

다국어 도메인을 위한 DNS 프로토콜: mIDNS

신혜원 이승익 이동만
한국정보통신대학원대학교 공학부
{hwhshin, silee, dlee}@icu.ac.kr

A New DNS Protocol for Multilingual Domain Names

Hyewon Shin Seungik Lee Dongman Lee
School of Engineering, Information and Communications University (ICU)

요 약

DNS(Domain Name Service)는 인터넷 호스트의 주소를 제공하는 서비스로서 기존의 시스템은 영어 알파벳 이외의 다른 언어들로 구성된 도메인 네임(Domain Name)은 처리하지 못한다. 최근, 인터넷의 국제화에 따른 다국어 도메인의 필요성이 점차 증대됨에 따라 이러한 문제점을 해결하고 다국어 도메인 이름을 처리할 수 있는 새로운 DNS 프로토콜인 mIDNS (Multilingual Domain Name Service)를 제안한다. 기존의 DNS 와의 호환성 및 상호 운용성을 보장하고 특정 언어에 종속되지 않는 시스템을 디자인하기 위해 mIDNS에서는 Unicode 문자 집합을 기반으로 모든 DNS 질의를 UTF-8 인코딩 방식으로 처리하고 이러한 새로운 mIDNS 질의와 기존의 DNS 질의를 구분하기 위해 DNS 질의 헤더에 'IN'이라는 새로운 비트 영역을 지정하여 사용한다.

1. 서론

인터넷이 대중화되고 국제화되면서 영어 문화권 중심적이던 인터넷 정책 및 기술 또한 유럽을 비롯한 아시아 국가들로 확대 적용되고 있다. 이러한 상황에서 인터넷 기술의 근간이 되는 도메인 이름 서비스, 즉 DNS (Domain Name Service)에도 국제화 및 대중화를 위한 새로운 형태의 서비스가 요구되기 시작하였다. 그러나 현재의 DNS는 도메인 이름을 영문자(US-ASCII)만 사용하도록 되어 있어 [4,5] 기타 문화권 내의 인터넷 사용자들은 고유의 언어 문자로 이루어진 도메인 이름을 사용하지 못함에 따라 국제화 및 대중화의 흐름에 대한 역할을 초래하게 되었다. 이에 본 연구에서는 새로운 DNS를 설계하여 영문자 뿐만 아니라 각종 국제 언어 문자로 이루어진 도메인 이름을 지원하기 위한 확장된 DNS 프로토콜, 이른바 mIDNS(Multilingual Domain Name Service)를 제안하고자 한다.

2. mIDNS 프로토콜 고려 사항

2.1 프로토콜 확장

본 연구에서 제안하는 mIDNS 프로토콜은 기존의 DNS 프로토콜 [4,5]을 확장하여 ASCII 코드 기반의 질의 뿐만 아니라 유니코드(Unicode) [10] 기반의 다국어 도메인 이름에 대한 질의도 처리할 수 있도록 한다. 이러한 확장된 프로토콜 질의와 기존의 질의를 구별하기 위해 mIDNS에서 사용하는 도메인 이름 질의의 헤더 포맷에 'IN' 비트 영역 [3]을 설정한다. 기존의 질의에는 이 비트 영역이 0으로 지정되며 [5] mIDNS 프로토콜의 질의에는 이 비트 영역이 1로 지정됨으로써 이를 mIDNS의 질의로 인식할 수 있다. 따라서 'IN' 비트는 mIDNS와 그 외의 DNS를 구별하는 데에 이용한다. 이는 또한 mIDNS의 호환성과 규범화를 위한 case folding [12]에 따른 서버측의 overhead를 줄일 수 있다.

2.1 기존 DNS와의 호환성과 상호 운용성 (Compatibility and interoperability)

본 연구에서 제안하는 mIDNS 프로토콜은 기존의 DNS

프로토콜 [4,5] 을 확장하여 ASCII 코드 기반의 질의 뿐만 아니라 유니코드(Unicode) [10] 기반의 다국어 도메인 이름에 대한 질의도 처리할 수 있도록 한다. 이러한 시스템은 기존의 DNS 프로토콜을 준수하는 다른 시스템과의 호환성(Compatibility) 및 상호 운용성(Interoperability)을 고려하여 기존 DNS 프로토콜을 그대로 유지할 수 있는 환경을 제공해야만 한다.

