

『컴퓨터네트워크』 프로그래밍 프로젝트

A Protocol for Domain Name Resolution

2024년 1학기

※ **선택형** 과제임. 본 프로젝트를 제출하고, 제출물 확인 및 인터뷰(데모, 질의응답)를 통과한 학생에게 전체 수강생 종합점수 평균의 0%~10%(완성도 따라 차등)를 가산점으로 부여함

1. 과제 개요

도메인 네임을 IP 주소로 바꿔주는 **가상의** DNS 기반 name resolution 서비스를 구현한다. 클라이언트 프로세스는 local DNS 서버에 query로 도메인 네임을 전달하고, local DNS 서버는 자신의 cache에 해당 query의 답이 있으면 바로, 아니면 클라이언트를 대신하여 다른 DNS 서버(들)를 통해 해당 query의 답을 알아낸 후 이를 클라이언트에게 전달한다. 사용자는 클라이언트에 query를 입력하여, 그 답이 화면에 출력되도록 한다. 이 프로젝트에서 구현하는 클라이언트와 local DNS 서버 사이의 통신, local DNS 서버와 다른 DNS 서버들 사이의 통신 모두 UDP를 사용하되, 동작 확인 데모 시 편의성을 위해 하나의 컴퓨터(노트북) 내(localhost:127.0.0.1)에서 port 번호로만 각 역할의 프로세스를 구분한다.

2. 전체 구조 및 흐름도

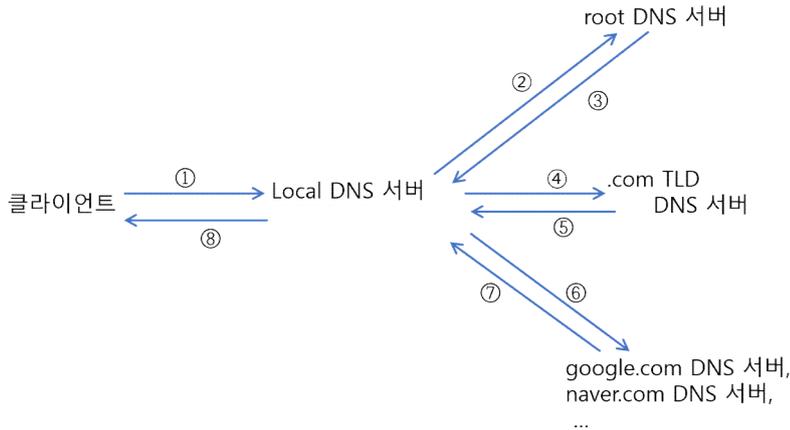


그림 1. Iterative Query Processing Flow

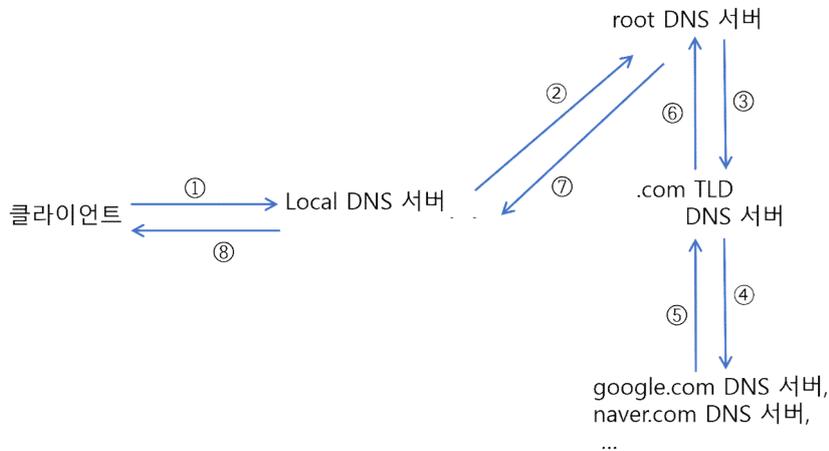


그림 2. Recursive Query Processing Flow

클라이언트는 도메인 yyy.com으로 끝나는 네임을 query로 보내고 이 네임에 대한 IP 주소를 얻고자 한다. Query를 처리하는 과정에서 참조하는 DNS 서버의 cache에는 여러 형태의 RR가 저장되는데, 이 프로젝트에서 사용하는 RR은 A, NS 또는 CNAME의 유형을 가진다. 이하, IP 주소는 모두 가상의, 출력만을 위한 IP 주소로서, 이 프로젝트에서 통신을 위해 특정 IP 주소를 목적지로 지정하는 일은 없다. 예를 들어, local DNS 서버가 .com TLD DNS 서버에 query를 보낼 때 .com TLD DNS 서버의 IP 주소를 목적지로 사용하는 것이 아니라, 해당 .com TLD DNS 서버 역할을 하는 같은 컴퓨터 내의 프로세스 (port 번호로 식별)에게 UDP segment를 전송함으로써 인터넷상의 전달 과정을 **훑내내**는 것이다.

