TTA Standard

정보통신단체표준 제정일 : 2007년 11월 8일

TTAS.IF-RFC1035

도메인 네임 - 구현 및 스펙

(Domain Names - Implementation and Specification)

TTA 한국정보통신기술협회
Telecommunications Technology Association

정보통신단체표준 제정일 : 2007년 11월 8일
TTAS.IF-RFC1035

도메인 네임 - 구현 및 스펙

(Domain Names - Implementation and Specification)

TTA 한국정보통신기술협회
Telecommunications Technology Association

서 문

1. 표준의 목적

도메인 네임(Domain Names)은 서로 다른 호스트, 네트워크, 프로토콜 패밀리, 인터넷 및 관리 조직에 사용 가능한 방식으로 네이밍(Naming) 자원에 대한 메커니즘을 제공하는 것이다. 즉, IP주소(IPv4 및 IPv6)에 매핑되는 도메인 네임에 대한 정보를 가지는 도메인 네임 서버가 사용자로부터의 DNS 질의에 대한 응답을 보내주는 것이다. 네임 서버는 IP주소에 대한 DNS 질의는 해당되는 도메인 네임을 응답하고, 도메인 네임에 대한 DNS 질의는 해당되는 IP주소를 응답한다.

본 표준은 이러한 DNS를 구현하기 위한 구문, 도메인 네임 공간과 표준 리소스 레코드들에 대한 정의, DNS 질의·응답 등에 필요한 메시지 포맷 정의, 존 파일 구성, 리졸버 실행 등에 대한 구현 지침을 정의한다.

2. 주요 내용 요약

주요 내용으로는 DNS를 구현하기 위한 메시지 포맷, 프로토콜, 존 파일 구성, 리소스 레코드 정의 등에 대한 상세정보(Specification)를 정의한다.

3. 표준 적용 산업 분야 및 산업에 미치는 영향

본 표준은 국내 DNS 구현 및 운영에 대한 표준을 제안함으로써 국내 DNS 서비스 환경에서의 상호 연동성을 보장한다. 안정적인 국내 DNS 서비스 환경 제공으로 인터넷 기반의 다양한 응용서비스 활성화에 기여할 것이다.

4. 참조권고 및 표준

4.1 국외표준(권고)

- P. Mockapetris, "Domain Names - Implementation and Specification", RFC-1035, Nov. 1987

4.2 국내표준

해당사항 없음

5. 참조표준(권고)과의 비교

5.1 참조표준(권고)과의 관련성 : 동일함

5.2 참조한 표준(권고)과 본 표준의 비교표 : 없음

6. 지적재산권 관련사항

2007년 11월 현재까지 이 표준과 관련하여 확인된 지적 재산권은 없음

7. 적합인증 관련사항 : 해당사항 없음

8. 표준의 이력

판수	제/개정일	제·개정내역
제1판	2007. 11. 8	제정

Preface

1. The Purpose of Standard

The goal of domain names is to provide a mechanism for naming resources in such a way that the names are usable in different hosts, networks, protocol families, internets, and administrative organizations. I.e., Domain name server which has a IP address(IPv4 and IPv6) mapping information for domain name response the DNS queries from the users. The name servers response the related domain name for DNS query about IP address and response the related IP address for DNS query about domain name.

This standard defines the implemenation guideline for context of DNS implementation, domain name space, definition of standard resourse records, message format of DNS query/response, zone file configuration and resolver implementation.

2. The summary of contents

The main topics are defined with message format for DNS implementation, protocols, zone file configuration, definition of resource records and so on.

3. Applicable fields of industry and its effect

This standard guarantees the interoperability on the domestic DNS service environment throughout the proposal for domestic DNS implementations and operations. As supporting the stable domestic DNS service environment, It could contribute the activation of Internet based various applications.

4. Reference Recommendations and/or Standards

4.1 International Standards

- P. Mockapetris, "Domain Names - Implementation and Specification", RFC-1035, Nov. 1987

4.2 Domestic Standards : None

5. Relationship to International Standards(Recommendations)

5.1 The relationship of international standards : Same

5.2 Differences between International Standard(recommendation) and this standard : None

6. The Statement of Intellectual Property Rights : None

7. The Statement of Conformance Testing and Certification : None

8. The History of Standard

Edition	Issued date	Contents
The 1st edition	2007. 11 . 8	Established

목 차

Contents

도메인 네임 - 구현 및 스펙
Domain Names - Implementaion and Specification

1. 개요

1.1. 개론

도메인 네임의 목적은 네임은 서로 다른 호스트, 네트워크, 프로토콜 패밀리, 인터넷 및 관리 조직에 사용 가능한 방식으로 네이밍 자원에 대한 메커니즘을 제공하는 것이다.

사용자 관점에서, 도메인 네임은 도메인 네임과 관련된 정보를 얻는 리졸버, 즉 로컬 에이전트에 대한 논의 대상으로 유용하다. 이처럼 사용자는 특별한 도메인 네임과 관련된 호스트 주소 혹은 메일 정보를 요청한다. 사용자는 도메인 네임으로 리졸버에 특별한 타입의 정보, 즉 적절한 질의 타입을 보냄으로써 요청할 수 있다. 사용자에게, 도메인 네임 트리는 단일한 정보 공간이다; 리졸버는 사용자로부터 네임서버들 사이의 데이터 분산을 숨겨야 한다.

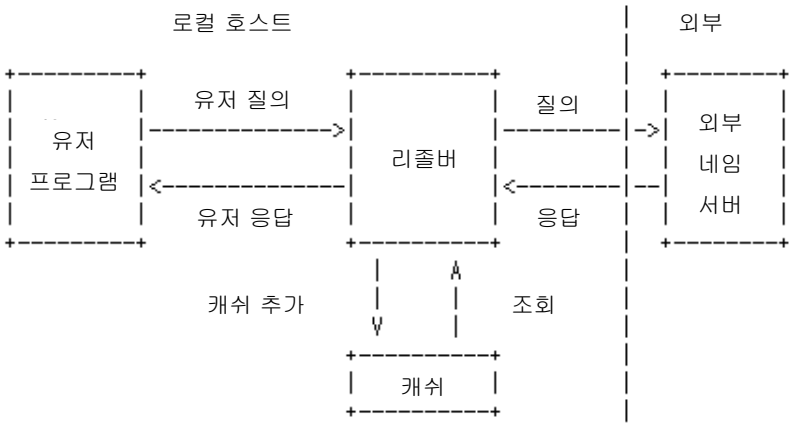
리졸버의 관점에서, 데이터베이스는 다양한 네임서버들간에 배분되어진 도메인 공간을 구성한다. 도메인 공간상의 서로 다른 부분은 서로 다른 네임서버들에 저장되며, 특별한 데이터 항목은 둘 혹은 그 이상의 네임서버에 반복적으로 저장된다. 리졸버는 최소한 하나의 네임서버의 정보를 갖고서 시작한다. 리졸버가 정보에 대해 알려진 네임서버에 요청하는 사용자 질의를 처리할 때, 리졸버는 차례로 원하는 정보를 얻거나 다른 네임서버에 조회한다. 이러한 조회의 사용으로 리졸버는 네임서버의 식별과 항목을 배운다. 리졸버는 도메인 공간의 배분을 다루어야 하고, 다른 서버에 있는 반복적인 데이터베이스 자문에 의한 네임서버 실패 효과를 처리해야 한다.

네임서버는 두종류의 데이터를 관리한다. 하나는 존이라 불리는 데이터이다; 각각의 존은 도메인 트리상의 특별히 “가지친(pruned)" 서브 트리에 대한 완전한 데이터베이스이다. 이러한 데이터는 authoritative라고 불린다. 네임서버는 주기적으로 존이 최신 정보인지를 체크하고 그렇지 않다면, 로컬하게 저장되어 있거나 혹은 다른 네임서버의 마스터 파일로부터 업데이트된 존의 새로운 복사본을 얻게된다.

두 번째 데이터는 로컬 리졸버로 얻은 캐시 데이터이다. 이 데이터는 아마도 완전하지는 않지만, non-로컬 데이터가 반복적으로 접근될때 수집 작업의 성능을 향상시킨다. 캐시 데이터는 언젠가는 타임아웃 메커니즘에 의해 버려진다.
이러한 기능적인 구조는 리졸버상의 사용자 인터페이스 문제, 실패 복구 및 분배를 고립시키며, 네임서버상의 데이터베이스 업데이트와 갱신을 고립시킨다.

1.2. 일반적인 구성

호스트는 다른 호스트로부터의 질의에 답하는 네임서버인 도메인 시스템으로부터 정보를 얻는 프로그램을 구동하는 호스트인지에 따라서 다수의 방법에 의해서 도메인 네임 시스템에 참여할 수 있다. 가장 간단하고 전형적인 구성은 아래와 같다:

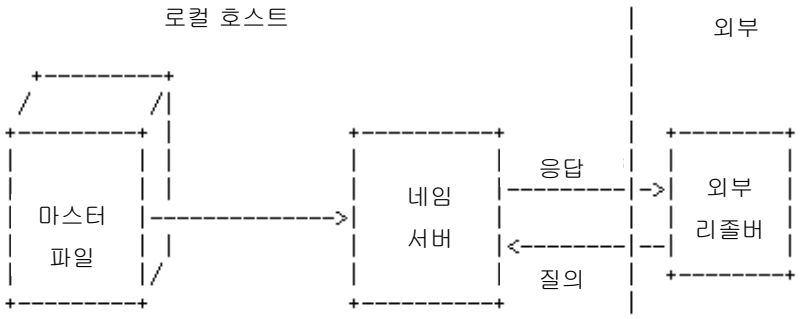


사용자 프로그램은 리졸버를 통해 도메인 네임 공간과 상호 작용한다; 사용자 질의와은 호스트와 운영체제에 따른다. 사용자 질의는 전형적으로 운영체제 호출에 의하고, 리졸버와 캐시는 호스트 운영체제의 한 부분이다. 성능이 떨어지는 호스트는 서비스를 필요로하는 모든 프로그램과 연계되는 서브 루틴으로써의 리졸버를 구현할 것이다.

리졸버는 외부의 네임서버에 질의해서 얻은 정보와 로컬 캐시로 사용자 질의에 응답한다.

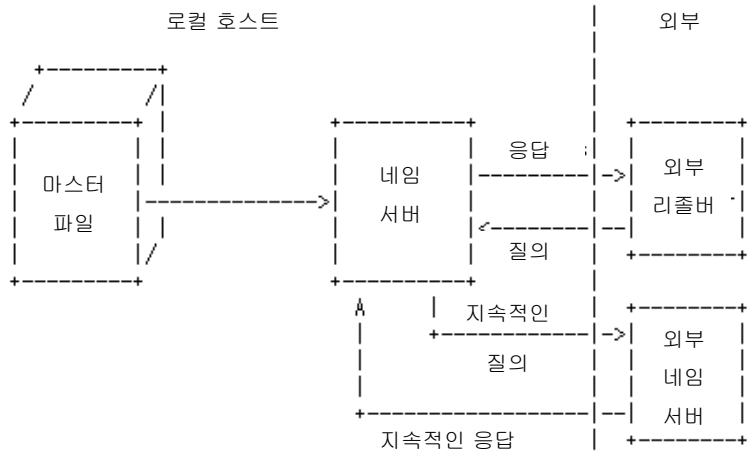
리졸버는 특별한 사용자 질의에 응답하는 다수의 서로 다른 외부 네임서버에 다수의 질의를 해야 할 것이고, 사용자 질의의 레졸루션은 다수의 네트워크 접근과 임의의 시간이 소요될 것이다. 외부 네임서버에 대한 질의와 그에 대한 응답은 여기서 묘사하는 표준 포멧을 가진다.

네임서버는 자신의 능력에 따라 전용 기계에 독립적인 프로그램이 되거나 큰 시간공유 호스트상의 프로세스를 처리한다. 간단한 구성은 아래와 같다.



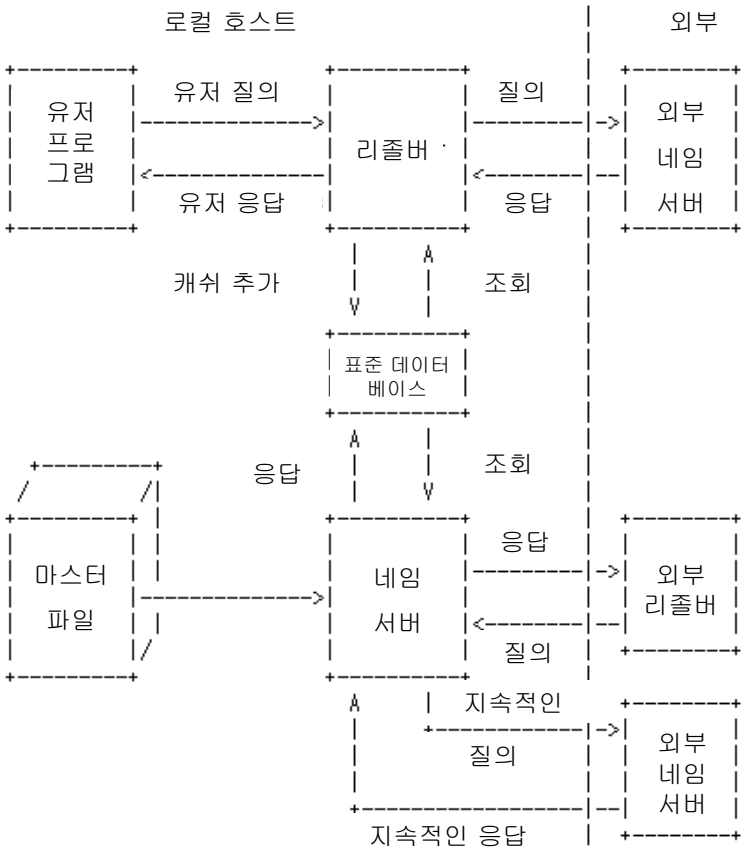
여기에 primary 네임서버는 자신의 로컬 파일 시스템으로부터 마스터 파일을 읽어들여 하나 혹은 그 이상의 존 정보를 얻어서 외부 리졸버로부터의 존에 대한 질의에 응답한다.

DNS는 하나 이상의 네임서버에 의해서 중복적으로 지원되는 존을 필요로 한다. 지정된 2차 네임서버는 DNS의 존 전달 프로토콜을 사용하여 primary 네임서버로부터 존을 얻고 업데이트를 위해 체크할 수 있다. 이러한 구성은 아래와 같다.



