정보학연구 제9권 제3호 2006년 9월(pp.59-66)
The Journal of Information Technology, Vol.9, No.3

분산환경에서의 디렉토리 서비스 프로토콜에 관한 연구

이지영*

목 차

Ⅰ. 서론

컴퓨터 기술과 정보통신 기술이 발달함에 따라 통신망을 이용한 다양한 정보 기술들이 사용되고 있다. 그러나 통신망에 연결된 자원이 늘어남에 따라 이 정보들을 효율적으로 관리하기 위한 시스템이 필요하게 되었다. 그러므로 정보통신에 필요한 정보를 데이터베이스화 하여 이를 효율적으로 관리하고 또한 사용자가 편리하게 사용할 수 있는 기능을 제공하기 위하여 디렉토리 서비스 시스템이 개발되었다. [1]

공개키 암호방식을 이 시스템에 적용할 때 공개키 암호방식은 키 알고리즘이 공개되어 공개키 디렉토리 게시판 등의 메커니즘이 불안하여 위조 및 변조의 문제가 발생하였다.

본 논문에서는 위조 및 변조가 되지 않았음을 보장하기 위하여 통신망에 적합한 디렉토리 접근 메카니즘을 연구하여 분산 환경에서의 디렉토리 서비스 프로토콜을 연구하였다.

그 결과 LDAP(Lightweight Directory Access Protocol)에서 질의관련 연산들을 통합하여 search 연산에 포함한 결과 search 연산의 기능을 확장하였다. 또한 특정 상황에 대해 순차적으로 반복되는 기존의 DAP 연산들을 대치하는 새로운 연산을 개발하여 LDAP 클라이언트가 고려해야 할 사항들은 간소화 하였다. 즉, 순차 반복 연산의 대체 연산을 개발하였다.

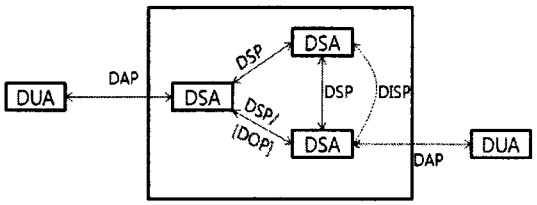
Ⅱ. 본론

디렉토리는 ISO에서 제정한 OSI 모델 중 응

* 세명대학교 컴퓨터학부 교수

용계층(Application layer)에 속하는 프로토콜로서 다른 계층서비스들이 필요한 정보를 디렉토리에 물어 볼 수도 있고, 사용자가 디렉토리에 직접 물어보는 것도 가능하도록 설계된 프로토콜이다[2,3]

디렉토리는 분산된 DSA(Directory Service Agent)들로 구성되고 디렉토리 액세스 프로토콜(Directory Access Protocol : DAP)을 이용하여 디렉토리 사용자에게 서비스를 제공한다. 사용자는 직접 디렉토리에게 서비스를 요구하지 않고 디렉토리 사용자 에이전트(Directory User Agent)를 통해 디렉토리에 액세스할 수 있다.[4]



〈그림 1〉 디렉토리 구성

디렉토리에 사용자의 요구를 전달하고, 응답을 받는 응용 프로세스는 디렉토리와 사용자와의 인터페이스 역할을 수행한다. 이 응용프로세스가 디렉토리 사용자 에이전트(Directory User Agent : DUA)이다. 디렉토리 내의 정보들은 각각의 시스템에 분산되어 관리 및 운영되므로 사용자는 분산된 전체 디렉토리를 하나의 지역 디렉토리로 간주하여 사용할 수 있다. 분산된 디렉토리 시스템에서 사용자의 요구를 디렉토리 프로토콜에 따라 수행하는 응용 프로세스들이 디렉토리 시스템 에이전트(Directory System Agent : DSA)이다.[5]

DSA는 사용자가 지시한 명령이나 요구들에 대해서 디렉토리를 통해서 그것을 수행하고 그것의 결과를 사용자에게 되돌리는 역할을 하며 디렉토리에는 여러개의 DSA가 존재할 수 있다. DUA는 DSA에 잇는 접근점을 통해서 디렉토리에 접근하게 되는데 하나의 DSA는 하나 이상의 접근점을 가지고 있다.