이를 위해 본 연구에서는 기존의 DNS 프로토콜을 서버와 리졸버 부분 모두 확장해야 한다. 그러나 기존의 DNS는 이러한 확장하에서도 그 기능을 유지해야 한다. 이러한 환경을 제공하기 위해서는 현재 DNS가 사용하고 있는 질의 기준-7 비트 아스키 코드- 과 양립될 수 있어야 한다. 이러한 필요조건을 만족시키기 위하여, wire format으로 UTF-8을 이용한다. UTF-8의 사용환경에서는 ASCII 코드가 변함없이 이용될 수 있기 때문이다.

2.2 국제성(Internationalization)

기존의 DNS는 알파벳으로 이루어진 도메인 네임(Domain name)만 처리할 수 있었다. 그러나 여기에서 제안하는 새로운 mDNS는 도메인 네임이 알파벳과 알파벳을 사용하지 않는 문자를 이용하는 다국어 모두를 처리할 수 있어야 한다. 이러한 여러 국제 문자의 사용 시 mDNS 프로토콜에서 사용하는 문자 데이터(character data)는 오직 하나의 표준으로 유지되어야 한다.

따라서 그 표준으로서 코드 문자 집합(Coded Character Set)으로 Unicode를 이용하고, 코드 인코딩 방식(Code Encoding Set)으로는 UTF-8 [8]을 이용하였다.

코드 문자 집합으로 Unicode를 쓰는 이유는 기존의 ASCII 코드가 알파벳과 아라비아 숫자 그리고 몇 가지의 특수 문자를 사용하기 위하여 이용한 비트 수는 7 비트 이었기 때문에 현재의 다양한 문자와 숫자, 특수 문자들을 표현하는 것을 역부족이다. 때문에 이러한 다양한 문자들을 모두 표현하고 다룰 수 있는 16 비트 혹은 32 비트를 사용하는 Unicode를 제안한다. Unicode는 그 비트수로 인하여 적절한 인코딩 방식이 필요하다. 따라서 기존의 ASCII 와 호환성을 유지하기 위한 UTF-8을 코드 인코딩 방식으로 이용한다.

클라이언트측의 인터페이스에서는 도메인 네임이 지역 문자로 표현되는 것을 허용하나, 이러한 지역 문자는 mDNS의 리졸버를 이용하기 위해서는 모두 Unicode로 변환되어야 하며, 리졸버에서 네임 서버측으로 질의를 전달할 경우에는 모두 UTF-8[8,10]으로 인코딩되어야 한다.

2.3 사용 문자의 규범화(Canonicalization)

기존의 DNS 프로토콜의 기본적인 작동원리는 네임서버(Name server)의 존파일(zone file)에서 찾고자 하는 호스트의 주소에 대한 대답을 네임서버 단계별로 참조하는 것이다. 기존의 DNS에서는 사용 문자의 규범화(Canonicalization)가 그리 큰 문제가 되지 않았다. 왜냐하면 기존의 DNS 프로토콜에서는 문자를 US-ASCII 만 사용하였기 때문에, 다른 문자의 사용을 전혀 고려할 필요성이 없었다. 그러나 다국어 도메인을 지원하기 위해서는 이러한 다국어 문자들의 일관적인 처리를 위하여 사용문자의 인코딩 방식에 대한 일정한 규범이 필요하다. Casefolding 문제 역시 여기에 속한다. mDNS에서는

Casefolding에 대한 서버측의 부담을 덜어주기 위하여, 리졸버측에서 미리 casefolding을 하고 그 이외에는 서버측에서 이 기능을 담당한다. 리졸버측에서의 casefolding 이후 UTF-8로 (인코딩 방식으로 UTF-8을 제안한다. UTF-8을 쓰는 이유는 기존의 DNS(8 비트 clean code 받아들이)가 8-비트를 지원하기 때문에 기존의 DNS 와의 효과적인 양립성을 위하여 이용된다.) 변환하고 'IN' 비트를 1로 지정한다. 이렇게 되면 casefolding에 대한 서버측의 부담을 낮출 수 있다. 이러한 방식에서 사용되는 문자는 그 문자의 종류에 상관없이 모두 Unicode[10]로 변환되어야 하며 그 또한 리졸버에서 네임서버로의 전달 전에 기존의 DNS 질의와 호환성을 위하여 UTF-8로의 변환이 있어야 한다. 이러한 mDNS의 문자 표준의 성립은 지역 문자 코드에 상관없이 하나의 표준으로 mDNS를 사용할 수 있는 장점을 제공한다.