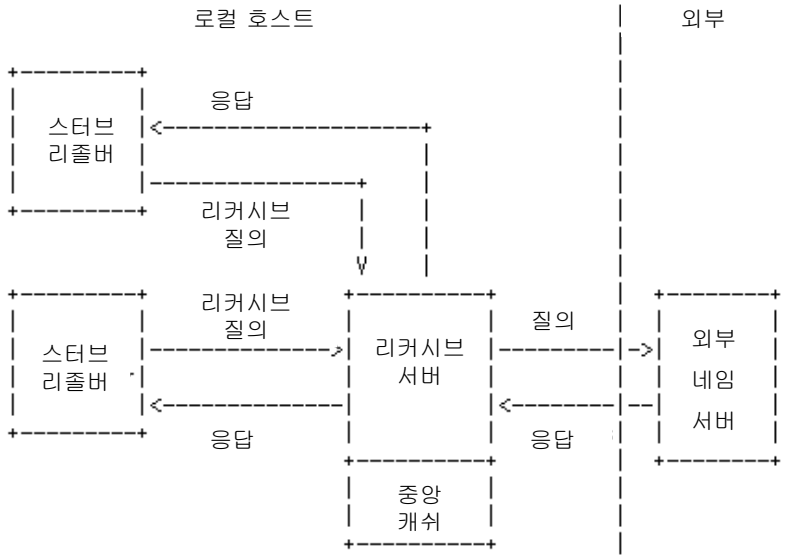
이러한 구성에서, 네임서버는 주기적으로 존의 복사본을 얻거나 기존 복사본이 변하지 않았음을 체크하는 외부 네임서버에 가상 회선을 구성한다. 이러한 상태관리를 위해 보내지는 메시지는 질의, 응답과 동일한 형태를 따른다. 그러나 메시지 순서는 좀 다르다.

도메인 네임 시스템의 모든 측면을 지원하는 호스트상의 정보 흐름은 아래와 같다.



공유 데이터베이스는 로컬 네임서버와 리졸버를 위한 도메인 공간 데이터를 갖고 있다. 공유 데이터베이스의 콘텐츠는 전형적으로 네임서버의 주기적인 갱신에 의해 관리되는 authoritative 데이터의 혼합이고 이전의 리졸버 요청으로부터 데이터를 캐시한다. 도메인 데이터의 구조와 네임서버와 리졸버간 동기화 필요성은 이러한 데이터베이스의 일반적인 특성을 암시하지만, 실제적인 포멧은 1로컬 구현자에 달려있다.

정보의 흐름 또한 천편일률적이지 않으며 호스트 그룹은 그들끼리 최적의 행위를 하게 된다. 때때로 이것은 성능이 좀 떨어지는 호스트의 부담을 덜어주게 되어 완전한 리졸버를 구현할 필요가 없게 된다. 이것은 PC 혹은 호스트가 필요로 하는 새로운 네트워크 코드의 양을 최소화하길 원할 경우에 적합하다. 이러한 설계는 또한 호스트 그룹으로 하여금 다량의 분리된 캐시를 유지하는 것보다 작은 수의 캐시를 공유하게끔 한다. 이러한 가정하에 중앙집중화된 캐시는 높은 히트율(hit ratio)을 가질 것이다. 어쨌든, 리졸버는 스터브 리졸버가 전면에 나서는 것을 대신하고 있으며 그러한 서비스를 수행하는 것으로 알려진 하나 혹은 그 이상의 네임서버상의 리커시브 네임서버에 위치한다.



어떤 경우든, 도메인 요소들은 가능한한 신뢰성을 위해 항상 복제된다는 점을 유의하라.

1.3. 규칙

도메인 시스템은 낮은 수준이지만 기본적이고 이슈사항을 다루기 위한 다수의 규칙을 갖고 있다. 개발자는 자신의 자체 시스템에 대해서는 이러한 규칙을 위한해도 무방하다. 그는 다른 호스트로부터 관찰되는 모든 행위상의 규칙을 관찰해야 한다.

1.3.1. 우선적인 네임 구문

DNS 스펙은 도메인 네임을 구성하기 위한 규칙상에서 가능한한 일반화된다. 그 개념은 기존의 어떠한 개체도 최소한의 변화로 도메인 네임으로 표현될 수 있다.

그렇지만, 개체에 대한 도메인 네임을 할당 시, 세심한 사용자는 도메인 시스템 규칙과 기존 프로그램에 의해 공개되거나 암시되었던지 간에 어떠한 기존 개체에 대한 규칙을 둘다 만족하는 네임을 선택할 것이다.

예를 들면, 메일 도메인을 네이밍할 때, 사용자는 이 메모와 RFC-822 모두를 만족해야 한다. 새로운 호스트 네임을 만들때, HOST.TXT의 오래된 규칙이 따를 것이다. 이것은 오래된 소프트웨어가 도메인 네임을 사용하기 위해 변경되는 것을 피하기 위함이다.

다음 구문은 도메인 네임을 사용하는 많은 응용(예, 메일, 텔넷)과 함께 소수의 문제점을 나타낼 것이다.

<domain> ::= <subdomain> | " "

<subdomain> ::= <label> | <subdomain> "." <label>

<label> ::= <letter> [[<ldh-str>] <let-dig>]

<ldh-str> ::= <let-dig-hyp> | <let-dig-hyp> <ldh-str>

<let-dig-hyp> ::= <let-dig> | "-"

<let-dig> ::= <letter> | <digit>

<letter> ::= A부터 Z까지의 대,소문자 52개 알파벳 문자들 중 하나

<digit> ::= 0부터 9까지 10개의 숫자 중의 하나

도메인 네임에서 대문자와 소문자는 허용되며, 어떠한 의미도 부여되지 않는다는 것을 유의하라. 즉, 동일한 철자상의 두 네임은 다른 활자 케이스이나 동일한 것으로 간주된다.

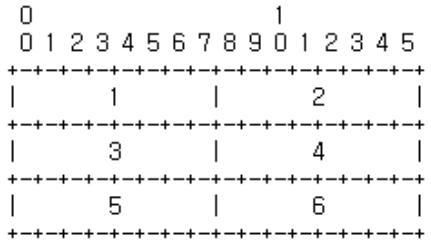
라벨은 ARPANET 호스트 네임에 대한 규칙을 따른다. 그것은 문자로 시작하며 문자 혹은 숫자로 끝나며, 내부 문자로써 단지 문자, 숫자, 하이픈을 가진다. 또한 다소 길이의 제한이 있다. 라벨은 63 문자이거나 그 이하여야 한다.

예를 들면, 다음의 문자열은 인터넷상에서 호스트를 식별한다:

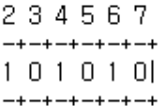
A.ISI.EDU XX.LCS.MIT.EDU SRI-NIC.ARPA

1.3.2. 데이터 전송 순서

이 문서에서 묘사되는 헤더와 데이터의 전송 순서는 옥텟 레벨을 수행한다. 다이어그램은 옥텟 그룹을 보여주고, 이러한 옥텟의 전송 순서는 영어로 읽혀지는 일반적인 순서이다. 예를 들면, 다음의 다이어그램에서, 옥텟은 그것의 번호 순서에 따라 전송된다.



옥텟이 숫자의 양을 대표할 때, 다이어그램의 가장 왼쪽의 비트가 높은 순서거나 가장 의미있는 비트이다. 즉, 비트 라벨 0은 가장 의미있는 비트이다. 예를 들면, 다음의 다이어그램은 값 170(10진수)을 대표한다.



유사하게, 멀티 옥텟 필드가 숫자의 양을 대표할 때, 전체 필드의 가장 왼쪽의 비트는 가장 의미있는 비트이다. 멀티 옥텟이 전송될때, 가장 의미 있는 옥텟이 먼저 전송된다.

1.3.3. 문자 케이스

공식적인 프로토콜의 일부분인 DNS의 모든 부분에 대해, 문자열(예, 라벨, 도메인 네임, 등)사이의 모든 비교는 활자 케이스 집중적인 특성하에 수행된다. 현재, 이러한 규칙은 예외없이 도메인 시스템을 통해 유효하다. 그렇지만, 현재 사용하는 것 이상의 향후 추가분은 네임상의 완전한 바이너리 옥텟 용량 사용을 필요로 하고, 7 비트 ASCIII 상의 도메인 네임을 저장하거나 특수한 바이트를 이용해서 라벨을 끝내려고 하는 것은 금지되어야 한다.

데이터가 도메인 시스템으로 유입될때, 가능한한 원래 활자 케이스를 보전하려 한다. 어떤 환경에서는 이것이 수행되지 않을 수 있다. 예를 들면, 두개의 RR이 데이터베이스에 저장되면, x.y상의 하나와 X.Y상의 하나는 데이터베이스상의 동일 장소에 실질적으로 저장된다. 그리고 단지 하나의 활자 케이스가 보존될 것이다. 기본 규칙은 데이터가 데이터베이스상에서 구조를 정의하기 위해 사용될때, 활자 케이스는 폐기될 수 있으며, 활자 케이스에 둔감한 방식에서 비교될때는 두개의 네임은 동일한 것이다.

활자 케이스 민감성 데이터의 분실은 최소화되어야 한다. 이처럼 x.y와 X.Y에 대한 데이터는 x.y 혹은 X.Y의 단일 위치에 저장되고 a.x와 B.X에 대한 데이터는 A.x, A.X, b.x 혹은 b.X로 저장되어서는 안된다. 일반적인 경우, 이것은 도메인 네임의 첫 번째 라벨의 활자 케이스를 보존하지만 내부 노드 라벨의 표준화를 강요하고 있다.
도메인 데이터베이스에 데이터를 입력하는 시스템 관리자들은 그들 시스템이 활자 케이스 민감한 경우 활자 케이스 일치 방식상의 도메인 시스템을 제공하는 데이터를 표현하는데 주의해야 한다. 도메인 시스템상의 데이터 분산 시스템은 일치하는 표현이 보존되도록 할 것이다.

1.3.4. Size limits

DNS상의 다양한 개체와 파라미터는 사이즈 한계를 갖는다. 그것은 아래와 같다. 어떤 것은 쉽게 변하고, 다른 것은 더 기본적이다.

labels	63 옥텟 이하
names	254 옥텟 이하
TTL	singed 32 비트 숫자의 양수
UDP messages	512 옥텟 이하

2. 도메인 네임 공간과 RR 정의

2.1. 도메인 공간 정의

메시지상의 도메인 네임은 라벨들의 연속된 형태로 표현된다. 각각의 라벨은 다수의 옥텟을 따르는 하나의 옥텟 길이 필드로써 표현된다. 모든 도메인 네임이 루트의 널(null) 라벨로 끝날 때, 도메인 네임은 제로 바이트 길이로 끝난다. 모든 길이 옥텟의 최상위 2 비트는 제로이어야 하고, 길이 필드의 남는 6 비트는 라벨을 63 옥텟 혹은 그 이하로 제한한다.

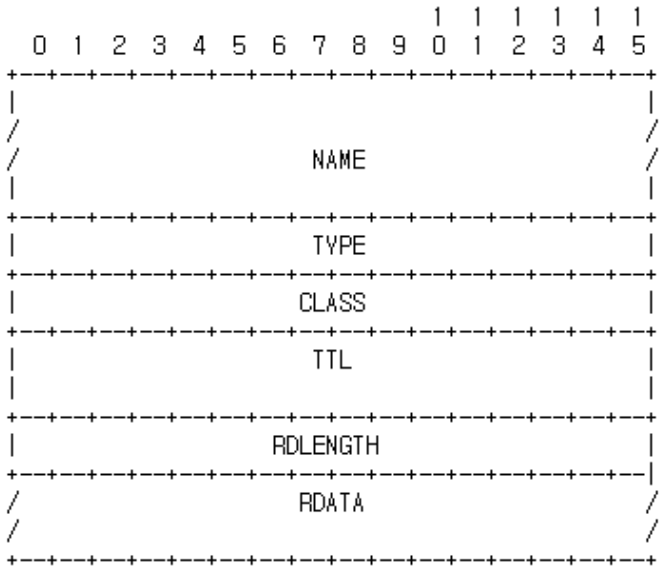
구현의 간소화를 위해, 도메인 네임의 총 길이(즉, 라벨 옥텟과 라벨 길이 옥텟)는 255 옥텟 혹은 그 이하로 제한된다.

비록 라벨은 라벨을 구성하는 옥텟상의 8 비트 값을 가질 수 있지만, 이 메모에서 묘사된 우선적인 구문을 따르는 라벨을 강력히 추천하며, 그것은 기존 호스트 네이밍 규칙과 조화를 이룬다. 네임서버와 리졸버는 ASCII를 가정한 활자 케이스 둔감성 방식(즉, A=a) 상의 라벨과 제로 패리티를 비교해야 한다. 비알파벳 코드는 정확하게 매칭되야 한다.

2.2. RR 정의

2.2.1. 포멧

모든 RR은 아래와 같은 최고 레벨 포멧과 동일하다.



where:

NAME 소유자 네임, 즉, 이러한 리소스 레코드가 속하고 있는 노드의 네임

TYPE RR 타입 코드의 하나를 갖는 두개의 옥텟

CLASS RR CLASS 코드의 하나를 갖는 두개의 옥텟

TTL 정보의 소스를 다시 참고하기 전 리소스 레코드가 캐시되는 시간 간격을 나타내는 32비트 singed integer. 제로 값은 단지 진행상의 처리를 위해 사용될 수 있는 RR을 의미하는 것으로 해석되고, 캐시되지 않는다. 예를 들어, SOA 레코드는 캐시를 금지하기 위해 제로 TTL로 항상 분배된다. 제로 값은 또한 궁극적으로 휘발성 데이터로 사용되어 질 수 있다.

RDLENGTH RDATA 필드의 옥텟 길이를 나타내는 unsigned 16 비트 integer

RDATA 리소스를 나타내는 가변 길이 옥텟 문자열. 리소스 레코드의 TYPE과 CLASS에 따라 정보 포apt이 변한다.

2.2.2. TYPE 값

TYPE 필드는 리소스 레코드내에서 사용된다. 이러한 타입은 QTYPE의 서브셋임을 주의하라.