디렉토리 엑세스 프로토콜(Directory Access Protocol : DAP)은 사용자가 디렉토리에 서비스를 요구하는 DUA와 DSA간의 프로토콜이다. 디렉토리 시스템 프로토콜(Directory System Protocol : DSP)은 사용자가 요구한 서비스를 수행하기 위한 두 DSA 간의 프로토콜이다. 디렉토리 정보 새도우 프로토콜(Directory Infomation Shadow Protocal : DISP)은 한 DSA가 보유하고 있는 정보를 다른 DSA에게 복사하여 사용할 경우 복사를 해준 DSA가 특정된 상황이 발생할 때마다 복사를 해주어야 하는데 이와 같이 복사된 정보를 두 DSA간에 주고받기 위해 사용되는 프로토콜이다. 디렉토리 동작 바인딩 관리 프로토콜(Directory Operational Binding Management Protocol : DOP)은 두 DSA간에 관리적인 동작 관계가 설정되어 있는 경우 이들 DSA간의 통신을 위해서 사용되는 프로토콜이다. 그림1은 디렉토리 구성이다.

Ⅲ. 분산 환경에서의 디렉토리 접근 프로토콜

디렉토리 서비스를 위한 프로토콜은 OSI 7계층 통신 모델에서 운용된다. 응용계층에서는 서비스를 제공하고 6계층은 통신상에서 연동될 수 있도록 ASN.1 코드로 바꾸어 주고, 5계층은 세션(session)을 설정한다. 하지만 일반사용자들에게 자신의 시스템에 DUA를 구현하여 디렉토리에 접근하는 것이 불편하므로 보다 사용하기 편리하고 구현하기도 용이한 개선된 LDAP(Light-

weight Directory Access Protocol)을 제안한다.

기존의 LDAP은 X.500 시리즈에서 제안한 DAP을 개선한 것이었다. 대부분의 인터넷 사용자들은 기본적으로 TCP/IP를 기반으로 하고 있다. 이에 LDAP은 기존의 DAP과 동일한 기능을 제공하면서도 TCP/IP상에서 동작할 수 있도록 구현 되었으며 사용자들이 고려해야할 연산자들의 수를 줄여 서비스의 질을 개선하였다. 그 결과 LDAP은 기존의 DAP라는 다른 형태를 가져 기존의 X.500 디렉토리 서버와 호환성을 가질 수 없게 되었으나 LDAP에 LDAP서버를 두어 이를 해결하였다.

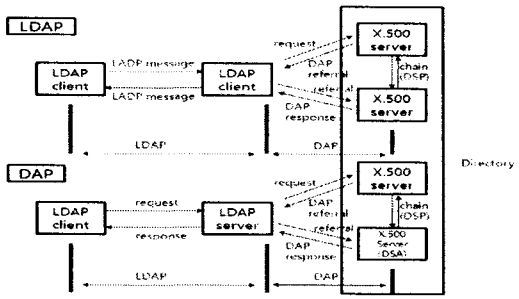
LDAP서버의 역할은 다음과 같다.

첫째, 사용자의 디렉토리 접근 제공

둘째, LDAP과 DAO의 게이트웨이

셋째, referral 처리이다.

그림 2는 LDAP과 DAP이 X.500 디렉토리와 동작하는 법을 나타내고 있다.



〈그림 2〉 LDAP과 DAP의 동작비교

그림 2에서 보듯이 DAP과 LDAP의 구조를 보면 3부분으로 구성되어 있다.

첫째, 사용자 부분은 일반사용자들 또는 사용자가 시작하는 응용프로그램이 위치하는 부분으로 사용자들은 디렉토리 접근 부분을 통하여 디렉토리 서비스를 받는다.

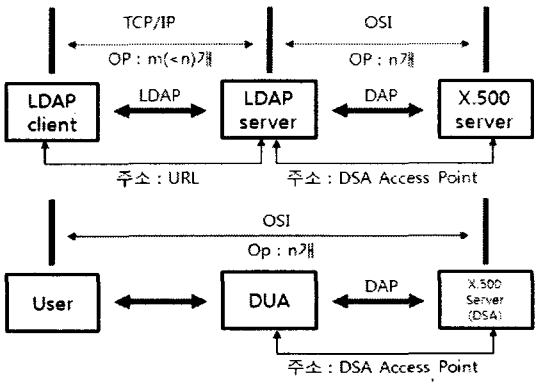
둘째, 디렉토리 접근부분은 사용자 부분으로부터 들어온 디렉토리 서비스 요구를 받아 이를 디렉토리에 전달한 후 다시 디렉토리로부터의 결과를 사용자에게 되돌리는 역할을 수행한다. DAP의 경우에는 DUA가 LDAP의 경우에는 LDAP서버가 이 기능을 수행한다. 이때, LDAP서버는 LDAP과 DAP간에 일치하지 않는 부분에 대해서 상호 변환을 해주어야하는 게이트웨이 기능을 추가로 포함한다.