초기(config.txt 파일 읽은 후)에 local DNS 서버는 root DNS 서버의 네임 및 IP 주소, root DNS 서버는 .com TLD DNS 서버의 네임 및 IP 주소, .com TLD DNS 서버는 다수의 .com(예: yyy.com) DNS 서버의 네임 및 IP 주소들을 가지고 있다(각 서버는 config.txt의 각자에게 필요한 부분만 읽어옴). 그리고 각각의 특정 .com DNS 서버는 실행되면서 command-line에 지정된 파일을 읽어서 해당 회사에 속한 다수 도메인 네임 및 IP 주소들을 가지게 된다. 초기 상태에서 클라이언트가 처음 query를 보내는 경우 그림 1 또는 그림 2와 같이 메시지가 교환된다. 클라이언트는 query에 'recursive 처리(참고: 그림 2) 원함'을 지정해서 보낼 수 있으며, 이런 query를 수신한 root DNS 서버 및 .com TLD DNS 서버는 recursive reply를 제공할지를 각자 결정하여 그에 따른 reply를 제공한다. 단, recursive 처리를 원하는 어떤 query를 받은 root DNS 서버가 만약 recursive reply를 제공하기로 하면 .com TLD DNS 서버 또한 반드시 recursive reply를 제공해야 하며(이때 설정값이 거부(off)로 되어 있더라도 제공해야 함), root DNS 서버가 recursive reply를 제공하지 않기로 하면 .com TLD DNS 서버는 recursive reply를 제공할지 독자적으로 결정(현재 설정값 따라)하여 행동한다. Recursive reply를 제공하지 않는다면 iterative 처리(그림 1)가 진행된다.

Local DNS 서버는 클라이언트의 query를 처리하면서 얻은 정보를 자신의 cache에 저장하고 필요시 저장된 정보를 최대한 활용한다. 예를 들어, 그림 1의 흐름을 통해 클라이언트가 machine77.google.com의 IP 주소를 요청해서 얻는 과정에서 local DNS 서버는 .com TLD DNS 서버의 네임 및 IP 주소, google.com의 authoritative DNS 서버의 네임 및 IP 주소, 그리고 machine77.google.com의 IP 주소를 저장한다. 따라서 이후(local DNS 서버에 관련 정보가 cache 된 후), local DNS 서버가 root DNS 서버나 .com TLD DNS 서버 중 하나 이상을 거치지 않고 특정 회사의 authoritative DNS 서버(예: dns.xyznet.com)에 query를 전달할 수 있다. 또는 이전 query를 통해 얻은 정보를 다시 query로 질문받으면 local DNS 서버는 자신의 데이터베이스 내용을 참조하여 바로 클라이언트에게 reply를 제공한다. Local DNS 서버를 포함하여 각 DNS 서버의 cache 저장 공간 및 저장된 정보의 유효기간은 무한대라고 가정한다.

3. 기능 설명

(1) 클라이언트

가상의 회사 dreamnet.com의 머신 swdpt0.dreamnet.com에서 실행되는 응용 프로그램을 클라이언트 프로세스라 하며, 이 클라이언트가 네임 resolution을 위한 query를 최초로 발생시키는 역할을 한다. 클라이언트 프로세스는 실행 후 config.txt 파일(접근 가능 지정된 위치 저장)에 저장된 local DNS 서버의 <네임, IP 주소> 및 port 번호를 읽어오고, 읽은 port 번호로 UDP segment를 보냄으로써 local DNS 서버에 query를 전달한다. Query 생성 및 resolution 과정은 클라이언트가 사용자의 키보드 입력을 통해 ipaddr <name> 명령어를 입력받고 이를 query로 만들어 local DNS 서버에 보냄으로써 시작된다. Local DNS 서버는 이 query의 reply를 얻기까지 필요한 모든 절차를 클라이언트 대신 진행한 후, reply 및 이를 얻기 위해 접근한 서버(들) 정보를 클라이언트에게 제공한다.