2.4 mDNS의 운영원리(Operational issues)

mDNS는 현재의 DNS와 동시성을 유지하기 위해서는 위에서 언급된 내용들이 들어있어야 한다. 즉 도메인 네임에서의 문자 데이터(character data) 표현의 국제적인 표준을 정하는 문제와 기존의 DNS 프로토콜과 새로운 mDNS 프로토콜과의 안전한 양립성 유지의 문제 해결 방안은 mDNS를 인터넷상에서 운영하는 데에 가장 핵심이 되는 주제라 할 수 있다.

3. mDNS 프로토콜 구현

3.1 IN 비트의 이용

IN 비트는 1비트로 구성되어 있으며 RFC 1035의 DNS 프로토콜에서 헤더 부분 포맷(Header format)에서 제안된 내용으로서 아직 그 사용이 명시되어 있지 않고 미래의 사용을 위하여 예약된 부분이다. 아래의 그림1을 보면 자세한 헤더 부분의 구조를 명시해 놓고 있다.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
ID															
QR	Opcode	AA	TC	RD	RA	IN	AD	CD	RCODE						
QDCOUNT															
ANCOUNT															
NSCOUNT															
ARCOUNT															

그림 1 DNS의 헤더 포맷

여기에서 제시된 IN 비트는 리졸버와 네임 서버측 모두에서 기존의 DNS 질문과 mDNS의 질문 내용을 구별하기 위하여 이용된다. 서버와 클라이언트 리졸버 부분에서 이용되는 데 우선 리졸버 부분에서는 기존의 DNS 프로토콜 질문이 들어오면, IN 비트를 0으로 만들고, 그렇지 않고 코드 엔코드 방식을 거쳐 UTF-8 인 경우에는 IN 비트를 1로 둔다. 서버인 경우에도 역시 IN 비트가 1인 경우와 그렇지 않은 경우로 나누어 처리하게 된다. 자세한 디자인 부분은 다음 단락에서

다루도록 하겠다.

3.2 mDNS 리졸버(resolver)구조

다국어 도메인 네임에서 사용할 코드 문자 집합(CCS)은 모든 지역 문자 집합을 Unicode로 전환한 뒤 사용할 수 있다. 이러한 Unicode가 들어오게 되면 현재의 상태를 저장시킨 후(UTF-8 변환 후 들어오기 때문에 기존에 있는 상태가 2개 이상 중첩될 수 있다.) Case folding을 시킨다. 기존의 DNS에서는 대소문자의 code의 변환이 쉬웠기 때문에 주로 서버에서 이루어졌으나, mDNS의 경우에는 이것의 변환이 기존의 DNS보다 Overhead가 존재하기 때문에 리졸버측에서도 변환해 준다. 이렇게 변환된 문자를 UTF-8으로 인코딩을 시킨 후 IN-비트를 1로 세팅시켜 네임 서버로 보낸다. 네임 서버에서 받은 응답은 다시 저장된 본래의 도메인 네임으로 바꾼 후(캐쉬 등의 저장공간에서 Lookup한다.) 응용프로그램으로 그 응답을 보낸다. 전체적인 구조는 그림 2와 같다.