TYPE 값과 의미

A	1 호스트 주소
NS	2 authoritative 네임서버
MD	3 메일 목적지(폐기 - MX 사용)
MF	4 메일 포워더(폐기 - MX 사용)
CNAME	5 별칭을 위한 캐노니컬 네임
SOA	6 권위있는 존의 시작을 표시
MB	7 메일박스 도메인 네임(실험용)
MG	8 메일 그룹 멤버(실험용)
MR	9 메일 리네임 도메인 네임(실험용)
NULL	10 a null RR (EXPERIMENTAL)
WKS	11 잘 알려진 서비스 묘사
PTR	12 도메인 네임 포인터
HINFO	13 호스트 정보
MINFO	14 메일박스 혹은 메일 리스트 정보
MX	15 메일 교환
TXT	16 텍스트 문자열

2.2.3. QTYPE 값

QTYPE 필드는 질의의 질문 부분에 나타난다. QTYPES는 TYPE의 슈퍼셋이고, 모든 TYPE은 유효한 QTYPE이다. 게다가, 다음은 QTYPE을 정의한다.

AXFR	252 전체 존의 전달 요구

MAILB 253 메일박스와 관련된 레코드(MB, MG 혹은 MG)의 요구

MAILA 254 메일 에이전트 RR(폐기- 보기MX) 의 요구

* 255 모든 레코드의 요구

2.2.4. CLASS 값

CLASS 필드는 리소스 레코드에서 나타난다. 다음의 CLASS 기억술과 값을 정의한다.

IN 1 인터넷

CS 2 CSNET class(폐기 - 다른 폐기된 RFC의 예로써 단지 사용)

CH 3 CHAOS class

HS 4 Hesiod [Dyer 87]

2.2.5. QCLASS 값

QCLASS 필드는 질의의 질문 섹션에 나타난다. QCLASS 값은 CLASS 값의 슈퍼셋이다; 모든 CLASS는 유효한 QCLASS이다. CLASS 값에 첨가하여, 다음의 QCLASS가 정의된다.

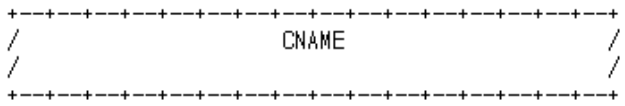
* 255 어떠한 클래스

2.3. 표준 RRs

다음의 RR 정의는 최소한 잠재적으로 모든 클래스에서 나타날 것으로 예상된다. 특히, NS, SOA, CNAME 및 PTR은 모든 클래스에서 사용되며, 모든 클래스에서 동일한 포멧을 가진다. RDATA 포멧은 알려졌기 때문에, 이런 RR의 RDATA 섹션상의 모든 도메인 네임은 압축될 수 있다.

<도메인-네임>은 일련의 라벨로써 대표되는 도메인 네임이고, 제로 길이의 라벨에 의해 끝난다. <문자-열>은 이진수 정보에 의해 다루어지고, 길이상 256 문자까지 될 수 있다.(길이 옥텟 포함)

2.3.1. CNAME RDATA 포멧

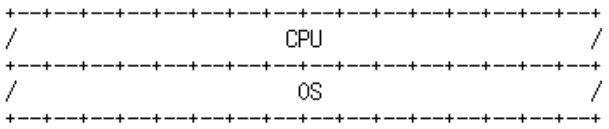


여기서:

CNAME <도메인-네임>소유자를 위한 canonical 혹은 primary 네임. 소유자 네임은 별칭이다.

CNAME RR은 additional 섹션 처리의 원인이 되지 않지만, 네임서버는 어떤 경우간에 canonical 네임에서 질의 시작을 선택할 수 있다. 자세한 사항은 네임서버 로직 [RFC-1034]의 기술을 보라.

2.3.2. HINFO RDATA 포멧



여기서:

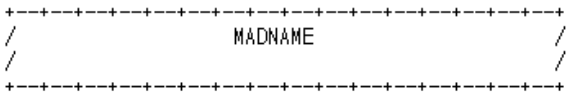
CPU <문자-열>은 CPU 타입을 상술한다.

OS <문자-열>은 운영체제 타입을 상술한다.

CPU와 운영체제에 대한 표준 값은 [RFC-1010]에서 찾을 수 있다.

HINFO 레코드는 호스트에 대한 일반적인 정보를 얻는데 사용된다. 주된 사용은 기계 혹은 동일한 타입의 운영체가 통신 시 특별한 절차를 사용할 수 있는 FTP와 같은 프로토콜을 위함이다.

2.3.3. MB RDATA 포멧(실험용)

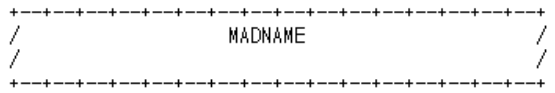


여기서:

MADNAME <도메인-네임>은 특별한 메일박스를 갖는 호스트를 상술한다.

MB 레코드는 MADNAME에 대응하는 A 타입 RR을 검색하는 additional 섹션 절차의 원인이 된다.

2.3.4. MD RDATA 포멧(폐기)



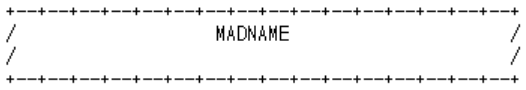
여기서:

MADNAME <도메인-네임>도메인에 대한 메일을 전달할 수 있는 도메인에 대한 메일 에이전트를 갖는 호스트를 상술한다.

MD 레코드는 MADNAME에 대응하는 A 타입 RR을 검색하는 additional 섹션 절차의 원인이 된다.

MD는 폐기되었다. 새로운 설계에 대한 상세내용은 MX 정의와 [RFC-974]를 참조하라. 마스터 파일에서 발견되는 MD RR을 처리하기 위한 권고 정책은 그것들을 제거하는 것이거나, 그것들을 preference 0로 MX RR을 변경하는 것이다.

2.3.5. MF RDATA 포멧(폐기)



여기서:

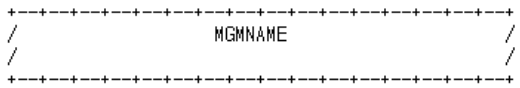
MADNAME <도메인-네임>도메인에 포워딩을 위한 메일을 수용하는 도메인에 대한 메일 에이전트를 갖는 호스트를 상술한다.

MF 레코드는 MADNAME에 대응하는 A 타입 RR을 검색하는 additional 섹션 절차의 원인이 된다.

MF는 폐기되었다. 새로운 설계에 대한 상세내용은 MX 정의와 [RFC-974]를 참조하라.

마스터 파일에서 발견되는 MD RR을 처리하기 위한 권고 정책은 그것들을 제거하는 것이거나, 그것들을 preference 10로 MX RR을 변경하는 것이다.

2.3.6. MG RDATA 포멧(실험용)

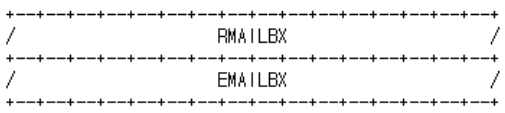


여기서:

MGMNAME <도메인-네임> 도메인 네임에 의해 상술된 메일 그룹의 멤버인 메일박스를 상술한다.

MG 레코드는 additional 섹션 처리의 원인이 되지 않는다.

2.3.7. MINFO RDATA 포멧(실험용)



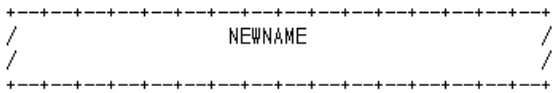
여기서:

RMAILBX <도메인-네임>메일링 리스트 혹은 메일박스에 책임을 갖는 메일박스를 상술한다. 이런 도메인 네임이 루트를 명명하면, MINFO RR의 소유자는 그자체에 대한 책임을 갖는다. 많은 기존의 메일링 리스트는 메일링 리스트 X(즉, Msgroup에 대한Msggroup-request)의 RMAILBX에 대한 메일박스 X-request를 사용한다. 이 필드는 더 많은 일반적인 메커니즘을 제공한다.

EMAILBX <도메인-네임> 메일일 리스트 혹은 MINFO RR(ERRORS-TO에 유사: 제안된 필드)의 소유자에 의해 상술된 메일박스에 관련된 에러 메시지를 받는 메일박스를 상술한다. 이러한 도메인 네임이 루트를 명명하면, 에러는 메시지의 송출자에게 되돌아 간다.

MINFO 레코드는 additional 섹션 처리의 원인이 되지 않는다. 비록 이러한 레코드가 간단한 메일박스와 연관이 있을 수 있지만, 그것은 일반적으로 메일링 리스트로 사용된다.

2.3.8. MR RDATA 포멧(실험용)

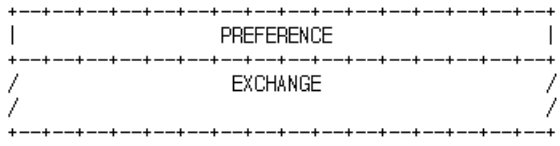


여기서:

NEWNAME <도메인-네임> 특별한 메일박스의 적절히 리네임된 메일 박스를 상술한다.

MR 레코드는 additional 섹션 처리의 원인이 되지 않는다. MR의 주된 사용은 다른 메일박스로 옮긴 사용자를 위한 포워딩 엔트리이다.

2.3.9. MX RDATA 포멧



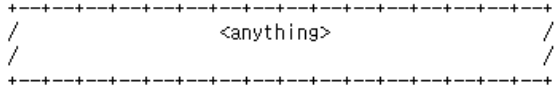
여기서:

PREFERENCE 동일 소유자들 사이의 RR에 부여된 preference를 상술하는 16 비트 정수. 낮은 값이 선호된다.

EXCHANGE <도메인-네임>소유자 이름을 위한 메일 교환으로써 기꺼이 행하는 호스트를 상술한다.

MX 레코드는 EXCHANGE에 의해 상술된 호스트를 위한 additional 섹션 처리에 의한다. MX RR의 사용은 [RFC-974]에서 자세히 설명되어 있다.

2.3.10. NULL RDATA 포멧(실험용)

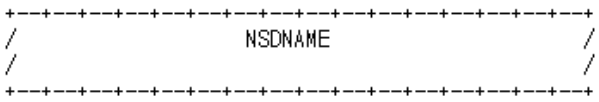


65535 옥텟 혹은 이하 일때 어떤 것도 RDATA가 될 수 있다.

NULL 레코드는 additional 섹션 처리의 원인이 되지 않는다. NULL RR은 마스터 파일

내에 허용되지 않는다. NULL은 DNS의 실험적인 확장으로 장소를 차지하는데 사용된다.

2.3.11. NS RDATA 포멧



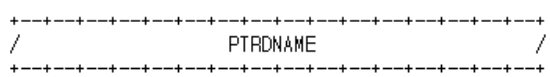
여기서:

NSDNAME <도메인-네임>특별한 클래스와 도메인에 대해 authoritative한 호스트를 상술한다.

NS 레코드는 A타입 레코드를 가리키는 일반적인 additional 섹션 처리와 참조(referral)에 사용될 때, 글루 정보에 존재하는 존을 특별히 찾을 경우의 근거가 된다.

NS RR은 named 호스트가 특별한 클래스의 소유자 네임에서 시작하는 존을 갖는 것을 상술한다. 비록 그것이 전형적으로 강력한 힌트일지라도, 클래스가 호스트와 통신하는데 사용되는 프로토콜 패밀리를 가리키지 않는 것을 주의하라. 예를 들면, 인터넷(IN) 혹은 Hesiod(HS) 클래스 정보에 대한 네임서버인 호스트가 일반적으로 IN 클래스 프로토콜을 사용하는 질의를 받는다.

2.3.12. PTR RDATA 포멧

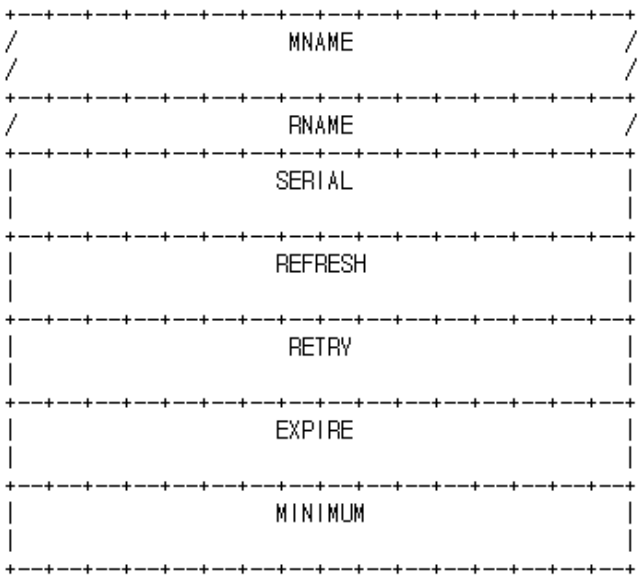


여기서:

PTRDNAME <도메인-네임> 도메인 네임 공간상에서 어떤 위치를 가리킨다.

PTR 레코드는 additional 섹션 처리의 근거가 되지 않는다. 이러한 RR은 도메인 공간상의 어떤 다른 위치를 지시하는 특별한 도메인으로 사용된다. 이러한 레코드는 간단한 데이터이고, 별칭(alias)을 식별하는 CNAME에 의한 수행과 유사한 어떤 특별한 처리를 암시하지 않는다. 예로써 IN-ADDR.ARPA 도메인의 서술을 보라.

2.3.13. SOA RDATA 포멧



여기서:

MNAME 존에 대한 원본 혹은 primary 소스인 네임서버의<도메인-네임>

RNAME <도메인-네임>이러한 존에 책임있는 사람의 메일박스를 상술한다.

SERIAL 그 존의 원본 복사의 unsigned 32 비트 숫자이다. 존 전달은 이 값을 보존한다. 이 값은 덮고 순서 공간 산술을 사용하는 것에 비교될 수 있다.

REFRESH 존이 리플레쉬 되기 전의 32비트 시간 간격

RETRY 실패된 리플레쉬가 재시도 하기 이전의 경과된 32비트 시간 간격

EXPIRE 존이 더 이상 authoritative가 아니기 이전의 경과된 시간 간격상의 상위 한계를 상술한다.

MINIMUM 이러한 존으로부터 어떠한 RR이 나가도록 하는 unsigned 32비트 최소 TTL 필드

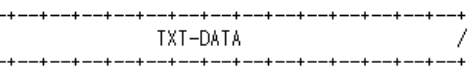
SOA 레코드는 additional 섹션 처리의 근거가 되지 않는다.

모든 시간은 초단위이다.