실행:

```
% ./client 23001
```

설명: port 번호 23001을 사용하여 접근할 수 있는 클라이언트 프로세스를 생성(이 포트 번호는 UDP segment로 local DNS 서버에 query를 전달할 때 source port 번호로 사용됨)

명령어:

- *ipaddr <name>* : local DNS 서버에게 <name>을 query로 보내고, 도착한 reply와 응답을 얻기 위해 접근한 서버(들)를 순서대로(순서 명확히 구분되도록) 출력한다.

(입력 예) >> ipaddr www.xyznet.com

(화면 출력 예)

```
www.xyznet.com : 203.249.88.112
```

```
(via: local DNS server -> .com TLD DNS server -> local DNS server
```

```
-> xyznet.com DNS server -> local DNS server )
```

(2) local DNS 서버

가상의 dreamnet.com 회사의 네임 서버 dns.dreamnet.com에서 네임 resolution 처리를 위한 실행되는 프로그램을 local DNS 서버 프로세스라 한다. Local DNS 서버 프로세스는 실행 후 config.txt 파일에 저장된 root DNS 서버의 <네임, IP 주소> 및 port 번호를 읽어오고 port 번호를 통해 root DNS 서버와 통신한다. 클라이언트로부터 수신한 query는 즉각 응답(자신의 cache에 query의 답이 있는 경우) 하거나, 또는 reply를 얻는 과정을 통해 답을 얻고 이를 클라이언트에게 전달한다. 다른 DNS 서버에 query를 보낼 때 local DNS 서버는 항상 recursive 처리를 요청하며, 상대 DNS 서버가 recursive 처리를 수락하지 않으면 iterative 처리를 통해 답을 얻는 과정을 진행한다. 대응하는 서버는 recursive 처리 수락을 독자적으로 결정하지만, root DNS 서버가 recursive 처리를 수락하면 하부의 .com TLD DNS 서버는 반드시 recursive 처리를 수락해야 한다(이때 설정값이 거부(off)로 되어 있더라도 수락해야 함). 따라서 local DNS 서버가 root DNS 서버에 먼저 접근한 경우, root DNS 서버가 recursive 처리를 거부하면 하부의 .com TLD DNS 서버는 recursive 처리 거부 또는 수락 중 하나를 독자 선택(현재 설정값에 따라 정해짐)할 수 있다.

답을 구하는 과정에서 얻은 정보는 새로운 정보이면 cache에 추가한다. Query에 대한 reply를 클라이언트에게 전달하면서 local DNS 서버는 이를 얻기까지 접근한 서버(들)를 그 접근 순서대로 적절한 형태로 전달한다(실제로는 클라이언트가 경로까지 알 필요가 없지만, 이 프로젝트에서는 어떤 경로를 통해 답을 얻었는지 확인하는 기능을 제공하려 함).

실행:

```
% ./localDNSserver 23002
```

설명: port 번호 23002를 사용하여 접근할 수 있는 local DNS 서버 프로세스를 생성(이 포트 번호는 클라이언트 프로세스가 UDP segment의 목적지로 사용할 수 있도록 config.txt 파일에 기록된 local DNS 서버 프로세스의 port 번호와 일치해야 함)

명령어:

- *cache* : local DNS 서버의 cache 내용을 화면에 출력한다.

(입력 예) >> cache

(화면 출력 예)

```
dns.rootDNSservice.com : 99.99.99.99 (A)
```

```
dns.comTLDDNSsolution.com : 111.112.113.44 (A)
```

```
www.xyznet.com : server1.xyznet.com (CNAME)
server1.xyznet.com : 203.249.88.112 (A)
...
```

(3) root DNS 서버

가상의 root DNS 서버 dns.rootDNSservice.com에서 root DNS 서버 역할을 하는 실행 프로그램을 root DNS 서버 프로세스라 한다. Root DNS 서버 프로세스는 실행 후 config.txt 파일에 저장된 .com TLD DNS 서버의 <네임, IP 주소> 및 port 번호를 읽어오고, port 번호를 통해 .com TLD DNS 서버와 통신한다. root DNS 서버가 recursive 처리를 수락 또는 거부하는 것은 명령어에 주어진 상태 값(flag: on 또는 off)으로 결정된다.

