개요

DNS (Domain Name System)는 인터넷에서 도메인 이름을 IP 주소로 변환하는 분산형 데이터베이스 시스템입니다. 예를 들어, `www.example.com` 같은 사람이 읽을 수 있는 주소를 컴퓨터가 처리할 수 있는 숫자 주소인 IP로 변환합니다. DNS는 1983년 Paul Mockapetris에 의해 도입되었으며, 인터넷 사용의 편리성을 크게 향상시켰습니다.

주요 특징

1. **계층적 구조**: DNS는 전 세계에 분산된 수많은 서버로 구성되어 있으며, 이들은 계층적으로 관리됩니다. 루트 서버, TLD (Top-Level Domain) 서버, 권한 있는 네임 서버 등 여러 단계로 구성됩니다.
2. **도메인 이름**: 각 도메인에는 고유한 이름이 있으며, 이는 해당 도메인의 인터넷 주소를 나타냅니다.
3. **캐싱**: DNS 쿼리 응답은 DNS 서버에 의해 캐시되어 성능을 향상시키고 지연 시간을 줄입니다.
4. **분산 데이터베이스**: DNS 정보는 전 세계에 걸쳐 여러 서버에 분산되어 저장되어 있어, 하나의 서버에 장애가 발생해도 DNS 서비스가 계속 작동할 수 있습니다.

주의 사항

1. **DNS 스푸핑**:
   - 공격자가 DNS 응답을 조작하여 사용자를 악성 웹사이트로 유도할 수 있는 보안 위험입니다.
2. **DNS 증폭 공격**:
   - 소량의 쿼리로 대량의 응답을 유도하여 서버나 네트워크를 과부하시킬 수 있습니다.
3. **개인 정보 노출**:
   - DNS 쿼리는 사용자의 인터넷 사용 행위를 나타낼 수 있으므로, 암호화되지 않은 채로 전송될 경우 개인 정보가 노출될 수 있습니다.

DNS 메시지의 헤더 구조는 DNS 프로토콜의 중심적인 부분이며, 모든 DNS 쿼리와 응답 메시지에서 처음에 위치합니다. 헤더는 DNS 작업을 제어하는 데 사용되는 여러 필드로 구성되어 있습니다. 각 필드는 특정 기능을 수행하며, DNS 메시지의 성격과 처리 방법을 정의합니다.

DNS 레코드

DNS 레코드는 도메인 이름 시스템(DNS)에서 도메인 이름과 관련된 다양한 정보를 저장하는 데 사용됩니다. 이 레코드들은 특정 도메인에 대한 IP 주소, 메일 서버, 다른 도메인 이름으로의 리디렉션 정보 등을 포함할 수 있으며, DNS 쿼리에 대한 응답으로 사용됩니다.

A 레코드 (Address Record)

- **목적**: 도메인 이름을 IPv4 주소로 매핑합니다.
- **예시**: `example.com` → `93.184.216.34`

AAAA 레코드 (IPv6 Address Record)

- **목적**: 도메인 이름을 IPv6 주소로 매핑합니다.
- **예시**: `example.com` → `2606:2800:220:1:248:1893:25c8:1946`

MX 레코드 (Mail Exchange Record)

- **목적**: 도메인의 이메일을 처리할 메일 서버의 주소를 지정합니다.
- **예시**: `example.com`의 이메일을 처리하는 서버는 `mail.example.com`입니다.

CNAME 레코드 (Canonical Name Record)

- **목적**: 한 도메인 이름을 다른 도메인 이름으로 매핑합니다(별칭 제공).
- **예시**: `www.example.com` → `example.com`

NS 레코드 (Name Server Record)

- **목적**: 도메인에 대한 권한 있는 DNS 서버를 지정합니다.
- **예시**: `example.com`의 DNS 쿼리를 처리하는 서버는 `ns1.example.com`입니다.

DNS 헤더 구조

```
HEADER:
    - ID = 1234
    - QR = 1 (응답)
    - OPCODE = 0 (표준 질의)
    - AA = 1 (권한 있는 응답)
    - TC = 0 (비절단 메시지)
    - RD = 1 (재귀 요청)
    - RA = 1 (재귀 허용)
    - Z = 0 (예약)
    - RCODE = 0 (응답 코드: 없음)

QUESTION SECTION:
    - www.example.com. IN A
```

```
ANSWER SECTION:
    - www.example.com. 86400 IN A 93.184.216.34

AUTHORITY SECTION:
    - example.com. 172800 IN NS ns1.example.com.
    - example.com. 172800 IN NS ns2.example.com.

ADDITIONAL SECTION:
    - ns1.example.com. 86400 IN A 93.184.216.1
    - ns2.example.com. 86400 IN A 93.184.216.2
```