이러한 필드의 대부분은 네임서버 관리 운영에 적절하다. 그렇지만, MINIMUM은 존에서

RR을 받는 모든 질의 작용에서 사용된다. RR이 질의에 대한 응답에 보내질때마다, TTL 필드는 RR과 적절한 SOA내의 MINIMUM 필드로부터 TTL 필드의 최고치로 설정된다. 이처럼 MINIMUM은 존 상의 모든 RR에 대한 TTL 필드에 더 낮게 되어있다. 이러한 MINIMUM의 사용은 존이 마스터 파일로부터 혹은 존 전달로 로드될때, RR이 응답에 복사가 되던 안되던지 간에 발생할 것이다. 이러한 규정의 이유는 향후 다이나믹 업데이트 시설이 알려진 구문으로 SOA RR을 변경하도록 하기 위함이다.

2.3.14. TXT RDATA 포멧



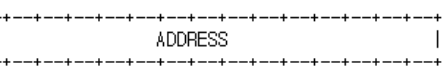
여기서:

TXT-DATA　　하나 혹은 그 이상의 <문자-열>

TXT RR은 묘사해야 할 텍스트를 유지하는데 사용한다. 텍스트의 구문은 그것이 발견될 도메인에 의존한다.

2.4. Internet specific RRs

2.4.1. A RDATA 포멧

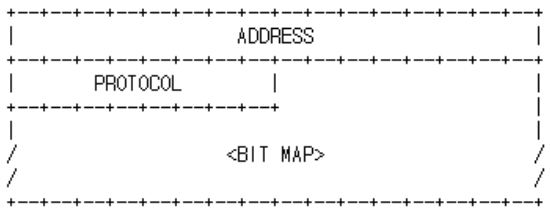


여기서:
ADDRESS　　32 비트 인터넷 주소

다수의 인터넷 주소를 갖는 호스트는 다수의 A 레코드 가진다.

A 레코드는 additional 섹션 처리의 근거가 아니다. 마스터 파일상의 A 라인의 RDATA 섹션은 어떠한 삽입되는 공간없이 점에 의해 구분되는 4개의 10진수 숫자로 표현되는 인터넷 주소이다.(즉, “10.2.0.52” 혹은 “192.0.5.6”)

2.4.2. WKS RDATA 포멧



여기서:

ADDRESS 32 비트 인터넷 주소

PROTOCOL 8 비트 IP 프로토콜 숫자

<BIT MAP> 가변 길이 비트맵. 이 비트맵은 다수의 8 비트 이상이어야 한다.

WKS 레코드는 특별한 인터넷 주소상의 특별한 프로토콜에 의해 지원되는 잘 알려진 서비스를 묘사하는데 사용된다. PROTOCOL 필드는 IP 프로토콜 숫자를 상술하고, 비트맵은 특별한 프로토콜의 포트별 1비트를 가진다. 첫 번째 비트는 포트 0에 해당하고, 두 번째는 포트 1, 등. 만약 비트맵이 관심 프로토콜에 대한 비트를 포함하지 않으면 그 비트는 제로로 가정한다. 포트와 프로토콜에 대한 적절한 값과 기억술은 [RFC-1010]에 서술되어 있다.

예를 들면, 만약 PROTOCOL=TCP(6) 이면, 26번째 비트는 TCP 포트 25(SMTP)에 해당한다. 만약 이 비트가 설정되면, SMTP 서버는 TCP 포트 25를 경청(listen)하고 있어야 한다. 만약 0이면, SMTP 서비스는 특별한 주소상에서 지원되지 않는다.

WKS RR의 목적은 서버에 대해 TCP와 UDP의 유효한 정보를 제공하기 위함이다. 만약 서버가 TCP와 UDP를 지원하거나, 다수의 인터넷 주소를 가지면 다수의 WKS RR이 사용된다.

WKS RR은 additional 섹션 처리의 근거가 아니다.

마스터 파일에서, 포트와 프로토콜은 기억술 혹은 10진수 사용을 통해 표현된다.

2.5. IN-ADDR.ARPA 도메인

인터넷은 특별한 도메인이 게이트웨이 위치를 지원하고 호스트에 매핑되는 인터넷 주소를 사용한다. 다른 클래스는 다른 도메인상에서 유사한 전략을 채용할 것이다. 이러한 도메인의 의도는 호스트 주소를 호스트 네임에 대한 매핑을 수행하는 보장된 방법을 제공하기 위함이다. 그리고, 인터넷상의 특별한 네트워크의 모든 게이트웨이에 질의되도록 하

기 위해서이다.

이러한 서비스가 인버스 질의에 의해 수행될 수 있는 기능과 비슷하다는 것과; 차이점은 도메인 네임 공간의 이러한 부분이 주소에 따라 구성되는 것이다, 그래서 도메인 공간을 찾는 낭비없이 적당한 데이터가 위치할 수 있도록 보장할 수 있다.

도메인은 IN-ADDR.ARPA에서 시작하고 인터넷 주소 체계를 따르는 서브체계를 갖는다.

IN-ADDR.ARPA 도메인 상의 도메인 네임은 IN-ADDR. ARPA 접미사에 4개 라벨을 갖도록 정의된다. 각각의 라벨은 인터넷 주소의 한개 옥텟을 대표하고 0-255범위의 10진수에 대한 문자열을 표현한다.(단일한 제로(0)을 나타내는 제로(0) 옥텟의 경우를 제외하고 앞서는 제로(0)은 생략된다.)

호스트 주소는 4개 라벨 모두를 갖는 도메인 네임에 의해 표현된다. 이처럼 인터넷 주소 10.2.0.52에 대한 데이터는 도메인 네임 52.0.2.10.IN-ADDR.ARPA에 위치한다. 반대의 경우는, 비록 읽기에는 서투른지만, 주소 공간의 하나의 네트워크로 정확하게 위임되는 존을 허락한다. 예를 들면, 26.IN-ADDR.ARPA이 MILNET에 대한 분리된 존이 된다면, 10.IN-ADDR.ARPA는 ARPANET에 대한 데이터를 포함하는 존이 될 수 있다. 주소 노드는 일반적인 도메인 공간상의 primary 호스트 네임에 대한 포인터를 갖는데 사용된다.

네트워크 숫자들은 IN-ADDR.ARPA 도메인상의 다양한 깊이에 있는 어떠한 비터미날 노드에 대응된다. 인터넷 네트워크 숫자는 1, 2 혹은 3 옥텟중의 하나이다. 네트워크 노드는 그 네트워크에 연결된 게이트웨이의 primary 호스트 네임에 대한 포인터를 갖는데 사용된다. 정의에 의해 하나 이상의 네트워크 상의 게이트웨이가 있다면, 그것은 전형적으로 두개 혹은 그 이상의 네트워크 노들을 가질 것이다. 게이트웨이는 또한 그것은 fully qualified 주소상의 호스트 레벨 포인터를 갖는다.

네트워크 노드상의 게이트웨이 포인터와 완전한 주소 노드상의 일반적인 호스트 포인터는 그것에 대응되는 호스트의 primary 도메인 네임에 대한 포인트를 하기 위해 PTR RR을 사용한다.

예를 들면, IN-ADDR.ARPA 도메인은 net 10과 26 사이의 ISI 게이트웨이, net 10에서 net 18까지의 MIT 게이트웨이와 호스트 A.ISI.EDU와 MULTICS.MIT.EDU.에 대한 정보를 가질 것이다.

ISI 게이트웨이가 주소 10.2.0.22와 26.0.0.103, 그리고 네임 MILNET-GW.ISI.EDU를 가지고, MIT 게이트웨이가 주소 10.0.0.77과 18.10.0.4, 그리고 네임 GW.LCS.MIT.EDU를 가지고 도메인 데이터베이스가 다음을 포함한다:

```
10.IN-ADDR.ARPA.            PTR MILNET-GW.ISI.EDU.
10.IN-ADDR.ARPA.            PTR GW.LCS.MIT.EDU.
18.IN-ADDR.ARPA.            PTR GW.LCS.MIT.EDU.
26.IN-ADDR.ARPA.            PTR MILNET-GW.ISI.EDU.
22.0.2.10.IN-ADDR.ARPA.     PTR MILNET-GW.ISI.EDU.
103.0.0.26.IN-ADDR.ARPA.    PTR MILNET-GW.ISI.EDU.
77.0.0.10.IN-ADDR.ARPA.     PTR GW.LCS.MIT.EDU.
4.0.10.18.IN-ADDR.ARPA.     PTR GW.LCS.MIT.EDU.
103.0.3.26.IN-ADDR.ARPA.    PTR A.ISI.EDU.
6.0.0.10.IN-ADDR.ARPA.      PTR MULTICS.MIT.EDU.
```

이처럼 net 10상의 게이트웨이를 놓고자 하는 프로그램은 QTYPE=PTR, QCLASS=IN, QNAME=10.IN-ADDR.ARPA의 형식으로 질의를 발생시킬 것이다. 그것은 다음과 같은 두가지 RR을 응답으로 받을 것이다:

```
10.IN-ADDR.ARPA.            PTR MILNET-GW.ISI.EDU.
10.IN-ADDR.ARPA.            PTR GW.LCS.MIT.EDU.
```

그러면 프로그램은 이러한 게이트웨이의 인터넷 주소를 얻기 위해 MILNET-GW.ISI.EDU. 과 GW.LCS.MIT.EDU.에 대한 QTYPE=A, QCLASS=IN 질의를 발생한다.

인터넷 호스트 주소 10.0.0.6에 대응하는 호스트 네임을 찾고자 하는 리졸버는 QTYPE=PTR, QCLASS=IN, QNAME=6.0.0.10.IN-ADDR.ARPA 형식의 질의를 추구하고 다음과 같이 받게 된다:

```
6.0.0.10.IN-ADDR.ARPA.      PTR MULTICS.MIT.EDU.
```

다수의 주의사항은 이러한 서비스 사용에 적용된다:

- IN-ADDR.ARPA 특수 도메인과 특별한 호스트 혹은 게이트웨이에 대한 일반적인 도메인이 다른 존에 있을 때, 데이터가 불일치하는 가능성이 존재하게 된다.

- 게이트웨이는 종종 분리된 도메인상에서 두개의 네임을 가지며, 단지 그것 중 하나가 primary가 될 수 있다.

- 라우팅 테이블을 초기화하기 위한 도메인 데이터베이스를 사용하는 시스템은 그것이 적절한 네임서버로 접근할 수 있는 보증된 충분한 게이트웨이 정보를 가지고 기동해야 한다.

- 게이트웨이 데이터는 단지 현재의 HOST.TXT 파일에 동등한 방식하에 게이트웨이 존재를 반영한다. 그것은 GGP 혹은 EGP로부터의 다이나믹한 유효한 정보를 바꾸지 못한다.

2.6. 새로운 타입의 정의, 클래스 및 특별한 네임공간

이전에 정의된 타입과 클래스들은 이 메모의 날짜로써 사용되는 것들이다. 새로운 정의가 기대된다. 이러한 섹션은 기존의 시설에 대한 추가를 고려하는 디자이너에게 어떠한 권고를 하게 된다. 다음 메일링 리스트(NAMEDROPPERS@SRI-NIC.ARPA) 는 일반적인 디자인 이슈가 발생되는 포럼이다.

일반적으로, 새로운 타입은 새로운 정보가 기존 개체에 대한 데이터베이스에 추가되거나, 혹은 어떠한 완전히 새로운 개체를 위한 새로운 데이터 포멧을 필요로 할때 적절하다. 디자이너는 타입을 정의하기를 시도할 것이고 모든 클래스에 일반적으로 적합한 RDATA 포멧은 정보의 중복을 피한다. 새로운 클래스는 DNS가 새로운 프로토콜 등을 위해 사용되고, 새로운 클래스에 특별한 데이터 포멧을 요구하거나 기존 네임 공간의 복사가 요구될때 적절하지만, 분리된 관리 도메인을 필요하다.

새로운 타입과 클래스는 마스터 파일을 위한 기억술이 필요하다; 마스터 파일의 포멧은 타입과 클래스가 분리되는 기억술을 필요로 한다.

타입과 클래스 값은 각각 QTYPE과 QCLASS의 적절한 서브셋이 되어야 한다.

현재의 시스템은 단일 RR의 RDATA 섹션상의 저장되는 다수의 값보다는 타입의 다수값을 대표하는 다수의 RR을 사용한다. 이것은 대다수 응용에 대해 효율적이지 않지만, RR을 더 짧게 유지하게 된다. 다수의 RR의 소비는 다이나믹 업데이트 방법상의 어떠한 실험적인 것과 연계되있다.

현재의 시스템은 일관성을 확실하게 하기 위한 데이터베이스상의 데이터의 중복을 최소화하는데 시도된다. 이처럼, 메일 교환에 대한 호스트 주소를 찾기 위해서, 당신은 호스트 네임에 메일 도메인 네임을 매핑하고, 호스트에 대한 직접적인 매핑대신에 주소에 호스트 네임을 매핑한다. 이러한 접근은 불일치성에 대한 기회를 피할 수 있기에 더 나은 방법이다.

새로운 데이터 타입의 정의에 있어서, 다수의 RR 타입은 개체 혹은 이러한 정보가 RR의 몸체에서 전달되는 대신에 동등한 바인딩에 대한 다른 포멧을 표현하는 것과 단일한 타입의 사용 사이에서 순서를 만들어서는 안된다. 이런 정책은 다수의 타입을 캐싱하는 문제를 피하고, 다수 타입에 매칭하는 QTYPE을 정의한다.

예를 들면, 메일 교환 바인딩의 원래 포멧은 두가지 RR 타입을 사용하는데 하나는 “더 근접한” 교환(MD)을 대표하고 하나는 “덜 근접한” 교환(MD)을 나타낸다. 캐시상의 하나의 RR 타입의 존재는 MF, MD의 QTYPE 혹은 MAILA(둘다 매치하는)를 사용하는 캐시 정보를 필요로 하는 질의 때문에, 다른 것에 대한 어떠한 정보도 전달하지 않는다는 것이다.

재 디자인된 서비스는 서로 다른 RR의 순서를 매길 수 있는 RDATA 섹션상의 “우선” 값으로 단일 타입(MX)을 사용한다.

그렇지만, 만약 어떤 MX RR이 캐시에 있다면, 모든 것은 거기에 있게된다.