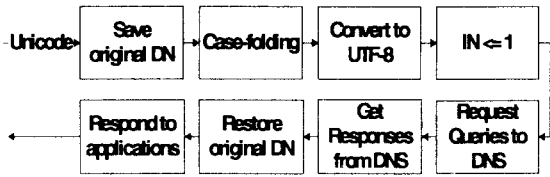


그림 2 mDNS의 리졸버 구조

3.3 mDNS 서버의 구조

mDNS에서의 운영은 기존의 DNS와 거의 유사하다. 차이점은 서버에서 받아들이는 wire format이 기존의 DNS가 보낸 ASCII 뿐만 아니라, mDNS에서 보낸 UTF-8 과 기타 다른 유사 형태가 공존한다. 이러한 것들은 IN 비트로 구별을 한다. 즉 IN 비트가 1인 경우에는 mDNS의 리졸버에서 보낸 것으로 인식하고 mDNS 네임 서버를 찾아 응답을 보내고 그렇지 않은 경우에는 유사한 형태의 wire format으로 인식하여, IN 비트를 1로 세팅한 후 원래의 질문을 저장한 후 대소문자 코드 변환을 한 후 기존의 DNS와 같은 동작을 한다. 응답을 받은 후에는 본래의 자신의 wire format으로 바꾼 후 응답을 보내게 된다. 기존의 DNS의 ASCII 인 경우에는 IN 비트를 인식하지 못하므로 mDNS가 아닌 기존의 DNS 네임 서버로 보내질 것이다. 전체적인 구조는 그림 3과 같다.

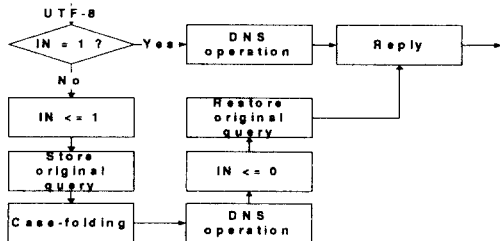


그림 3 mDNS의 서버구조

4. 결론

여기서 제시한 mDNS는 기존의 알파벳 문자만을 사용한 DNS 문제점을 해결하고 보다 쉬운 DNS 환경 구축을 위하여 제시되었다. 즉 기존의 DNS 헤더내의 IN 비트를 이용, IN 비트의 상태에 따라 mDNS와 기존의 DNS 그리고 다른 BIND를 지닌 DNS를 구별하여 처리하였다. 기존 DNS와의 호환성과 상호 운용을 위하여 IN 비트의 이용과 그에 따른 casefolding 문제의 해결 그리고 mDNS의 다양한 문자 집합의 사용을 위해 하나의 표준으로 정하여, 모든 문자 집합까지 포함할 수 있는 코드 문자 집합을 Unicode로 하고, Wire format을 코드 인코딩 집합인 UTF-8으로 한다. 리졸버에서 IN 비트의 상태를 정하고 그에 따라서 mDNS 서버측의 동작이 달라지게 된다. 그러나 질의의 방법이나 운영방식은 기존의 DNS와 같아지므로 현 DNS와 동시에 작동이 가능하여 기존의 클라이언트 시스템 및 기존 DNS 시스템의 변경을 최소화하는 기술을 제시하여 기존의 기술들이 클라이언트 시스템의 특수 응용프로그램에 의존적이던 단점을 해결하였다.

5. 참고 문헌

- [1] James Seng, "Requirements of Internationalized Domain Names," Internet Draft, June 2000
- [2] Stuart Kwan, "Using the UTF-8 Character Set in the Domain Name System," Internet Draft, February 2000
- [3] Dan Oscarsson, "Internationalisation of the Domain Name Service," Internet Draft, February 2000
- [4] Mockapetris, P., "Domain Names - Concepts and Facilities," STD 13, RFC 1034, USC/ISI, November 1987
- [5] Mockapetris, P., "Domain Names - Implementation and Specification," STD 13, RFC 1035, USC/ISI, November 1987
- [6] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, March 1997
- [7] C. Weider et. Al., "The Report of the IAB Character Set Workshop held 29 February - 1 March 1996," RFC 2130, Apr 1997.
- [8] F. Yergeau, "UTF-8, a transformation format of ISO 10646," RFC 2279, January 1998
- [9] D. Eastlake, "Domain Name System Security Extensions," RFC 2535, March 1999
- [10] The Unicode Consortium, "The Unicode Standard - Version 3.0," <http://www.unicode.org/unicode/>
- [11] M. Davis and M. Duerst, "Unicode Normalization Forms," Unicode Technical Report #15, Nov 1999, <http://www.unicode.org/unicode/reports/tr15/>
- [12] Mark Davis, "Case Mappings," Unicode Technical Report #21, May 2000, <http://www.unicode.org/unicode/reports/tr21>