1. **Transaction ID (16 bits)**:
   - **용도**: DNS 쿼리와 해당 응답을 서로 연결하는 데 사용됩니다. 쿼리를 발생시킨 클라이언트가 응답을 올바르게 식별할 수 있게 해줍니다.
2. **Flags (16 bits)**:
   - 이 필드는 여러 작은 비트 필드로 나누어져 있으며, 각각 특정 기능을 수행합니다:
     - **QR (1 bit)**: 메시지가 쿼리인지(0), 응답인지(1)를 나타냅니다.
     - **Opcode (4 bits)**: 쿼리의 유형을 나타냅니다(예: 표준 쿼리는 0).
     - **AA (Authoritative Answer, 1 bit)**: 응답이 권한 있는 네임 서버에서 왔는지를 나타냅니다.
     - **TC (Truncated, 1 bit)**: 메시지가 길어서 절단되었음을 나타냅니다.
     - **RD (Recursion Desired, 1 bit)**: 쿼리에서 재귀를 원하는지를 나타냅니다.
     - **RA (Recursion Available, 1 bit)**: 서버가 재귀를 지원하는지 나타냅니다.
     - **Z (3 bits)**: 예약되어 사용되지 않습니다.
     - **RCODE (Response Code, 4 bits)**: 응답 코드로, 쿼리 처리 결과를 나타냅니다 (예: 오류 없음은 0).
3. **Questions (16 bits)**:
   - 쿼리에서 요구하는 질문의 수를 나타냅니다. 대부분의 쿼리에서는 이 값이 1입니다.
4. **Answer RRs (16 bits)**:
   - 응답 섹션에 있는 리소스 레코드의 수입니다.
5. **Authority RRs (16 bits)**:
   - 권한 있는 섹션에 있는 리소스 레코드의 수입니다.
6. **Additional RRs (16 bits)**:
   - 추가 정보 섹션에 있는 리소스 레코드의 수입니다.

DNS 계층적 구조

DNS의 계층 구조는 크게 네 가지 주요 수준으로 나눌 수 있습니다:

1. 루트 DNS 서버

- **위치**: 계층 구조의 최상위에 있습니다.
- **기능**: 루트 DNS 서버는 전체 DNS 시스템의 근간을 이룹니다. 전 세계에는 13개의 루트 서버가 있으며, 이들은 여러 위치에 복제되어 고장 없이 운영됩니다. 루트 서버는 최상위 도메인(Top-Level Domain, TLD) 서버의 주소를 알고 있으며, 도메인 이름 질의가 들어오면 적절한 TLD 서버로 요청을 전달합니다.

2. 최상위 도메인 (TLD) 서버

- **위치**: 루트 서버 바로 아래입니다.
- **기능**: TLD 서버는 도메인 이름의 확장자를 관리합니다. 예를 들어, `.com`, `.net`, `.org`, 국가 코드 TLDs 예: `.uk`, `.de` 등을 담당합니다. 각 TLD 서버는 해당 TLD에 등록된 모든 도메인 이름의 정보를 갖고 있으며, 특정 도메인에 대한 질의가 발생하면 해당 도메인의 권한 있는 서버 주소를 제공합니다.

3. 권한 있는 DNS 서버

- **위치**: TLD 서버 다음의 계층입니다.
- **기능**: 권한 있는 서버는 특정 도메인의 DNS 레코드를 직접 관리합니다. 이 서버들은 해당 도메인에 대한 모든 세부 정보를 갖고 있으며, 해당 도메인 이름에 대한 모든 질의(예: A 레코드, MX 레코드, 등)에 직접 응답합니다. 도메인의 소유자가 설정한 정보를 바탕으로 DNS 쿼리에 응답합니다.

4. 재귀적 DNS 서버

- **위치**: 사용자와 권한 있는 서버 사이에서 중개자 역할을 합니다.
- **기능**: 일반적으로 ISP(Internet Service Provider)에 의해 제공되며, 사용자의 DNS 쿼리를 받아 처리하는 서버입니다. 사용자가 웹사이트에 접속하려 할 때, 재귀적 DNS 서버가 사용자를 대신해 루트 서버로부터 시작하여 순차적으로 TLD 서버, 그리고 필요한 권한 있는 서버까지 요청을 전달하고 최종적으로 원하는 정보를 사용자에게 반환합니다. 이 과정에서 재귀적 서버는 질의에 대한 응답을 캐시하여, 동일한 질의에 대한 응답 속도를 빠르게 합니다.