3. 메시지

3.1. 포멧

도메인 프로토콜내의 모든 통신은 메시지로 불리는 단일 포apt에 의해 전송된다. 메시지의 최상위 포멧은 아래에 보는 5개의 섹션(어떤 것은 어떤 경우에는 비어있다) 으로 구분된다.

```
+---------------------+
|        Header       |
+---------------------+
|       Question      | the question for the name server
+---------------------+
|        Answer       | RRs answering the question
+---------------------+
|      Authority      | RRs pointing toward an authority
+---------------------+
|      Additional     | RRs holding additional information
+---------------------+
```

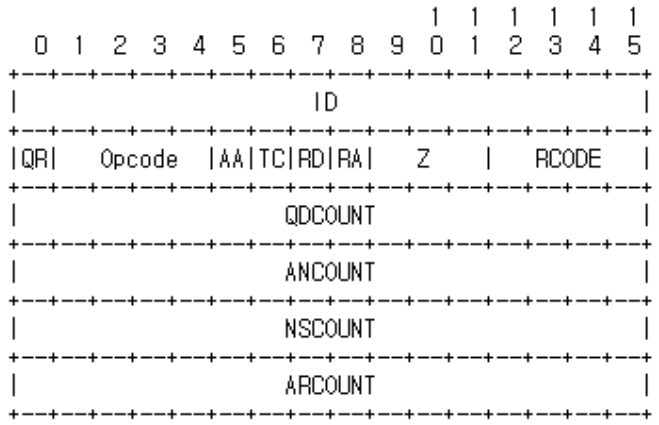
헤더 섹션은 항상 존재한다. 헤더는 남아있는 섹션 중 어떤 것들이 존재하는지를 나타내는 필드를 갖고 있다. 그리고 메시지가 질의 혹은 응답인지, 표준 질의인지 다른 Opcode인지를 나타낸다.

헤더 다음의 섹션의 이름은 표준 질의상의 용도에 따른다. 질문 섹션은 네임서버에 질문하는 것을 표현하는 필드를 포함한다. 이러한 필드는 질의 타입(QTYPE), 질의 클래스(QCLASS)와 질의 도메인 네임(QNAME)이다.

마지막 세 개 섹션은 동일한 포멧을 갖는다: 연결된 리소스 레코드(RR)의 목록으로서 비어있을 수도 있다. 응답 섹션은 질문에 대답하는 RR을 포함 한다; 권한 섹션은 권한을 갖는 네임서버에 대한 포인터를 갖는 RR을 포함 한다; 부가적인 레코드 섹션은 질의에 관련된 RR을 포함하지만, 질문에 대한 엄격한 응답은 아니다.

3.1.1. 헤더 섹션 포멧

헤더는 다음의 필드를 포함 한다:



여기서:

ID 임의의 질의를 발생하는 프로그램에 의해 할당되는 16 비트 식별자이다. 이러한 식별자는 대응하는 응답에 복사되고 질의 자에 의해 질의에 대해 매칭 되는 응답인지를 확인하는데 사용될 수 있다.

QR 메시지가 질의(0) 혹은 응답(1) 인지를 나타내는 1비트 필드이다.

OPCODE 이러한 메시지상의 질의 종류를 나타내는 4비트 필드이다. 이 값은 질의 발생한 측에 의해 설정되고 응답에 복사된다. 값은 다음과 같다:

0	표준 질의 (QUERY)
1	역 질의 (IQUERY)
2	서버 상태 요청(STATUS)
3-15	향후 사용을 위해 예약

AA Authoritative Answer - 이 비트는 응답 상에서 유효하고, 응답하는 네임 서버가 질의 섹션상의 도메인 네임에 대해 권한이 있음을 나타낸다. 별칭에 의해서 응답 섹션의 항목은 다수의 소유자 네임을 가질 수 있다. AA 비트는 질의 네임에 매칭 되는 네임과 일치하거나, 응답 섹션상의 첫 번

째 소유자 네임과 일치한다.

TC　　Turncation - 전송 채널이 허용하는 크기보다 더 크다거나하는 이유로 인해 이 메시지가 잘려졌음을 나타낸다.

RD　　Recursion Desired - 이 비트는 질의 내에서 셋팅되며 응답으로 복사된다, 만일 RD가 셋팅되어 있으면, 네임 서버에게 재귀적 질의를 처리하도록 지시한다. 재귀적 질의 지원은 선택적이다.

RA　　Recursion Available - 이 비트는 응답 내에서 셋팅 되거나 클리어 된다. 그리고 그 네임 서버에서 재귀적 질의가 지원되는지를 나타낸다.

Z　　나중을 위해 예약되어져 있다. 모든 질의의 응답에서 0이다.

RCODE　　Response code - 4bit 필드로서 응답의 일부분이다.
값은 다음과 같다.

0	애러 없음
1	Format error - 네임 서버가 질의를 해석할 수 없었다.
2	Server failure - 네임 서버에 문제가 있어서 네임서버가 이 질의를 처리할 수 없었다.
3	Name Error - 권한을 갖고 있는 네임 서버로부터의 응답일 때만 의미가 있다. 질의에 언급된 도메인 네임이 존재하지 않는다는 것을 의미한다.
4	Not Implemented - 네임 서버가 요청한 종류의 질의를 지원하지 않는다.
5	Refused - 정책전인 이유로 인해 네임서버가 지정된 동작을 수행하는 것을 거부했다. 예를 들어, 네임 서버가 특정 요청자에게 정보를 제공하는 것을 거부하거나, 영역 전송 같은 특정한 동작을 수행하지 않게 하는 것이다.
6-15	나중를 위해 예약되어져 있다.

QDCOUNT　　질문 부분 내에 있는 항목들의 개수를 나타내는 부호 없는 16 비트 정수

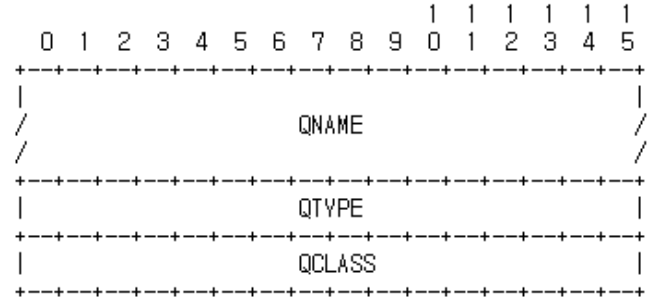
ANCOUNT　　질문 부분 내에 있는 리소스 레코드들의 개수를 나타내는 부호 없는 16bit 정수

NSCOUNT 권한 부분 내에 있는 네임 서버 리소스 레코드들의 개수를 나타내는 부호 없는 16bit 정수

ARCOUNT 기타 부분 내에 있는 리소스 레코드들의 개수를 나타내는 부호 없는 16bit 정수

3.1.2. 질문 부분 포멧

질문 부분은 질의 내의 '질문', 즉 무엇을 물어보는지 정의하는 인자들을 운반하는데 이용된다, 질문 부분은 QDCOUNT(보통 1) 항목들을 갖고 있고, 각 항목의 포멧은 다음과 같다.



범례:

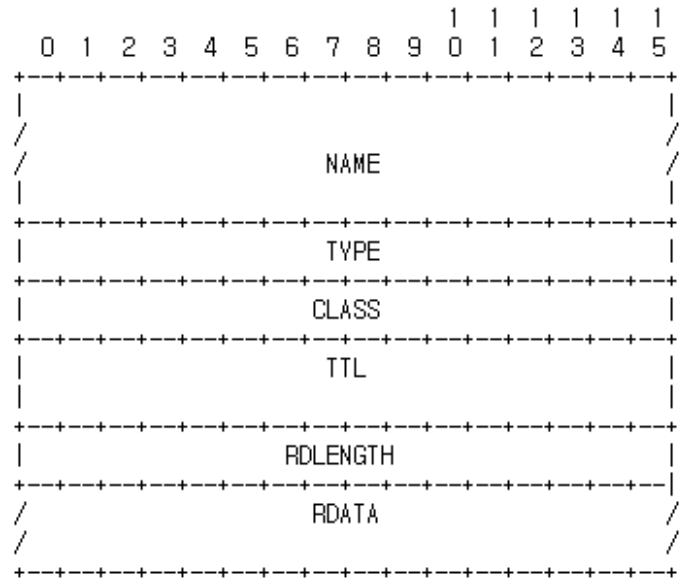
QNAME 레이블들의 나열로 표현되는 도메인 네임으로서, 각 레이블은 하나의 길이 옥텟과 그 숫자만큼의 옥텟들로 구성되어 있다. 도메인 네임은 루트의 null 레이블에 대한 옥텟인 길이가 0인 옥텟으로 끝난다. 이 필드는 홀수 개수의 옥텟들이라는 것에 유의하자. 패팅(padding)은 이용되지 않는다.

QTYPE 질의의 종류를 나타내는 2옥텟 코드이다. 이 필드의 값들은 TYPE 필드에서 이용되는 코드들을 모두 포함하고, 더하여 한 종류이상의 RR과 일치할 수 있는 몇 가지 일반적인 코드들을 포함한다.

QCLASS 질의 클래스를 나타내는 2옥텟 코드이다. 예를 들어, 인터넷에 대한 QCLASS 필드는 IN이다.

3.1.3. 리소스 레코드 포멧

응답 부분, 권한 부분, 그리고 기타 부분은 모두 형식이 같다. 리소스 레코드들의 개수는 가변적이고, 이 개수는 헤더의 count 필드에 명시되어 있다. 각 리소스 레코드는 다음과 같은 형식을 갖고 있다.



범례:

NAME 이 리소스 레코드에 관계된 도메인 네임이다.

TYPE RR 종류 코드들 중의 하나를 갖고 있는 2옥텟으로서, 이 필드는 RDATA 필드에 있는 데이터의 의미를 나타낸다.

CLASS RDATA 필드에 있는 데이터의 클래스를 나타내는 2옥텟이다.

TTL 리소스 레코드가 캐시에 남아있을 초 단위의 시간 간격을 나타내는 부호 없는 32bit 정수이다. 아 값이 0이면 캐시에 저장되지 않는다.

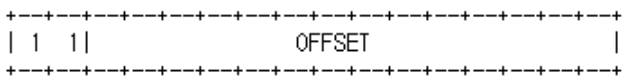
RDLENGTH RDATA 필드의 옥텟들의 길이를 나타내는 부호 없는 16bit 정수이다.

RDATA 리소스 레코드를 설명하는 가변 길이 문자열의 옥텟들이다. 이정보의 형식은 그 리소스 레코드의 TYPE 및 CLASS에 따라 달라진다. 예를 들어, 만일 TYPE이 A이고 CLASS가 IN이면, RDATA 필드는 4옥텟 ARPA 인터넷 주소이다.

3.1.4 메시지 압축

도메인 네임 시스템은 메시지들의 길이를 줄이기 위해 메시지 내에서 반복되는 도메인 네임들을 제거하는 압축 기법을 이용한다. 전체 도메인 네임 또는 도메인 네임의 끝부분의 레이블들을, 이전에 나왔던 동일한 이름의 위치를 가리키는 포인터로 대체하는 기법이다.

이 포인터는 2옥텟의 모습이다.



처음 두 비트는 모두 1로써, 레이블이 아니라 포인터라는 것을 나타낸다. 레이블은 처음 두 비트가 모두 0이고 레이블의 길이가 63옥텟 이하로 제한된다. 10및 01의 경우는 나중을 위해 남겨 놓았다. OFFSET 필드는 메시지의 시작 위치(즉, 도메인 헤더 내의 ID 필드의 첫 번째 옥텟)로부터의 오프셋을 나타낸다. 오프셋이 0인 경우는 ID 필드의 첫 번째 바이트를 나타낸다.

압축 기법은 메시지에 도메인네임 이하의 어떤 것이든 표현하는 것을 허락 하는 것은

- 0 옥텟으로 끝나는 순차적인 라벨

- 포인터

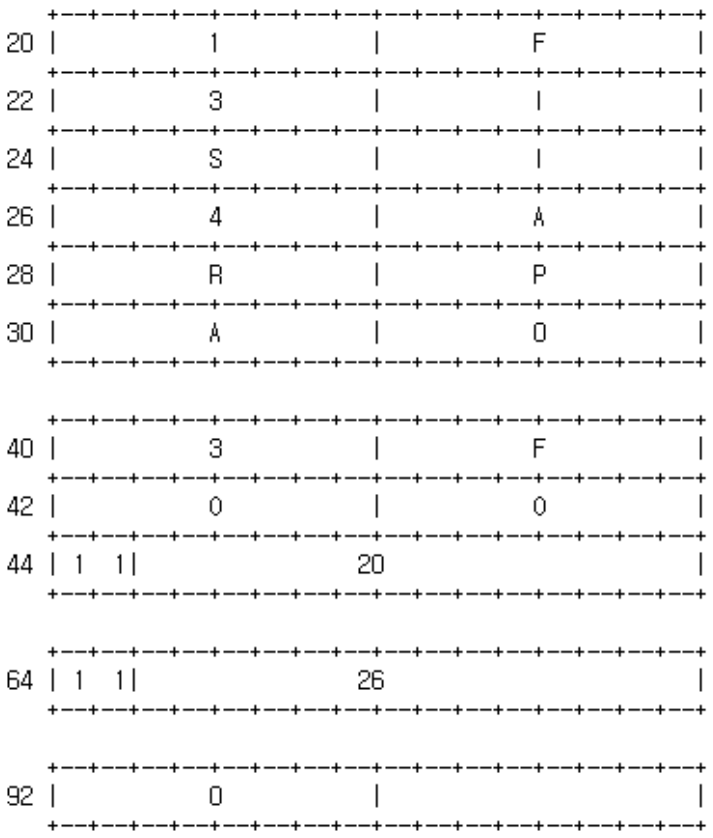
- 포인터로 끝나는 순차적인 라벨

특정한 클래스가 아닌 포멧에서 포인터는 단지 도메인 명으로만 사용될 수 있다. 그렇지 않다면 네임서버나 리졸버가 처리한 모든 RRs에 대한 포멧을 알도록 요구되어진다. 그러나 아직, 이러한 케이스는 없다, 미래의 RDATA 포멧에서는 발생할 것이다.

만약 도메인명이 (RR 섹션의 RDATA 같은) 메시지내의 길이 필드에 적용받는 부분에 포함되어져 있고 압축방법이 사용된다면, 압축된 이름의 길이는 길이 계산이 사용되는 것이 아니라 확장 이름의 길이로 된다.

데이터그램의 용량을 줄여주고 잘라냄에도 불구하고, 프로그램은 생성하는 메시지로 포인터를 사용하는 것에 대해서 자유롭다. 그러나 모든 프로그램은 도착한 메시지가 포인터를 포함하고 있는지 확인하는 것이 요구되어진다.

