서 처리하는 방식을 채택하는 반면, SAX는 문서 전체적인 표상을 메모리에 기억시키지 않고, 필요한 사건(event)에 적합한 처리를 지정함으로써, 메모리 부담을 줄이면서도 필요한 부분에 대해서 처리할 수 있는 장점을 갖고 있다. 파서 기능과 프로그램 API들은 프로그램과의 연결을 목적으로 하는 것이니만큼, 우선적으로 프로그램 언어의 선택에 따른 적합한 파서와 API를 선택해야한다. 현재 웹 브라우저와 바로 연동되면서 결과를 볼 수 있는 장점 때문에 Microsoft XML parser(MSXML)[4]을 이용하였다. 하지만 WinCE 3.0를 사용하는 PDA는 MSXML의 SAX 방식은 지원하지 않는다[5]. 따라서 DOM 방식을 이용하여 구현한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 제안된 프로토콜 분석 기법의 설계 및 구현 방법에 대해 기술한다. 마지막으로 결론 및 향후 연구 과제를 3장에서 언급하겠다.

2. 설계 및 구현

제안한 프로토콜 분석 기법을 XML 기반으로 설계하였다. 프로토콜 정의 문서는 본 논문에서 정의한 문법으로 만들어진 것으로 프로토콜 분석 과정에서 사용될 잘 정의된(well-formed) XML 문서이고, 프로토콜 분석 기법은 XML DOM API를 이용하여 각 프로토콜 계층 구조를 탐색하면서 해당하는 헤더를 분석하는 것이다.

2.1 프로토콜 정의 문서

XML에서 정보를 표현하는 방식은 크게 태그의 형태로 표시되는 element와 이 태그의 부속 정보를 표현하는 속성(attribute)으로 이루어진다. 따라서 본 논문에서는 프로토콜에서 필요한 정보를 직접 다음과 같이 모델링 하였다.

각 프로토콜을 element로 표시하고, 세부 정보를 서브 element로 정의하였다. element의 attribute는 프로그램에서 사용되어질 이름(name)과 사용자가 인식할 수 있는 프로토콜 이름(sourcename)을 가지고, 서브 element는 헤더 정보(fields)와 상위 프로토콜 정보(nextprotocol)를 가진다. 또한 서브 element에서 사용할 태그를 [그림 1]과 같이 제안하였다.

fixed	바이트 단위를 가지는 헤더 정보
masked	비트 단위를 가지는 헤더 정보
switch	헤더 정보를 이용한 조건 판단문
variable	헤더 정보를 제외한 데이터 정보

[그림 1] 중요 태그 설명

[그림 1]과 같은 태그를 사용하여 본 논문에서는 프로토콜 분석 기법에서 사용하는 프로토콜 정의 문서를 설계하였다. [그림 2]는 XML 언어와 제안된 태그를 이용하여 생성된 프로토콜 정의 문서의 한 예이다. [그림 2]와 같이 프로토콜 정의 문서에는 startprotocol element와 defaultprotocol element가 존재한다. Ethernet을 기반으로 한 LAN에서는 Ethernet 프로토콜과 IEEE 802.3 프로토콜은 서로 호환성을 가진다. 따라서 startprotocol이라는 하나의 element로 정의하였다. 사용자가 인식할 수 있는 이름(sourcename)은 프로토콜 분석시 동적으로 변경할 수 있도록 하였다. defaultprotocol element는 모든 프로토콜 분석이 끝난 후 또는 상위 계층의 프로토콜이 존재하지 않을 경우에 대비하여 정의하였다. 각 계층의 프로토콜을 분석한 후에 상위 계층의 프로토콜이 존재하는지를 판단한다. 만약 존재하지 않을 경우에는 문제가 발생된다. 해당 계층의 프로토콜 분석 후에 상위 계층의 프로토콜이 존재하지 않으면 나머지를 데이터로 간주해야 하는데 이러한 정보를 프로토콜마다 태그를 추가할 경우에는 불필요한 중복이 발생하기 때문에 defaultprotocol element를 정의하였다. [그림 3]은 IP 프로토콜의 헤더 정보를 상세하게 보여주고 있다.

```
<?xml version="1.0" ?>
- <ProtocolDef name="ProtocolDefinition">
  + <startprotocol name="Ethernet" sourcename="Ethernet or IEEE 802.3">
  + <protocol name="IP" sourcename="IPv4">
  + <protocol name="ARP" sourcename="ARP(Address Resolution Protocol)">
  + <protocol name="ICMP" sourcename="ICMP(Internet Control Message Protocol)">
  + <protocol name="TCP" sourcename="TCP(Transmission Control protocol)">
  + <protocol name="FTP" sourcename="FTP(File Transfer Protocol)">
  + <protocol name="TELNET" sourcename="TELNET(Teletype Network)">
  + <protocol name="SMTP" sourcename="SMTP(Simple Mail Transfer Protocol)">
  + <protocol name="HTTP" sourcename="HTTP(Hyper Text Transfer Protocol)">
  + <protocol name="POP3" sourcename="POP3(Post Office Protocol - Version 3)">
  + <protocol name="UDP" sourcename="UDP(User Datagram protocol)">
  + <protocol name="DNS" sourcename="DNS(Domain Name Server)">
  + <protocol name="NetBios" sourcename="NetBios Name Service">
  + <defaultprotocol name="data" sourcename="Data">
  </ProtocolDef>
```