DNS의 동작 과정

1. DNS 쿼리 발생

사용자가 웹 브라우저에 URL을 입력하면, 운영 체제는 먼저 호스트 파일을 확인합니다. 호스트 파일에 해당 도메인 이름에 대한 항목이 있는 경우, 해당 IP 주소를 사용하여 연결을 시도합니다.

호스트파일에 항목이 없다면, 운영체제는 사용자의 시스템이나 라우터에 설정된 로컬 DNS 서버(재귀적 DNS 서버)에 도메인 이름을 IP 주소로 변환해 달라는 요청을 보냅니다.

```
HEADER:
    - ID = 1234
    - QR = 0 (질의)
    - OPCODE = 0 (표준 질의)
    - AA = 0 (권한 없음)
    - TC = 0 (비절단 메시지)
    - RD = 1 (재귀 요청)
    - RA = 0 (재귀 허용 안 됨)
    - Z = 0 (예약)
    - RCODE = 0 (응답 코드: 없음)

QUESTION SECTION:
    - www.example.com. IN A

ANSWER SECTION:
    - (빈 섹션)

AUTHORITY SECTION:
    - (빈 섹션)

ADDITIONAL SECTION:
    - (빈 섹션)
```

2. 루트 서버 접근

로컬 DNS 서버는 루트 DNS 서버에 접근하여 도메인의 최상위 도메인 (예: `.com`, `.net`)에 대한 정보를 요청합니다.

3. TLD 서버 접근

루트 서버는 해당 TLD 서버의 주소를 로컬 DNS 서버에 알려줍니다. 로컬 서버는 TLD 서버에 접근하여 도메인의 권한 있는 네임 서버 주소를 요청합니다.

- **권한 있는 네임 서버(Authoritative Name Server):** 특정 도메인의 DNS 레코드에 대한 최종적이고 권한 있는 정보를 제공하는 DNS 서버

4. 권한 있는 네임 서버 접근

TLD 서버는 도메인의 권한 있는 네임 서버 주소를 로컬 DNS 서버에 제공합니다. 로컬 서버는 이 네임 서버에 접근하여 해당 도메인의 IP 주소를 요청합니다.

5. IP 주소 반환 및 캐싱

권한 있는 네임 서버는 도메인의 IP 주소를 로컬 DNS 서버에 반환합니다. 로컬 서버는 이 IP 주소를 캐싱하고, 사용자에게 IP 주소를 반환하여 웹 브라우저가 해당 IP 주소로 웹 서버에 연결할 수 있도록 합니다.

```
HEADER:
    - ID = 1234
    - QR = 1 (응답)
    - OPCODE = 0 (표준 질의)
    - AA = 1 (권한 있는 응답)
    - TC = 0 (비절단 메시지)
    - RD = 1 (재귀 요청)
    - RA = 1 (재귀 허용)
    - Z = 0 (예약)
    - RCODE = 0 (응답 코드: 없음)

QUESTION SECTION:
    - www.example.com. IN A

ANSWER SECTION:
    - www.example.com. 86400 IN A 93.184.216.34

AUTHORITY SECTION:
    - example.com. 172800 IN NS ns1.example.com.
    - example.com. 172800 IN NS ns2.example.com.

ADDITIONAL SECTION:
    - ns1.example.com. 86400 IN A 93.184.216.1
    - ns2.example.com. 86400 IN A 93.184.216.2
```

사용 사례

- **웹 브라우징**: 사용자가 웹사이트를 방문할 때 매번 사용하는 기본적인 DNS의 사용 사례입니다.