셋째, 디렉토리 부분은 일반적으로 X.500디렉토리서버 부분이며 DSA를 의미하고 이들간에는 DSP를 통해서 통신을 하게 된다. 디렉토리 부분이 LDAP서버에 포함되는 경우에는 사용자가 서비스를 받을 수 있는 영역은 지역 디렉토리로 한정되며 다른 지역 디렉토리로부터 서비스를 받고자 하는 경우에는 사용자가 직접 해당영역에 접근해야 한다.

3.1. DAP과 LDAP의 비교

DAP은 X.500디렉토리 시스템 상에서 사용자에게 접근을 제공하는데 사용되나 OSI 7계층을 기반으로 동작하므로 각 계층 고유의 헤더로 캡슐화 되므로 단순한 데이터에도 그 크기가 커진다. 그러므로 메시지 전송시간이 길어지며 수행속도가 저하되고 일반성이 떨어진다. 또한 데이터 표현에 있어서 복잡한 구조를 가지고 코딩을 해야 하고 비슷한 연산자들이 중복되어있다. LDAP은 DAP의 단점들을 해결하도록 개발되었으며 DAP과 LDAP의 차이점은 다음과 같다.

위에서 본 바와 같이 LDAP 기반의 TCP/IP는 4계층으로 되어 있어 데이터의 길이가 OSI의 경우보다 짧고 전송률도 개선된다. 그리고 URL을 LDAP서버의 주소로 사용하여 사용자들이 보다 수월하게 이용할 수 있으며 DN에 대하여 스트링 코딩을 할 수 있도록 하여 코딩의 길이를 줄이고



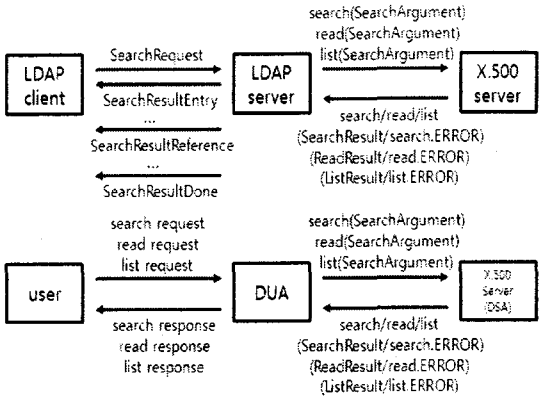
〈그림 3〉 DAP과 LDAP의 차이점

단순화 하였다. 또한 비슷한 기능을 하는 연산자들을 하나의 연산자로 표현하여 연산자 수를 줄여 사용자들이 서비스 받기가 더 용이해졌다.

<표 1>은 DAP과 LDAP의 검색결과 전달방식의 비교를 나타낸 것이고 그림 4는 LDAP과 DAP의 search 연산을 비교한 것이다.

〈표 1〉 DAP과 LDAP의 검색 결과 전달방식의 비교

	DAP	LDAP
장점	-검색된 결과를 한 번에 전송하므로 전송률이 좋다.	-전송 되는 데이터의 각각 내용을 확인할 수 있다. -검색된 결과를 한 번에 전송하므로 전송률이 좋다.
단점	-수신측에서는 전송되는 데이터를 모두 받기 전에는 내용을 볼 수 없다. -수신측에서는 전송되는 데이터를 저장할 메모리나 버퍼 또는 별도의 저장소가 필요하다.	-각각의 메시지에 헤더가 별도로 붙게 되므로 전송량이 증가하여 전송률이 낮다.



〈그림 4〉 LDAP과 DAP의 search 연산 비교

Ⅳ. 분산 환경에서의 개선된 디렉토리 접근 프로토콜

기존의 DAP이 많은 문제점을 가지고 있어 이를 개선한 LDAP이 개발되었으나 연산자의 기능 확장과 인증서비스에 유용한 연산자의 부족 및 순차적으로 반복되는 연산의 대체 연산 등이 필요한 과제로 남아있다.