[그림 2] 프로토콜 정의 내용

```
- <protocol name="IP" sourcename="IPv4">
  - <fields>
    + <masked name="verhlen" sourcename="Version and header Length" type="hex" size="1">
      <fixed name="tos" sourcename="Type of service" type="hex" size="1" />
      <fixed name="tlen" sourcename="Total length" size="2" />
      <fixed name="identification" sourcename="Identifier" size="2" />
    + <masked name="ffo" sourcename="Flags and Fragment offset" type="hex" size="2">
      <fixed name="ttl" sourcename="Time to live" size="1" />
      <fixed name="pt" sourcename="Protocol" size="1" />
      <fixed name="hchecksum" sourcename="Header Checksum" type="hex" size="2" />
      <fixed name="src" sourcename="Source address" group="1" seperate="." size="4" />
      <fixed name="dst" sourcename="Destination address" group="1" seperate="." size="4" />
    </fields>
  + <nextprotocol>
  </protocol>
```

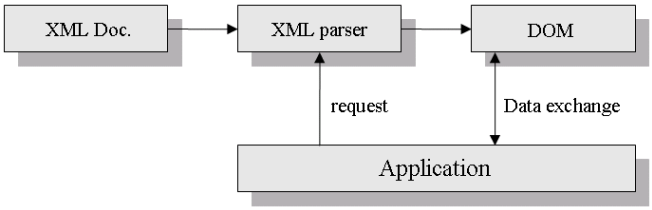
[그림 3] 세부적인 프로토콜 정의 내용

프로토콜을 독립된 태그인 element로 모델링 하거나, 아니면 다른 element의 속성인 attribute로 모델링 하는 것에 따라 프로토콜 정의 문서의 확장성에 많은 영향을 미친다. 뿐만 아니라 프로토콜 분석을 위한 프로그램 설계시 어려움이 있다. 따라서 본 논문에서는 이와 같은 문제들을 고려하여 모델링 하였다.

2.2 프로토콜 분석 기법

DOM은 기본적으로 XML 문서를 구조적으로 표현한 것이다. DOM 프로그래밍[6][7][8][9]의 일반적인 구조는, 먼저 Document 클래스의 인스턴스를 만든 후 XML 문서를 로드하여, 하위 트리 구조를 만든 다음 그 트리 구조를 탐색하거나, 변형을 가하는 것이 DOM 프로그래밍이다. 그리고 XML 파서의 기능은 XML 문서의 내용에 일대일로 대응되는 DOM 노드들을 생성하는 것이다.

본 논문에서는 XML 파서와 DOM 방식을 이용하여 본 논문에서 설계한 프로토콜 정의 문서를 로드하여 DOM 노드를 생성한 후, DOM API를 사용하여 얻어진 프로토콜 정의 정보를 분석할 프로토콜 헤더 정보와 비교 분석하는 프로토콜 분석 기법을 설계 및 구현하였다. [그린 4]는 DOM 프로그래밍의 일반적인 방식을 보여주고 있다.