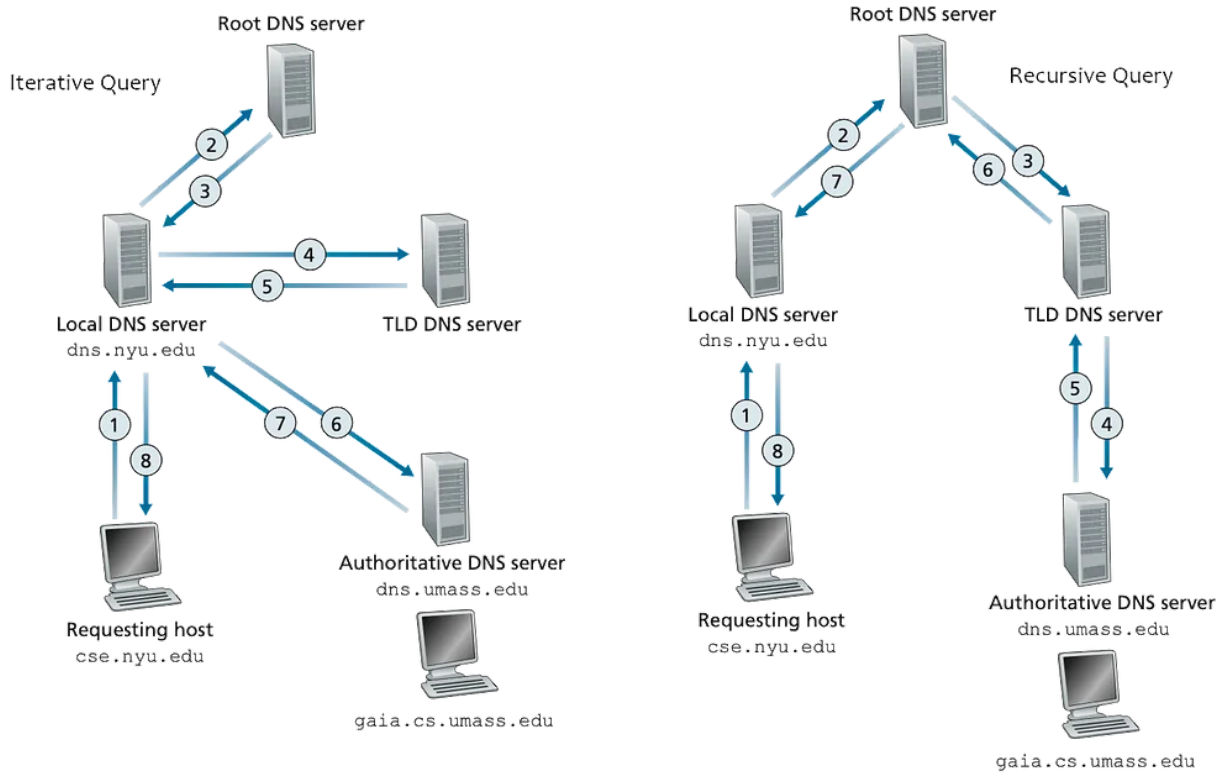
- **이메일 라우팅**: 이메일 서버가 수신자의 도메인을 IP 주소로 변환하여 올바른 메일 서버로 이메일을 전송합니다.
- **네트워크 관리**: IT 관리자가 네트워크 내의 기기와 서비스를 관리하기 위해 DNS를 사용하여 이름을 IP 주소로 쉽게 변환할 수 있습니다.

참고

순방향(Foward) DNS 해석과 역방향(Reverse) DNS 해석

- 순방향 DNS 해석
  - 가장 일반적인 형태의 DNS 조회로, 도메인 이름(예: `www.example.com`)을 해당 도메인의 IP 주소로 변환하는 과정
- 역방향 DNS 해석
  - IP 주소(예: `192.0.2.1`)를 해당하는 도메인 이름으로 변환하는 과정
  - 네트워크 관리, 보안 검사, 로그 분석 등에서 사용
    - 네트워크 상의 호스트가 실제로 어떤 도메인에 속해 있는지를 확인
  - **동작 과정**:
    1. 네트워크 관리자나 시스템이 IP 주소를 제공합니다.
    2. DNS 서버는 IP 주소에 대응하는 도메인 이름을 찾기 위해 PTR(PoinTeR) 레코드를 조회합니다.
    3. PTR 레코드는 `.arpa` 도메인 아래에 역방향으로 기록되며, 각각의 숫자 블록은 점으로 구분되어 역순으로 기록됩니다 (예: `1.2.0.192.in-addr.arpa`).
    4. 해당하는 PTR 레코드를 찾은 DNS 서버는 IP 주소에 매핑된 도메인 이름을 반환합니다.

재귀적 쿼리와 반복적 쿼리



일반적인 DNS 쿼리 실패 처리 흐름

1. **로컬 DNS 서버 캐시 확인**:
   - 캐시에 정보가 없을 경우 상위 DNS 서버로 쿼리를 보냅니다.
2. **상위 DNS 서버의 응답 확인**
   - 상위 DNS 서버로부터 NXDOMAIN, SERVFAIL 등의 오류 응답을 받거나, 타임아웃이 발생할 경우 해당 오류를 클라이언트에게 반환합니다.
3. **재시도 및 대체 DNS 서버 사용**:
   - 일부 경우 로컬 DNS 서버는 동일한 질의를 다른 서버에 재시도하거나, 설정된 대체 DNS 서버로 요청을 보냅니다.
4. **최종 오류 반환**:
   - 응답이 없거나 오류가 지속될 경우, 클라이언트에게 오류를 전달하여 "도메인을 찾을 수 없음", "타임아웃", "DNS 서버 문제" 등의 메시지를 표시합니다.