그러므로 본 논문에서는 기존의 LDAP의 연산들을 더욱 개선하고 인증서비스를 디렉토리상에서 구현하는데 더욱 유용한 연산을 제안한다.

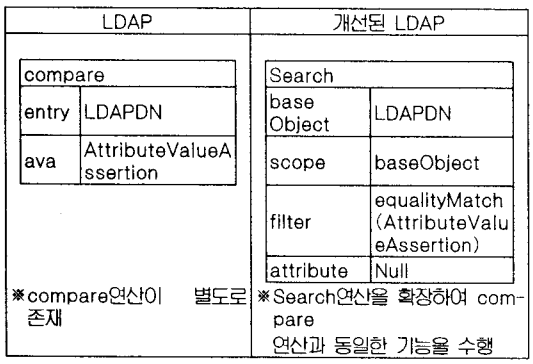
4.1. search연산의 기능확장

search 연산의 기능확장에는 compare 연산기능을 추가한다. compare 연산은 object에서 지정된 엔트리에 대해서 selection에서 선택한 속성값들과 비교하여 이것이 동일하다면, True를 전달하고 그렇지 않으면 False를 전달한다.

이것을 search 연산으로 표현할 수 있다. 전달되는 인수들을 지정한 후 되돌아오는 연산의 결

과가 null이 아닌 경우는 비교하고자 하는 엔트리가 디렉토리에 존재하고 그것의 지정한 속성값과 동일하다는 것을 의미한다. 반대로 null인 경우에는 비교엔트리가 디렉토리에 존재하지 않거나 해당 속성값이 동일하지 않다는 것을 나타낸다.

그림 5는 DAP에서의 compare 연산을 search 연산으로 나타내었을 경우, search연산의 인수들이 가져야 할 값을 나타내고 있다.



〈그림 5〉 LDAP과 개선된 LDAP의 compare 기능 비교

여기서 baseObject는 object에서 지정하는 엔트리를 지정하며 filter는 현재 디렉토리에 존재하는 엔트리와 지정된 값과 동일한 경우로 제한한다.

4.2. tree 연산기능 추가

상대방에게 인증서를 전송하는 경우 상대방까지의 인증경로, 그리고 자신에게 돌아올 수 있는 반대 인증경로를 지정하여 전송해 주어야 하는데 이를 위해서 사용자는 계층적으로 구성되어 있는 인증기관의 구조를 알아야 한다. 이에 검색 연산의 인자를 조절하면 인증 서비스에서 지정된 위치를 루트로 하는 서브 트리에서 인증 기관의 계층적 구조 정보를 사용자가 알 수 있다. 각각의 사용자 및 인증기관은 자신의 인증서를 엔트리에 보관한다. tree연산을 search 연산으로 나타내었을 경우, search 연산의 인수들이 가져야 할 값은 다음과 같다.

LDAP에서는 tree기능을 하는 연산이 없다. 개선된 LDAP에서는 search에서 base object는 LDAPDN이고 scope는 whole subtree 이며 filter는 present.type={user Certificate|cA Certificate} 이며 attributes는 Null이다. 이때 baseobject는 서브 트리의 루트가 되는 엔트리를 지정한다. scope는 검색하고자 하는 범위를 baseObject를 루트로 갖는 서브트리 전부로 한다.

filter는 특정 엔트리 이하에 존재하는 모든 인증 서비스 사용자들을 검색하는 경우, user Certificate를 갖는 엔트리만으로 제한하고 특정엔트리 이하의 인증기관의 구조를 검색하는 경우, cA certificate 속성을 가지는 엔트리만으로 제한한다. attributes는 엔트리의 정보 중 결과로 되돌려 받고자 속성 및 속성값을 지정한다. 표2에서 제안하는 search 연산은 DAP의 compare 연산과 인증 서비스에서 유용하게 사용될 수 있는 tree 연산 기능을 포함한다.