예를 들면, 데이터그램의 도메인 명이 F.ISI.ARPA, F.ISI.ARPA, ARPA 그리고 root를 사용을 필요로 한고 가정하자. 메시지의 다른 필드는 무시하자, 이 도메인 네임은 다음과 같이 표현될 것이다.



도메인 명 F.SIS.ARPA이 오프셋(offset) 20에 있다. 도메인명 FOO.F.ISI.ARPA에 오프셋(offset)이 40에 있다. 이 정의는 전에 정의된 F.ISI.ARPA에 FOO 라벨을 연결하는 포인터로 사용한다. 도메인명 ARPA에 오프셋(offset)이 64로 정의되어 20에 이름 F.ISI.ARPA의 부분으로 되어있는 ARPA의 포인터로 사용한다.; 이 포인터는 ARPA가 스트링 20 레벨의 마지막에 있다고 믿는 것에 주의해라. 루트 도메인은 92에 0 single octet로 정의 되어있다. 루트 도메인 명은 라벨을 가지지 않는다.

3.2 전송

DNS는 메시지가 데이터 그램으로 또는 가상 회로에 의해 이동되어진 byte의 흐름 안에서 통신되어지는 것을 가정한다. 가상 회로가 어떠한 DNS 활동도 사용할 수 있으나, 데이터그램이 더 작은 오버헤드(overhead)와 좋은 성능(performance) 때문에 질의를 위해서 사용되어진다. Zone 리플래시 활동으로는 신뢰할 수 있는 통신이 필요하기 때문에 가상 회로를 사용하여야한다.

인터넷은 네임서버 접근을 위해 TCP 53번(10진수) 포트 사용을 지원하고 마찬가지로 데이터 그램의 접근을 위해 UDP 53번(10진수) 포트를 사용을 지원한다.

3.2.1 UDP 사용법

메시지가 UDP의 유저 서버 포트 53번(10진수)을 이용하여 보낸다.

UDP에 의해서 전송된 메시지는 (IP나 UDP헤더를 제외하고) 512byte로 제한된다. 긴 메시지는 잘려지고 헤더의 TC bit이 셋팅되어진다.

UDP는 zone transfer를 위해서 사용될 수 없다. 그러나 인터넷에서 정상적인 질의를 위해서 사용되도록 권장되어지는 방법이다. 질의는 UDP를 사용한 보내진다, 그리고 따라서 재발송 전략이 요구되어진다. 질의 또는 그것의 응답은 네트워크 또는 네임서버의 프로세싱에 의해서 다시 요구되어질 수 있다. 그래서 리졸버는 반송 순서에 의존하면 안된다.

UDP 재발송 지침은 인터넷의 성능과 클라이언트의 필요에 따라 변하겠지만 다음 사항을 권유한다.

- 클라이언트는 서버의 특정 주소에 질의 하고 반복하기 전에 다른 서버와 서버 주소를 시도해야한다.

- 재질의 간격은, 가능하면, 사전의 통계치에 근거해야한다. 너무 공적적인 재질의가 커다란 커뮤니티에서 늦은 응답을 줄 수 있다. 클라이언트가 그것이 기대한 서버에 얼마나 잘 연결되어져 있느냐에 의존한다. 최소 재질의 시간은 2-3초 정도여야 한다.

서버 선택과 재질의 정책에 대한 좀 더 많은 제안이 리졸버 섹션의 메모에 있다.

3.2.2 TCP 사용법

TCP 연결에 의해 보내어지는 메시지가 서버 포트 53번(10진수)를 사용한다. 메시지의 앞에 메시지의 길이를 알려주기 위한 2byte 길이의 필드가 있다. 2byte 길이의 필드는 자신을 포함하지 않는다. 이 필드는 메시지를 해석하기 전에 낮은 레벨(low-level)의 프로세싱으로 완전한 메시지를 구성하는 것을 허용한다.

몇 개의 접속 관리 정책이 권유되어진다.

- 서버는 TCP 데이터를 기다리는 동안에 다른 활동을 멈추어야 하는 것이 아니다.

- 서버는 다중 접속을 지원해야한다.

- 서버는 클라이언트가 연결 닫기를 시작한다는 것을 가정한다, 그리고 모든 미처리 클라이언트 요구가 완료할 때까지 접속을 끊지 않아야 한다.

- 만약 서버가 자원 반환을 요구하기 위해서 휴면 중 접속을 끊을 필요가 있다면, 접속이 2분간 일어나지 않을 때까지 기다려야한다. 특히, 서버는 한 개의 접속으로 SOA, AXFR 요구를 순차적으로 처리하도록 요구되어진다. 서버가 질의에 대한 응답이 불가능 할때, 정상적인 종료 대신 정상적이지 않은 종료 또는 리셋이 사용되어 진다.

4. 마스터 파일

마스터파일은 텍스트 형식의 RRs를 포함하는 텍스트 파일이다. 존의 네용이 RRs 리스트의 모습으로 표현할 수 있기 때문에, Master 파일은 zone 정의로 사용되어져야만 한다, 따라서, 이번 섹션에서는 먼저 마스터 파일의 RRs 포멧에 대해 그리고 마스터 파일이 있는 네임서버로 존을 만들기 위한 특별한 고려사항을 논의 한다.

4.1 포멧

파일 포멧은 항목의 연속이다. 항목은 기본적으로 행단위이다, 소괄호를 사용하면 라인를 따라 항목을 계속 사용할 수 있다. 텍스트 문자에 텍스트 안에 CRLF를 포함할 수 있다. 탭과 공백의 조합이 항목을 만들어내는 요소간의 단락이 된다. master 파일의 어느 행도 코멘트로 종료할 수 있다. 코멘트는";"(세미콜론)으로 시작한다.

이하의 항목이 정의되어있다;

<blank>[<comment>]

$ORIGIN <domain-name> [<comment>]

$INCLUDE <file-name> [<domain-name>] [<comment>]

<domain-name><rr> [<comment>]

<blank><rr> [<comment>]

코멘트의 유무에 관계없이, 파일 내에서 공백행이 허용된다.

2개의 제어장치 항목이 정의 된다 : $ORIGIN와 $INCLUDE. $ORIGIN의 후에는 도메인 이름이 나오고, 도메인 이름에 상대적인 도메인 명을 현재 origin을 설정한다. $INCLUDE 지정된 파일을 현재의 파일에 삽입한다, 그리고 옵션으로 삽입 파일내의 상대적인 도메인명의 origin을 지정해도 괜찮다. $INCLUDE 코멘트를 가질 수 있다. $INCLUDE 항목이 삽입 파일 중에서 상대적인 origin을 변경해도 , 결코 원 파일에 상대적인 도메인의 origin을 바꾸지 않아야 한다.

마지막 2개의 서식은 RRs을 나타낸다. 만약 RR 항목이 공백으로 시작된다면, RR는 마지막에 나온 소유자에게 소유된다고 생각되고. 만약 RR항목이 <도메인명>으로 시작된다면 , 소유자명은 리셋 된다.

<rr>내용이 다음의 서식의 하나를 사용한다.

[<TTL>] [<class>] <type> <RDATA>

[<class>] [<TTL>] <type> <RDATA>

RR는 선택 가능한 TTL과 class field로 시작된다. 클래스와 타입은 표준 명칭을 사용한다. TTL은 10진법의 정수이다. 생략 된 클래스와 TTL값은 마지막 명시적으로 진술되고 그 값이 디폴트이다. 타입과 클래스 명칭이 공통 요소를 가지지 않기 때문에, 각각은 유일하다.

<domain-name>은 마스터 파일에서 데이터의 대부분을 공유 한다. 도메인명의 라벨은 점으로 단락 지어진 캐릭터 라인으로 나타내어진다. 인용 규칙이 임의의 문자를 도메인명으로 설정할 수 있도록 한다. 점으로 끝나는 도메인명이 완전한 도메인이라 불려진다. 점에 끝나지 않는 도메인명이 상대적적인 도메인으로 불려진다. 실제의 도메인명은 ,$ORIGIN이나 $INCLUDE나 마스터 파일을 로드하는 루틴의 인수로 지정 origin과 상대적 부분의 결합이다. 상대적인 이름 origin이 이용 할 수 없을 때 에러가 발생한다.

<character-string>는 하나 혹은 두개의 방법으로 표현된다. : 내부의 스페이스가 없는 연속 문자 , 혹은 "으로 시작해서 "으로 끝나는 캐릭터 라인. 그리고 단락 지어진 캐릭터 라인의 내부는 "이외 문자도 설정할 수 있고 "자신은 ₩(backslash) 으로 인용할 수 있다.

이러한 파일이 텍스트 파일이기 때문에, 몇개의 특별한 코드화가 임의의 데이터에 로드 되는 것을 허락하기 위해서 필요하다. 특히 :

루트에 대해

@	단독의 것 @가 현재의 origin을 의미하기 위해서 사용된다.
₩X	10이 자리 수(0-9) 이외의 캐릭터로 , 특별한 의미를 적용시키지 않고서, 그 문자를 인용하기 위해서 사용된다. 예를 들면, "₩."이 라벨 내에 닷 문자를 두기 위해서 사용할 수 있다.
₩DDD	각 D가 숫자로 DDD의 10진수로 나타나는 8진수입니다. 결과로서 생기는 8진수은 텍스트라고 생각되고 그리고 특별한 의미가 체크되지 않는다.
( )	괄호가 행을 넘는 그룹 데이터에 사용됩니다. 실제 , 괄호 내에서 줄 끝이 인식되지 않는다.
;	세미콜론은 코멘트를 시작하기 위해서 사용 된다; 라인의 나머지 부분이 무시된다.

4.2. 존을 정의하기 위한 마스터 파일의 사용

마스터 파일이 존을 읽어 들이는데 사용될 때 , 마스터 파일에 에러가 있다면 , 조작은 중지되어야 한다. 중지하는 이유는 하나의 에러가 광범하게 영향을 미치는 일이 있기 때문이다.

예를 들면 , 위임을 정의하고 있는 RR에 문법적 에러가 있다고 가정하면; 서버는 모든 subzone에 대해서 authoritative name 에러를 응답해준다. (서브 존이 같은 서버에 있는 경우 제외).

파일 구문이 올바른 일을 보증하기 때문에, 다른 추가적인 정당성 체크를 해야 한다. :

1. 파일중의 모든 RR는 같은 클래스를 가져야 한다.

2. 존의 처음에는 1개의SOA RR이 있어야 한다.

3. 만약 위임이 있어 , glue 정보가 필요하면 , 그것은 존재해야 한다.

4. 존재하고 있지만 존의 정식적 노드가 아닌 정보는 glue 정보여야 하는 것으로 , origin이나 비슷한 에러가 아니다.

4.3. 마스터 파일의 예

아래의 내용은 ISI.EDU존을 정의해 ,ISI.EDU 출신 정보를 로드 하는데 사용되는 파일의 예이다:

```
@    IN  SOA     VENERA       Action₩.domains (
                                20      ; SERIAL
                                7200    ; REFRESH
                                600     ; RETRY
                                3600000; EXPIRE
                                60)     ; MINIMUM

          NS      A.ISI.EDU.
          NS      VENERA
          NS      VAXA
          MX      10      VENERA
          MX      20      VAXA

A         A       26.3.0.103

VENERA    A       10.1.0.52
          A       128.9.0.32

VAXA      A       10.2.0.27
          A       128.9.0.33


$INCLUDE <SUBSYS>ISI-MAILBOXES.TXT
```

파일<SUBSYS>ISI-MAILBOXES.TXT은 다음과 같다 :

```
MOE         MB        A.ISI.EDU.
LARRY       MB        A.ISI.EDU.
CURLEY      MB        A.ISI.EDU.
STOOGES     MG        MOE
            MG        LARRY
            MG        CURLEY
```

책임자의 메일 박스 「Action.domains@E.ISI.EDU」를 지정하기 위해서 SOA RR에서 ₩표시를 사용한 것에 주의해라.

5. 네임서버 정보

5.1. 구조

네임서버의 최적인 구조는 호스트 operating system에 그리고 리졸버 동작을 포함하든지 포함하지 않던지, recursive 서비스를 지원하든지 지원하지 않든지, 또는 그 리졸버에게 데이터베이스를 공유되는지에 의존한다. 이 장은 리졸버와 데이터베이스를 공유하는 네임서버에 대한 충족의 고려를 논의 한다 , 그러나 이러한 것에 대부분의 관계는 현재의 네임 서버이다.

5.1.1. 제어

네임서버는 , 호스트 OS상 또는 하나의 네임서버 프로그램 안에서 멀티플랙싱이 되어 네임서버는 다중 처리를 지원해야만 한다. 네임서버가 , refresh 혹은 질의 동작에 대한 TCP 데이터 때문에, UDP 질의 서비스를 멈추어서는 안된다. 비슷하게, 네임서버가 request에 대한 병렬적인 처리 없이 recursive 서비스를 제공하거나 1개의 클라이언트로부터 연속하는 문의나, 또는 같은 클라이언트로부터의 같은 문의하고를 중복 한 문의로 간주해야 하는 것은 아니다. 네임서버가 , 마스터 파일로부터 존을 reload하는 동안이나 , 데이타베이스에 존을 refresh 할때, request를 늦추어야 하는 것은 아니다.

5.1.2. 데이터베이스

네임서버는 어떤 내부 데이터 구조를 선택해도 괜찮지만, 제안된 구조는 3개의 주요한 부분으로 구성되어있다.

- 이 서버가 취급하는 존을 리스트업 하는 “카탈로그”데이터 구조와 존 데이터 구조인 “포인터”. 이 구조의 주된 목적은, 만약 표준 질의가 왔을 때에 , 가장 가까운 존을 찾아내는 것이다.

- 네임서버가 가지는 존 각각 병렬적인 데이터 구조를 가져야한다.

- 캐시 데이터의 데이터 구조. (클래스별로 다른 캐시를 가져야한다.).

대문자 소문자의 문제를 해결하는 하나의 방법은 노드의 라벨을 2개의 부분에서 등록하는 일이다: 아스키 문자의 대문자 소문자가 어딘가에 되어 있는 표준화 한 대문자 소문자의 라벨과 각 문자의 대문자 소문자가 차이가 나는 비트열을. 분기 하고 있는 요소가 있는 반응을 일으키는 최소의 물리량을 넘을 때까지, 노드의 단순한 링크 리스트로 대처해, 반응을 일으키는 최소의 물리량을 넘으면 해시 구조로 이행할 수 있다. 어떤 조합에서도 , 기억 키로 사용하는 해시 구조에서의, 해시 기능과 순서가 DNS의 약속을 보관 유지하는 것을 보증하지 않으면 안된다.