실행:

```
% ./rootDNSserver 23003
```

설명: port 번호 23003을 사용하여 접근할 수 있는 root DNS 서버 프로세스를 생성(이 포트 번호는 local DNS 서버 프로세스가 UDP segment의 목적으로 사용할 수 있도록 config.txt 파일에 기록된 root DNS 서버 프로세스의 port 번호와 일치해야 함)

명령어:

- *cache* : root DNS 서버의 cache 내용을 화면에 출력한다.

```
(입력 예) >> cache
```

```
(화면 출력 예)
```

```
dns.comTLDDNSsolution.com : 111.112.113.44 (A)
```

```
...
```

- *recursiveFlag on/off* : root DNS 서버가 recursive 처리를 수락(on)/거부(off)하게 설정한다.

```
(입력 예) >> recursiveFlag on // 처리 수락 상태로 설정(현재 수락 상태이면 그대로)
```

```
(화면 출력 예)
```

```
recursive processing : ON
```

(4) .com TLD DNS 서버

가상의 네임 서버 dns.comTLDDNSsolution.com에서 .com TLD의 DNS 서버 역할을 하는 실행 프로그램을 .com TLD DNS 서버 프로세스라 한다. .com TLD DNS 서버 프로세스는 실행 후 config.txt 파일에 저장된 여러 .com 회사들 각각의 authoritative DNS 서버의 <네임, IP 주소> 및 port 번호(UDP 통신에 필요)를 읽어와 네임과 IP 주소의 쌍(들)을 RR 형태로 cache에 저장한다. 각 .com 회사의 authoritative DNS 서버는 해당 회사 도메인의 모든 네임 query를 resolve하기 위한 IP 주소 관련 정보를 제공한다. 단, 예를 들어, www.xyznet.com에 대한 query에 대해 www.xyznet.com이 실제로 머신 server1.xyznet.com을 통해 제공되는 웹 사이트라면 CNAME과 A 두 가지 형태의 RR을 관리, 제공한다. .com TLD DNS 서버가 recursive 처리를 수락 또는 거부하는 것은 명령어에 주어진 상태 값(flag: on 또는 off)으로 결정된다(단 recursive 처리를 수락한 root DNS 서버가 query를 보내면 상태 값 설정된 것과 무관하게 recursive 처리를 수락해야 함).

실행:

```
% ./comTLDDNSserver 23004
```

설명: port 번호 23004를 사용하여 접근할 수 있는 .com TLD DNS 서버 프로세스를 생성(이 포트 번호는 config.txt 파일에 기록된 .com TLD DNS 서버 프로세스의 port 번호와 일치해야 함)

명령어:

- *cache* : .com TLD DNS 서버의 cache 내용을 화면에 출력한다.
(입력 예) >> cache
(화면 출력 예)
 abcdef.com : dns.abcdef.com (NS)
 dns.abcdef.com : 77.88.99.11 (A)
 ...
- *recursiveFlag on/off* : .com TLD DNS 서버가 recursive 처리를 수락(on)/거부(off)하게 설정한다.
(입력 예) >> recursiveFlag on // 처리 수락 상태로 설정(현재 수락 상태이면 그대로)
(화면 출력 예)

(5) 특정 .com authoritative DNS 서버

특정 .com authoritative DNS 서버 프로세스는 실행 후 command-line argument로 주어지는 파일로부터 해당 DNS 서버가 관할하는 다수의 도메인 이름 각각의 IP 주소 또는 real name(CNAME)을 읽어와 RR 형태로 cache 저장한다. 이 프로세스는 다른 DNS 서버(local DNS 서버 또는 .com TLD DNS 서버)가 보낸 query에 대하여 네임 resolution에 필요한 RR 정보를 찾아 이(들)를 응답으로 제공한다.

실행(예): config.txt 파일의 여러 특정 .com 회사에 대하여 회사별로 아래 형태의 command-line 실행
% ./companyDNSserver 10001 <abcdef.txt>

설명: port 번호 10001을 사용하여 접근할 수 있는 abcdef.com authoritative DNS 서버(이름 dns.abcdef.com (이름 생성 규칙: dns.회사이름.com, 회사 이름은 <filename>의 .txt 확장자 이전의 prefix 문자열이라 간주함)) 프로세스를 생성(이 포트 번호는 config.txt 파일에 기록된 abcdef.com TLD DNS 서버 프로세스의 port 번호와 일치해야 함). 또한 파일 abcdef.txt에는 이 dns.abcdef.com의 cache에 저장될 모든 RR가 포함되어, 프로세스 생성 시 이들 RR가 cache 됨.