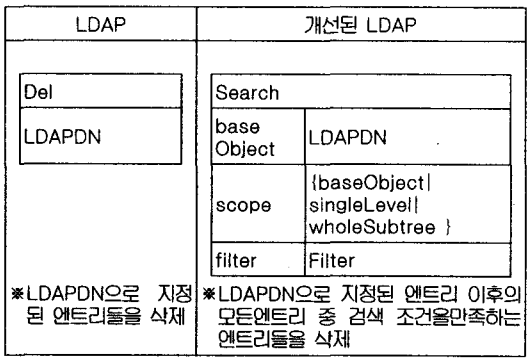
〈표 2〉 확장된 search 연산

연산 / 인수	개선된 LDAP			
	기존의 Search 연산		확장된 Search 연산	
	read	list	compare	tree
base Object	base Object	base Object	base Object	base Object
scope	base Object	single-Level	base Object	whole Subtree
filter	present	present	equality	present =cACertificate/userCertificate
attri-bute	Attrivute Descrip-tion List	RDN	Attribute Description List	Attribute Description List
비고			if SearchResult= null then Result←T else Result←F	

4.3. 순차반복 연산의 대체 연산 개발

LDAP에서 지원하고 있는 Del, Modify 연산들은 하나의 엔트리뿐만 아니라 여러 개의 엔트리에 대해서 명시적으로 지정하면 동시수행이 가능하다. 엔트리수가 많아지면 문제가 될 수도 있다. 그러므로 검색기능을 추가할 경우 필터에서 주어진 조건을 만족하는 엔트리들에 대해 연산을 적용할 수 있으므로 매우 유용하나 제안하는 Ext-Del 연산은 기존의 Del 연산에 조건검색 기능을 추가하였다. ExtDel 연산은 baseObject에서 지정한 엔트리 이하의 서브트리에 대해서 검색을 하는데 검색조건을 만족하는 엔트리가 존재하면 해당 엔트리를 삭제한다. 그러므로 검색 후 다시 삭제하는 불편을 개선하였다.

그림 6은 LDAP의 Del 연산과 개선된 LDAP의 ExtDel의 연산 비교이다.



〈그림 6〉 LDAP과 개선된 LDAP의 연산비교

아래는 개선된 LDAP의 ExtModify 연산을 ANS.1 코드로 나타낸 것이다

```
ExtDelRequest ::= SEQUENCE {
baseObject LDAPDN,
scape ENUMERATED {
  baseObject (0),
  singleLevel (1),
  wholeSubtree (2) },
derefAliases ENUMERATED {
  neverDerefAliases (0),200
  derefInSearching (1),
  derefFindingBaseObj (2),
  derefAlways (3) },
sizeLimit INTEGER (0 .. maxInt),
timeLimit INTEGER (0 .. maxInt),
filter Filter }

ExtDelResultReference ::
   = SEQUENCE OF LDAPURL
ExtDelResultDone ::= LDAPResult
```

V. 결론

기존의 연산은 해당 엔트리를 검색한 후 Modify 연산을 적용해야 하지만 본 논문에서는 기존의 Modify 연산에 조건 검색기능을 추가하였다. ExtDel 연산은 먼저 baseObject에서 지정한 엔트리 이하의 서브 트리에 대해 검색을 하는데, 검색 조건을 만족하는 엔트리가 존재하면 해당 엔트리를 수정한다. 그러므로 사용자는 search 연산을 전달한 후 다시 Modify 연산을 전달하는 불편을 개선하였다.

참고문헌

1) 우종식, 안순신, "디렉토리 서비스", 「정보과학회지」, 제8권 제6호, 1992.

2) CCITT X.500, *The directory: Overview of*

concepts, models and services, CCITT, 1992.

3) CCITT X.501, *The directory: The models,* CCITT, 1992.

4) 이재광, 이용준, "X.500 디렉토리 정보보호", 「통신 정보 보호 학회지」, 제4권 제31호, 1994. 9, pp.22-23.

5) 정권성, 김영수, 이형규, 조동욱, 원동호, "디렉토리 서비스에 관한 고찰", 「통신정보보호 학회지」, 제8권 제1호, 1998. 3, pp.31-57.

A Study in the Directory Service Protocol in Distributed Environment

Jie-Young Lee*

Abstract

This paper proposes directory service protocol in distributed environment. By adding an condition retrieval function to Modify operation, the ExtModify operation improves the inconvenience to transfer a search operation and resume a Modify operation.

The method suggests how to correct all the entries which require retrieval condition out of other entries designated by LDAPND

Key Words : Directory Service Protocol, LDAPND

* Professor, Dept. of Computer, Semyung University.