5.1.3 시간

RRs에 대한 TTL 데이터와 refreshing activity에 대한 시간 데이터는 초단위의 32bit 타이머에 의존한다. 데이터가 일정 TTL에 의해 Zone에 있을 동안, 데이터베이스에서 refresh 타이머와 케시 데이터의 TTL은 개념적으로 “카운트 다운”된다.

추천되는 실행 전략은 시간을 두 가지 방법으로 저장하는 것이다: 상대적 증가와 절대적 시간. 이것을 하는 하나의 방법은 하나의 타입을 위해 32bit positive number를 사용하는 고 다른 것을 을 위해서는 negative number를 사용하는 것이다. Zone에 있는 RR들은 상대적인 시간을 가진다. refresh timer와 케시 데이터들은 절대적은 시간을 사용한다. 절대적인 number는 알려진 origin과 관계되어지고 질의에 대한 응답이 있을 때 상대적인 값으로 변환된다. 절대적은 TTL은 상대적으로 변화된 후에는 음수이고 그러면 데이터는 파괴괴고 무시되어져야 한다.

5.2. 표준 질의 처리

표준 질의에 대한 주요 알고리즘은 [RFC-1034]에 있다.

QCLASS=* 혹은 Multiple classes와 일치하는 QCLASS를 처리할 때, 서버가 모든 클래스에 대한 응답을 보증할 수 없는 한 응답은 정식적으로 되지 않는다.

응답이 구성될 때, additional section에 들어가는 RR은, 그러나 answer 또는 authority section의 RR들과 중복되지 않는, additional section에서 빠진다.

응답이 잘려서 두 개로 나누어 져야 할 만큼 길 때는, 잘림은 응답의 마지막으로부터 시작해야만 하고 데이터 그램에서 전방 방향에 실시해야 한다. 그리고 authority section의 데이터가 있다면 , answer section은 유일한 것으로 보증된다.

SOA의 최소 값는 존으로부터 배포되는 데이터의 생존 시간의 하한을 설정하기 위해서 사용되어야 한다. 이 하한은 , 데이터가 응답 카피될 때 , 되어야 하는 것이다. 이것은 미래의 dynamic update 프로토콜로 불명료한 것 없이 SOA MINIMUM 필드를 고칠 수 있도록 허락할 것이다.

5.3. 존 갱신과 reload 처리

서버의 노력에도 불구하고 , 마스터 파일의 구문 잘못이나 기타 등등의 이유로 zone data를 load할 수 없거나 또는 기간이 만료된 설정에서 zone을 refresh할 수 없을 경우가 있다. 이 경우, 네임서버는 그 존을 소유하고 있지 않는 것처럼, 질의에 응답해야 한다.

만약 마스터가 AXFR로 Zone을 보냈다면, 그리고 전송 중에 새로운 버전이 만들어진다면, 마스터는 가능하면 이전 버전을 계속 보내야 한다. 어떤 경우라도, 일부분은 새로운 버전으로 일부분은 이전 버전으로 보내지 말아야 한다. 만약 Zone 전송을 완료하는 것이 가능하지 않다면, 마스터는 존 전송을 하고 있는 것에 대한 접속을 RESET해야 한다.

5.4. 역 질의(옵션)

역 질의는 DNS의 옵션 부분입니다. 네임서버가 역 질의 형식의 어떠한 형식을 지원하도록 요구 되지 않는다. 만약 네임서버가 지원하고 있지 않는 역 질의를 받는다면, 헤더에 "Not Implemented" 에러를 설정해 에러 메시지를 돌려준다. 역질의 지원은 옵션이지만, 모든 네임서버는 적어도 에러 메시지를 돌려줄 수 있어야 한다.

5.4.1. 역 질의와 응답의 내용

역 질의 역 맵핑은 표준 질의 동작 형식에 맞추어 실행된다. 표준 질의가 도메인 명을 RR 레코드로 변환하는데 대해, 역 질의가 RR 레코드를 도메인 명으로 변환한다. 예를 들면, 표준 질의가 도메인명을 호스트 주소로 변환 한다 대응하는 역 질의는 호스트 주소를 도메인 명으로 변환한다.

역 질의는 question section은 비어있는 체로 메시지의 answer section에 하나의 RR 레코드 폼으로 결과를 받는다.
질의 RR 레코드의 소유자명과 생존 시간은 의미가 없다. 응답은 네임서버가 지는 모든 질의 RR 레코드를 소유하고 있는 이름을 질문부에 설정 한 것이다.
네임서버 도메인명 공간의 모든 것을 알고 있지 않기 때문에, 회답은 결코 완전한 것이라고 생각되지 않는다. 이 때문에 역 질의는 주로 데이터베이스 관리와 디버그 활동에 도움이 된다. 역 질의는 호스트 주소를 호스트 명으로 변환하는 좋은 방법이 아니다; 그 대신 IN-ADDR.ARPA도메인을 사용해야 한다.

가능한 경우, 네임서버가 역 질의에 대해 대문자 소문자의 차이를 무시하는 비교를 제 제공해야 합니다. 그래서"Venera.isi.edu"에 MX RR 레코드를 요구하는 역 질의가 "VENERA.ISI.EDU"의 역 질의와 같은 응답을 가져야 한다 "IBM-PC UNIX"는 "IBM-pc unix"의 역 질의와 같은 결과를 제공해야 한다. 그러나, 네임서버가 캐릭터 라인을 포함한 RR 레코드를 소유할지도 모르지만, 네임서버는 데이터가 문자인지 어떤지를 모르기 때문에 보증은 하기는 힘들다.

네임서버가 역 질의를 처리할 때 , 다음 사항을 돌려줄 수도 있다:

1. question section안에 QNAME에 만족하는 0개, 1개 혹은 다중 도메인 명.

2. 네임서버가 지정된 resource type에 대한 RR 메핑을 지원하지 않는 것을 나타내는 에러 코드.

역 질의에 대한 응답이 1개 이상이 QNAME을 포함할 때, 역 질의를 정의하는 응답부분에 RR 레코드의 소유자명과 생존 시간은 처음 QNAME에서 찾은 RR과 매치시켜 수정된다.

역 질의로 돌려주어진 RR 레코드가 표준 질의응답에 사용되고 있는 캐시 메커니즘으로 캐시 할 수 없다. 이 이유는 이름이 같은 타입의 다수의 RR 레코드 가질지도 모르지만, 한 개 만이 나타나기 때문이다. 예를 들면 , 멀티플 홈의 싱글 주소를 위한 역 질의는 호스트에게 주소가 1개만 있는 것 같은 인상을 줄지도 모른다.

5.4.2. 역 질의와 응답의 예

인터넷 주소10.1.0.52에 대응하는 도메인명을 돌려주는 역 질의의 전체적인 구조는 다음과 같다 :

```
              +------------------------------------------+
  Header      |          OPCODE=IQUERY, ID=997           |
              +------------------------------------------+
 Question     |                 <empty>                  |
              +------------------------------------------+
  Answer      |        <anyname> A IN 10.1.0.52          |
              +------------------------------------------+
 Authority    |                 <empty>                  |
              +------------------------------------------+
Additional    |                 <empty>                  |
              +------------------------------------------+
```

이 질의는 어떤 질문의 대답이 인터넷 스타일 주소10.1.0.52일까를 묻는다. 소유자명이 알려지지 않았기 때문에, 임의의 도메인명이 additional를 이용 할 수 있다. Root를 의미하는 하나의 팔진수 제로 값은 메시지 크기를 최소로 하기 때문에 일반적으로 이용된다. RR 레코드의 TTL 시간은 의미가 없다. 질의에 대한 응답은 아래에 있다:

```
              +------------------------------------------+
  Header      |          OPCODE=RESPONSE, ID=997         |
              +------------------------------------------+
 Question     |QTYPE=A, QCLASS=IN, QNAME=VENERA.ISI.EDU  |
              +------------------------------------------+
  Answer      |  VENERA.ISI.EDU  A IN 10.1.0.52          |
              +------------------------------------------+
 Authority    |                 <empty>                  |
              +------------------------------------------+
Additional    |                 <empty>                  |
              +------------------------------------------+
```

역 질의에 대한 응답으로 QTYPE은 역 질의의 answer section의 TYPE 필드와 같음에 주의해라. 역 질의에 대한 응답이 역 질의가 유일한 것이 아닐 때 다수의 질문을 포함할 지도 모른다. 만약 응답에서 question section이 비어있다면, answer section의 RR는 처음 QNAME에서의 RR의 정확한 복사에 대응하는 것 같게 수정된다.

5.4.3. 역 질의 처리

역 질의를 지원하는 네임서버는 데이터베이스의 철저한 탐색에 의해 이 같은 처리를 지원할 수 있다, 그러나 데이터베이스의 크기가 증가하면 실용적이지 않게 된다. 이를 대처하는 방법은 탐색의 키에 따라 데이타베이스를 뒤집는 것이다.

다수의 존과 대량의 데이터를 지원하는 네임서버에 대해서, 추천된 접근 방법은 각 존마다 역 검색이다. 특정의 존이 refresh로 바뀌었을 때 , 그 존의 inversions의 제 질의가 필요하다.

이 inversion 종류의 전송 지원이 도메인 시스템의 미래의 버전에 포함할 수 있을까에 대해서 알려져 있지 않고 이 문서에는 포함되지 않았다.

5.5. 완성 질의와 응답

RFC-882과 RFC-883에 기술된 임의의 완성 서비스는 삭제되었다. 디자인을 변경한 서비스가 장래 이용 가능하게 될 수도 있다.

6. 리졸버의 실행

리졸버 알고리즘의 개념은 [RFC-1034]에서 논의되어진다. 이 장은 네임서버 실행에서 제안된 데이타베이스 구조를 제안하고 있는 실행을 상세히 논의한다.

6.1. 질의로 사용자 요구사항의 전환

리졸버의 처음 순서는, 로컬 OS에 적절한 포멧으로 기술된 클라이언트 요구를, 지정한 이름에 일치하는 QTYPE과 QCLASS RR를 검색하는 것이다. 캐시 데이터의 이용은 단순하기 때문에, 가능한 경우 , QTYPE과 QCLASS는 하나의 종별과 하나의 클래스에 대응해야 하는 것이다. 이 같은 이유는 캐시 중에 있는 데이터 안의 하나의 타입에 존재가 다른 종류의 데이터 안에 존재 유무의 확인에 사용할 수 없다고 하는 것이다. 그러므로 확인하는 유일한 방법은 authoritative으로 조사하는 것이다. 만약QCLASS=*이라면 authoritative 대답은 이용 가능하지 않다.

리졸버가 효율적으로 기능을 실행한다면, 다수의 요구를 동시에 송신할 수 있기 때문에, 각각의 미결정 요구가 일반적으로 존재하는 상태의 정보로 나타내어진다. 상태 정보가 일반적으로 다음에 나타나 있다:

- 요구를 시작한 시간을 나타내는 타임 스탬프. 타임 스탬프는 데이타베이스 중의 RR이 사용될 수 있는지 결정하는데 사용된다, 이 타임 스탬프는 전의 존과 캐시의 RR의 등록에 의해 결정된 절대 시간 포멧을 사용한다. RR TTL 시간은 상대적인 시간을 가르킨다, RR은 존의 일부이므로, 정확한 최신정보인 것에 주의해야 한다. RR이 절대 시간을 가질 때, 그것은 캐시의 일부이다, 그리고 RR의 TTL은 요구한 타임 스탬프를 기준으로 해 판단된다. 제로의 TTL에 RR은 일반적인 방법으로 캐시에 넣을 수가 있지만, 시스템 부하나 재 발송 타임아웃 등으로 몇 초간 처리가 늦을까도 모르는 지금 실행중인 질의로 아직 RR를 사용하므로 , 현재 시각은 아니고 타임 스탬프를 사용하는 것에 주의해라.

- 요구를 실시하는 량을 제한하는 어떤 종류의 파라미터

- LIST 데이터 구조는[RFC-1034]에서 논의 된다.

  이 구조는 , 만약 다른 네임서버의 응답을 기다려야 한다면, 요구 상태를 기록을 추적해야한다.

6.2. 질의 보내기

[RFC-1034]에서 기술되었듯이, 리졸버의 기본적인 일은 클라이언트의 요구에 응답하는 것이다 그리고 질의를 생성해, 정보를 공급하는 네임서버를 향해 질의하는 일이다.
리졸버는 일반적으로 네임서버 RR의 형식에서 어느 서버에 요청 할 것인가 매우 강력한 힌트를 가지고 있다, 그리고 CNAME에 대한 응답 질의를 수정해야만 한다 또한 리졸버에 바람직한 정보에 의해 가까운 네임서버를 나타내는 위임 응답에 응해 리졸버가 묻는 네임서버의 집합을 수정해야 할지에 대해서도 알려져 있지 않다. 클라이언트에 의해 구할 수 있던 정보 외에, 리졸버는 연결하고 자하는 네임서버의 주소를 결정하기 위해 요구해야 할 수도 있다.

어떤 경우라도, 이 메모로 사용된 모델의 리졸버는 클라이언트로부터 요구된 요구와 내부에서 생성된 요구의, 다중의 요구를 처리해야 된다고 가정한다. 각 요구는 약간의 상태 정보로 나타내진다. 그리고 바람직한 행동은 리졸버는 대답할 가능성이 가장 높고 시간을 최소로 하는 네임서버로 질의를 보낸다. Key 알고리즘은 조회하는 다음의 네임서버 주소를 선택하기 위해서 요구 상태 정보를 사용해, 응답이 도착하지 않았던 다음의 행동을 일으키는 timeout을 계산한다. 다음의 행동은 일반적으로 다른 서버로 전송한다, 그러나 클라이언트에 일시적 에러를 전송할 수도 있다.

리졸버는 항상 조회해야 할 서버명의 SLIST로부터 시작한다. 이 리스트는 리졸버가 알고 있는 가장 가까운 곳의 선행되는 존에 대응하는 모든 NS RR이다. 시작 할 때 문제를 피하기 위해, 적절한 현재의 네임서버 자원 레코드를 가지지 않으면 , 리졸버는 그것을 묻는 디폴트 서버 군를 가져야 합니다. 리졸버는 서버 리스트에 네임서버에 대한 알려진 모든 주소를 SLIST로 더한다. 그리고 리졸버가 네임서버의 주소가 아닌 이름만을 가지고 있을 때, 서버의 주소를 획득하는 요구를 시작해도 괜찮다.