명령어:

- *cache* : 특정 .com (예: abcdef.com) authoritative DNS 서버의 cache 내용을 화면에 출력한다.
(입력 예) >> cache
(화면 출력 예)
 machine49.abcdef.com : 77.88.99.49 (A)
 ftp.abcdef.com : machine49.abcdef.com (CNAME)
 ...

4. config.txt 및 특정 .com 회사별 dns 서버 cache 파일

- config.txt

```
local_dns_server = [<name> : <ip_address>] <port_number>
    # 실제 내용 예: [dns.dreamnet.com, 100.100.100.99] 23002
root_dns_server = [<name> : <ip_address>] <port_number>
comTLD_dns_server = [<name> : <ip_address>] <port_number>
# 특정 회사들
abcdef_dns_server = [<name> : <ip_address>] <port_number>
    # 실제 내용 예: [dns.abcdef.com, 77.88.99.11] 10001
```

```
xyz_dns_server = [<name> : <ip_address>] <port_number>
ahobho_dns_server = [<name> : <ip_address>] <port_number>
...
```

- abcdef.txt (예: dns.abcdef.com 서버의 cache 내용)

```
machine49.abcdef.com , 77.88.99.49 , A
ftp.abcdef.com , machine49.abcdef.com , CNAME
www.abcdef.com , machine49.abcdef.com , CNAME
...
```

5. 유의 사항

- 유인물에 구체적으로 명시된 요구 사항은 그대로 따라야 하며 그렇지 못한 경우 Fail 처리될 수 있음 (단, 프로세스 실행 방식은 command-line interface 대신 GUI 등을 통해 실행되도록 하는 것 가능). 유인물에 구체적으로 명시되지 않은 부분(자료구조, 알고리즘)은 자유롭게 정해서 구현하되, 그렇게 결정한 이유를 인터뷰에서 설명할 수 있도록 준비할 것
- 최대한 효율적으로, 불필요한 UDP segment 전송이 일어나지 않도록 할 것(예를 들어, cache 내용으로 바로 응답을 보낼 수 있음에도 다른 DNS 서버에게 query를 전달하는 일이 없도록)
- UDP segment의 전송 중 오류에 대비한 처리는 고려할 필요 없음
- 인터뷰 시 데모를 위해 사전에 가능한 모든 시나리오에 대하여 프로그램이 정상 동작함을 확인할 것

6. 제출 기한 및 방법, 인터뷰

- 제출 마감: 5/24(금) 17:00 (늦은 제출은 받지 않음). ※ 이메일 “도착” 시간을 제출 시각으로 봄
- 제출 방법: 이메일(주소: jcpark@hongik.ac.kr)로 아래 전부를 제출함
 - (1) 설계보고서 : 프로그램 구조, 주요 알고리즘 등을 포함하여 본인이 필요하다고 생각하는 내용으로 작성하되, 분량은 A4 4장 이내(double-space, 11pt 기준)로 만들어 첨부할 것
 - (2) 소스코드 출력본 pdf 파일(코드의 **폰트 크기 11pt 이상**(미만인 경우 미제출로 간주)) 첨부할 것
 - (3) 소스코드 : 소스코드 전체를 zip한 후 메일에 첨부하기, **또는**
소스코드 전체가 포함된 github.com의 repository 주소를 이메일 본문에 쓰기(선택함)
- 인터뷰: 5/27(월) 08:30-09:30, 14:00-15:00, 18:30-19:30 (제출물에 이상 없는 대상자에게 인터뷰 시작 시간 개별 통지함) • 필요할 경우 5/28(화)에 추가 시간을 내어 인터뷰 진행할 수 있음
 - (1) 인터뷰는 개인당 최대 10분, T709에서 진행
 - (2) **노트북 컴퓨터 지참**하고, 정해진 시간에 인터뷰 시작할 수 있도록 **미리 준비한 후 대기할 것**
 - (3) 코드 내용 질문, 설계보고서 내용 질문, 동작 실행 데모 등에 대한 답변, 데모 결과를 평가함 (일부의 오작동이나 출력 오류 발생 시 감점하며, 프로그램 실행 불가 발생 시는 Fail 처리함)

7. 부정행위 처벌

타인, 외부의 도움을 받아 프로젝트를 수행한 것이 적발되면, 관련자 전원은 이 과목에서 F를 받으며, 학과 교수진에 부정행위 관련 내용 공개함