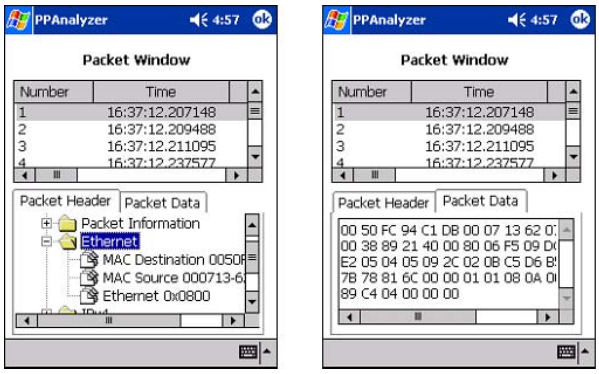


[그림 4] 일반적인 DOM 프로그래밍

프로토콜 헤더 정보를 분석하기 위한 방법으로 [그림 5]와 같은 함수들을 지원한다. InitializeProtocolDefinition 함수는 XML Document 클래스의 인스턴스를 만들어 프로토콜 정의 문서를 읽어 들인다. 읽어 들인 문서를 DOM API를 이용하여 element 이름이 protocol인 노드의 개수를 검색한다. 검색된 protocol 노드의 수만큼 Element 인스턴스를 만들어 위치 정보를 모두 저장한다. 또한 startprotocol 노드와 defaultprotocol 노드의 위치 정보도 저장한다. 이것은 구현시 불필요한 연산을 최소화 시키기 위한 것이다. 그리고 프로토콜 헤더 정보를 나타내는 서브 element의 fields 노드와 프로토콜의 상위 계층 프로토콜이 존재하는지 판단하기 위하여 nextprotocol 노드의 위치를 저장한다. DecodeFields 함수는 노드 타입 정보를 이용하여 분석될 프로토콜 헤더가 fixed 필드인지 아니면 masked 필드인지 또는 switch 필드인지 구분하여 수행한다. 이는 ICMP 프로토콜과 같이 type 필드 값에 따라서 다음에 오는 필드의 크기 및 의미하는 정보가 달라질 수 있기 때문이다. DecodeFields 함수에는 사용되는 함수로는 HeaderElementInitialize와 HeaderElementUpdate가 있다. HeaderElementInitialize 함수는 각 프로토콜당 한번 호출되는데, 현재 프로토콜의 이름과 사용자가 인식할 수 있는 이름 정보를 프로토콜 정의 문서에서 가져와 저장한다. HeaderElementUpdate 함수는 프로토콜 정의 문서의 서브 element의 헤더 정보를 이용하여 분석된 결과를 저장한다. GetNextProtocol 함수는 InitializeProtocolDefinition 함수를 통하여 얻어진 상위 계층의 프로토콜 위치 정보를 이용하여 계속 프로토콜이 분석되어야 하는지 결정한다. 상위 계층의 프로토콜이 존재할 경우 switch 연산을 수행하여 분석되어야 할 프로토콜 위치 정보를 변경한 후 계속 분석을 수행하지만, 없을 경우에는 defaultprotocol 노드로 위치 정보를 변경한 후 현재까지 분석된 프로토콜 헤더 정보를 제외한 나머지를 전부 데이터로 간주하고 저장하게 된다. [그림 6]은 프로토콜 분석 기법을 적용하여 구현된 프로그램의 실행 화면이다.

```
BOOL   InitializeProtocolDefinition(CString xmlFile);
int    DecodeFields(IXMLDOMElement *Parent);
int    GetNextProtocol();
```

[그림 5] 중요 프로토콜 분석 함수



[그림 6] 프로토콜 분석 결과

3. 결론 및 향후 연구

본 논문에서는 XML의 DOM 기술을 이용한 이더넷 프로토콜 분석 기법을 설계하고 무선 LAN 환경의 PDA에서 이 분석 기법을 기반으로 한 이더넷 프로토콜 분석기를 실행 가능하도록 구현하였다. 이렇게 구현된 이더넷 프로토콜 분석기는 테스트를 통해 프로토콜 정의 문서에 기술되어 있는 구조대로 수집된 프로토콜 헤더를 정확하게 분석함을 확인하였다. 또한 분석된 패킷의 프로토콜 정보가 구조화된 데이터 정보를 이용하기에 적합한 형태로 설계된 XML 언어로 변환되어졌다. 그러므로 이렇게 XML 언어로 변환된 정보는 OSI 7 Layer에 정의된 계층 구조의 형태를 보이며 이더넷 패킷의 정보를 편리하게 활용할 수 있게 되었다. 또한 기존의 프로토콜 정의 문서에 포함되어 있지 않은 새로운 프로토콜에 대해서도 분석 프로그램의 수정 없이 프로토콜 정의 문서에 미리 정의된 문법에 따라 프로토콜을 추가함으로서 프로토콜 분석의 확장성 면에서 많은 장점이 있음을 확인하였다. 뿐만 아니라 이러한 XML로 만들어진 프로토콜 분석 기법은 프로그램에 종속되어 있지 않기 때문에 새롭게 구현되어질 다른 프로그램에서도 기존의 프로토콜 분석 모듈의 사용을 통해 기술의 축적과 프로그램을 유지 및 보수에 드는 비용을 최소화 할 수 있게 되었다.

본 논문에 이은 향후 연구로는 분석할 수 있는 이더넷 프로토콜의 종류를 추가하고, 기존의 일반 PC보다 성능이 현저하게 낮은 PDA 상에서 XML을 이용한 이더넷 프로토콜 분석 기법의 성능을 평가하는 것이다.

참고문헌

[1] “Extensible Markup Language (XML) 1.0”, World Wide Web Consortium Recommendation. (http://www.w3.org/TR/REC-xml)

[2] XML White Paper (http://msdn.microsoft.com/xml/articles/xmlwhite.asp)

[3] W3 Consortium, "Document Object Model(DOM)" (http://www.w3.org/DOM/)

[4] “msxml파서”, MSDN Online: XML 개발자 센터 (http://www.microsoft.com/korea/msdn/xml/default.asp)

[5] Windows CE Feature by Feature (http://msdn.microsoft.com/embedded/prodinfo/compare/wincefea/default.aspx)

[6] 이경하, 이강찬, 이규철, “XML프로그래밍”, 정보과학회 제18권 4호 pp4-12.

[7] 김창수, 정희경, “XML 응용 개발환경”, 정보과학회지 제19권 제1호, 2001. 1

[8] 이강찬, 손홍, 박기식, “XML 표준화 동향”, 정보과학회지, 제19권 제1호

[9] 선승상, 박상윤, 엄영익, “XML 문서의 객체지향적 관리를 위한 XML DOM 소프트웨어의 설계 및 구현”, 정보처리학회 춘계학술대회, Vol 07, No 1, 2000. 4