SLIST의 초기화를 완료하기 위해서 , 리졸버는 SLIST의 각각의 주소에 대한 이력 정보를 부가한다. 이것은 일반적으로 주소의 응답 시간과 주소 응답률의 가중치 평균에 의한 순서로 구성된다. (그 주소가 어느 정도 요구에 응답했는지)의 순서 붙이게 된다. 어느 서버의 응답 시간이나 응답률은 주소마다 차이가 있으므로, 응답 시간이나 응답률은 네임서버 마다는 아니고 주소마다 보관되어야 한다. 그리고 이 정보가 리졸버 주소와 서버 주소의 대마다 특별한 일로 주의해야 한다, 다수의 주소를 가지는 리졸버가 각 주소 마다의 이력의 보관 유지를 바랄지도 모른다. 이 스텝에서 이력을 가지지 않는 주소도 취급하지 않으면 안된다 이 경우5-10초의 왕복 시간이 최악의 경우로 , 로컬 네트워크에서는 보다 낮게 나타난다.

위임을 할 때는 언제라도 , 리졸버 알고리즘이 SLIST를 다시 초기화하는 것에 주의해라.

정보는 이용 가능한 네임서버 주소의 부분적인 순위를 확립한다. 주소가 선택되어진 각각의 시간에 모든 다른 주소가 시험 받을 때까지 다시 같은 주소를 선택하지 않도록 상태를 변화시킨다. 각 전송의 타임아웃은 분산을 고려해 평균 예측치 보다 50-100%크게 있어야 한다.

우수한 포인트 :

- 리졸버는 SLIST로 지정된 네임서버의 모두가 이용 가능하지 않고 , 리스트중의 서버가 일반적으로 그 서버 자신의 주소에 문의하지 않는 상태를 만난다. 이 상태는 일반적으로 , glue address RR이 NS RR이 위임을 만드는 시간보다 좀 더 작은 TTL일 때나, 또는 리졸버가 NS 검색의 결과를 캐시 하는 경우에 발생한다. 리졸버는 이 상태를 검출 해야만 한다. 그리고 다음 부모 존에서 검색을 다시 시작 하거나 또는 루트로부터 검색을 다시 해야 한다.

- 만일 리졸버가 네임서버로부터 서버 에러나 다른 기이한 반응을 받았다면, 그 네임서버를 SLIST로부터 제외해야 한다, 그리고 다음의 후보 서버 주소에 즉각 송신하는 것이 좋다.

6.3. 처리 응답

받아들인 응답 데이터그램을 처리하는 최초의 순서는 응답을 분석하는 것이다. 이순서가 아래에 포함되어져야 한다;

- 헤더가 합리적안가를 체크 한다. 응답이 기대되는데 질의 데이터 그램을 버려라.

- 메세지의 각부를 분석해라, 모든 RR가 정확 포멧인 것을 보증해라.

- 옵션 순서로서, TTL이 현저히 긴 RR이 없는가 확인한다. 만약 RR의 TTL이 현저히 길면 , 예를 들면 1주간이 상이라면, 응답 전체를 버리거나, 또는 모든 응답의 TTL을 1주간 제한해야 한다.

다음의 순서는 현재의 리졸버 요구에 대한 회답과 일치하는 것이다. 추천된 전략은 도메인 헤더로 ID 필드를 사용해 일치하는 것을 사전 검사해, 다음에 질의가 바람직한 정보에 대응하는 것을 확인하는 일이다. 이것은 송신 알고리즘이 도메인 ID의 몇 비트인지를 메세지의 분류에 사용할 수 있는 것을 요구한다. 이 순서에는 몇 개의 미묘한 포인트가 있다:

- 어떤 네임서버는 질의하고 응답하는데 사용한 것과 다른 주소를 응답으로 보낸다. 즉, 질의와 같은 주소로 응답을 기다리는 리졸버는 응답받을 수 없다. 이 네임서버 버그는 UNIX 시스템에서 전형적으로 나타난다.

- 만약 리졸버가 네임서버에 있는 request를 재 발송한다면, 어떤 응답도 사용할 수 있도록 해야 한다. 그러나, 응답을 한 네임서버에 접속하는 왕복 시간을 조사한다면, 어느 질의과 어느 응답이 대응되는지에 대한 결정은 (그리고 모든 송신 메세지의 송신 시간을 보관 유지한다), 최초의 송신에 근거해 왕복 시간을 계산해야한다.

- 네임서버가 때때로 있는 네임서버 RR에 관한 존에 최신의 복사본을 가지지 않을 것이다. 리졸버는 다만 네임서버를 현재의 SLIST으로부터 제외하고 계속해야 한다.

6.4. 캐시의 사용

일반적으로 , 리졸버는 미래의 클라이언트 요구에 답할 때에 유용할지도 모르기 때문에, 응답으로 받아들이는 모든 데이터를 캐시하는 것을 기본으로 한다. 그러나 캐시해서는 안되는 몇 종류의 데이터가 있다 :

- 특별한 소유자명의 같은 타입에 대한 몇 개의 RR이 이용 가능할 때, 리졸버는 그것들 모두를 캐시를 하든지 , 모두 캐시하지 않든지를 결정해야 한다. 응답이 잘리어 졌을 때, 이것이 완전한 것인지 아닌지 리졸버는 알지 못한다. 부분적인 RR 집합을 캐시해야 하지는 않는다.

- 캐시된 데이터가 결코 정상적인 데이터에 대해서 우선적으로 사용되지 않는다, 만약 캐시로 이것이 생긴다면 , 데이터는 캐시 되지 않아야 한다.

- 역 질의 결과는 캐시되지 않아야 한다.

- 만약 데이터가 와일드 카드를 구성한다면 QNAME에 "*"라벨을 포함한 표준 쿼리의 결과이다. 이유는 캐시가 와일드 카드 자원 레코드의 어플리케이션을 한정하기 위해서 필요한 기존의 자원 레코드나 존 한계의 정보를 포함하지 않기 때문이다.

- 신뢰성의 의심스러운 회답의 RR 데이터. 리졸버가 바라지 않는 응답이나 요구된 것 이외의 RR 데이터를 받을 때 , 그것을 캐시하지 않고 버려져야 한다. 기본적인 의미는 모든 패킷의 안전 점검이 , 그것을 캐시하기 전에 실시해야 하기 때문이다.

유사한 경향으로, 리졸버가 있는 이름의 RR의 응답 얻고 그 RR를 캐시 하는 것을 바랄 때, 이미 캐시 되어진 기존의 RR이 없는가 캐시를 체크해야 한다. 상황에 의해, 응답 데이터나 캐시 데이터가 우선되어져야 한다, 그러나 이 2개는 결합되어야 하는 것이 아니다. 만약 응답의 데이터가 응답 부분에서 정상적인 데이터라면, 항상 우선된다.

7. 메일 지원

도메인 시스템은 도메인명 안에 메일 박스를 도메인으로 맵핑 하기 위한 표준과 메일 전달 정보를 얻기 위한 메일 박스 정보를 사용하는 2개의 방법을 정의한다. 첫번째 방법은 mail exchange binding으로 불리고 다른 방법은 mailbox binding이라고 불린다. Mailbox encoding 표준과 mail exchange binding는 DNS 공식 프로토콜의 일부이고, 인터넷으로 메일 전달을 위해 추천되는 방법이다. Mailbox binding은 실험적 기능으로 개발 중이며 변경 가능성이 있다.

메일 박스 코딩 표준은"<local-part>@<mail-domain>"형식의mailbox명을 가정한다. 이러한 이 같은 섹션의 구문은 인터넷에서 다양한 메일을 위해 충분히 다양하다, ARPA인터넷을 위해 취하여진 구문은[RFC-822]에 있다.

DNS는 하나의 라벨로서<local-part>를 코드화하고, 도메인 명으로서 <mail-domain>를 코드화한다. 메일 박스에 대응하고 있는 도메인명을 구성하기 위해서 ,<mail-domain>의 전에<local-part>가 붙는다. 그래서 메일박스 HOSTMASTER@SRI- NIC.ARPA는 도메인명HOSTMASTER.SRI-NIC.ARPA.으로 변환 된다. 만약<local-part>이 도트이거나 다른 특별한 문자를 포함한다면, 마스터 파일 내에서 backslash로 표현하는 것으로 도메인명을 정확하게 코드화 시키는 것을 보증하도록 요구되어져야 한다. 예를 들면 , 메일 박스 Action. domains@ISI.EDU는 Action₩.domains.ISI.EDU이라고 표현된다.

7.1. 메일 exchange binding

메일 exchange binding은 메일 박스의<mail-domain>부분이 메일의 발송지를 결정하는데 사용한다. <local-part>는 조사할 수 없다. [RFC-974]에 이 부분에 대해 자세히 설명되어져 있고, 메일 교환을 사용하기 전에 조사할 수 있어야 한다.

이 방법의 이점중의 하나는 메일 서비스에 사용하는 호스트와 메일의 행선지명을 분리시킬 때, 다른 레이어의 가중치를 통한 검색 기능이다. 그러나 추가적인 레이어는 <local-part>의 코딩으로 복잡한 "%"나 "!"의 필요성을 배제해야 한다.

<mail-domain>은 도메인 이름을 <mail-domain>에 대한 메일을 받아들이는 MX RR에 위치시키는 것으로 이용된다, <mail-domain>에 대한 관리자의 지정한 순서를 나타내는 우선순위로 호스트를 늘어놓는 것이다.

이 메모에서, <mail-domain>ISI.EDU가 예로 사용되어진다, 호스트VENERA.ISI.EDU와 VAXA.ISI.EDU가ISI.EDU에 의해서 메일을 교환된다. 만약 메일 송신자가 Mockapetris@ISI.EDU 메세지를 가지고 있으면, ISI.EDU에 대한 MX RR를 조사해 전달

경로를 결정한다. ISI.EDU의 MX RR는VENERA.ISI.EDU과VAXA.ISI.EDU의 이름을 나타내고, type A 질의는 호스트의 주소를 찾을 수 있다.

7.2. 메일박스 binding(실험용)

메일박스 binding에서, 메일 송신자는 도메인명을 조립하기 위해서 전부의 메일 행선지 사양을 사용한다. 메일 박스를 위해 코드화 된 도메인 명은 QTYPE=MAILB안의 QNAME 필드 쿼리를 사용한다.

몇개의 결과는 이 질의를 위해 가능하다 :

1. 질의는 메일박스가 도메인 명으로써 존재하지 않는 것을 나타내는 이름 에러를 돌려줄 수 있다.

   장기간 동안, 이것은 지정된 메일 박스가 존재하지 않는 것을 나타낸다. 그러나 mailbox binding 사용이 일반적으로 될 때까지, 이러한 에러 조건은 메일 주소의 글로벌인 부분에 나타내어지는 조직이 mailbox 할당을 지원하지 않는 다는 것을 나타낸다.

2. 질의는 Mail Rename(MR) RR를 돌려줄 수가 있다.

   메일 개명 RR는 그 RDATA 필드에 새로운 메일 박스 사양을 싣는다. 메일러는 낡은 메일 박스를 새롭게 바꾸고, 처리를 속행해야 한다.

3. 문의는 메일 박스 RR를 돌려줄 수가 있다.

   MB RR은 RDATA 필드 호스트의 도메인명을 옮긴다. 메일 송신자는, 적용 가능한 프로토콜로 , 예를 들면 SMTP로, 호스트에게 메세지를 보내야 한다.

4. 질의는 1개이상의 Mail Group(MG) RR를 돌려줄 수가 있다.

   이 조건은 메일 박스가 , 하나의 메일 박스가 아니고 , 실제는 메이링 리스트나 메일 그룹인 것을 의미한다. 각 메일 MG RR은 그룹의 멤버인 메일 박스를 식별하는 RDATA 필드를 가지고 있다. 메일 송신자는 각 멤버에게 메시지의 복사본을 보내야 한다.

5. 질의는 1개 이상의 MG RR로서 MB RR을 돌려줄 수 있다.

   이 상태는 메일 박스가 실제는 메일링 리스트인 것을 의미한다. 메일 송신자는 MB RR로 지정되는 호스트에게 메시지를 보낸다.

이러한 경우의 모두 응답에 메일 정보(MINFO) RR를 포함한다. 이 RR는 일반적으로 메일 그룹과 연결된다, 그러나 RR은 MB로도 맞다. MINFO RR는 2개의 메일 박스를 식별한다. 이 같은 식별자의 한 가지는 오리지널 메일 박스명의 책임자를 지정하는 것이다.

이 메일박스는 메일 그룹에 가입을 요구하는 것에 사용되어야 한다. MINFO RR의 두번째 메일박스 이름은 메일 장애에 대한 에러 메시지를 받아야 할 메일 박스를 지정한다. 이것은 특히 메일링리스트로 , 리스트에 메시지를 보내는 것 이외에 멤버 명에서의 에러를 사람들에게 보고되어져야만 할 때, 적절한 방법이다.

새로운 필드가 장래 이 자원 레코드에 더해질지도 모른다.

표준작성 공헌자

표준 번호 : TTAS.IF-RFC1035

이 표준의 제·개정 및 발간을 위해 아래와 같이 여러분들이 공헌하였습니다.

구분	성명	위원회 및 직위	연락처 (Tel, E-mail)	소속사
과제 제안	이승훈	IAR PG 위원	02-2186-4585 sehlee@nida.or.kr	한국인터넷진흥원
표준 초안 제출	이승훈	IAR PG 위원	02-2186-4585 sehlee@nida.or.kr	한국인터넷진흥원
표준 초안 검토	김 원	IAR PG 의장	02-286-4502 wkim@nida.or.kr	한국인터넷진흥원
	외	IAR PG 위원 15명		
표준안 심의	민경선	전송통신기술위원회 의장	042-870-8340 minks@kt.co.kr	KT
		외 위원 20명		
사무국 담당	김선	정보통신팀 팀장	031-724-0080 skim@tta.or.kr	TTA
	조은주	정보통신팀 과장	031-724-0082 jej@tta.or.kr	TTA

정보통신 단체표준

도메인 네임 - 구현 및 스펙
(Domain Names - Implementation and Specification)

발행인 : 김 원 식
발행처 : 한국정보통신기술협회
463-824, 경기도 성남시 분당구 서현동 267-2
Tel : 031-724-0114, Fax : 031-724-0019
발행일 : 2007. 11. 8