<인터넷 프로토콜>

작성자_2018320161_송대선

작성일_05_02

-RR types- : Domain 이름을 키로 준다.

타입	의미	번호
A	IPv4 주소	1
AAAA	IPv6 주소	28
NS	DNS 서버	2
MX	메일 서버	15
CNAME	본명 (canonical name)	5
PTR	IP 주소에 대응되는 도메인 이름	12
SOA	도메인 관리 정보	6
SRV	NS, MX 이외의 서버 정보	33
TXT	임의의 텍스트 정보	16

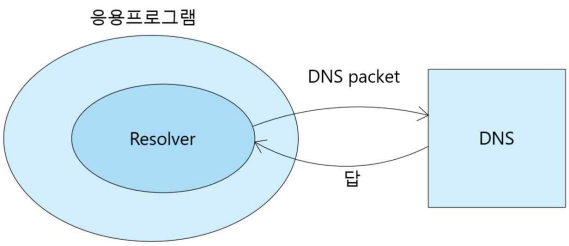
CNAME: 예명, .이 많아지면 외우기가 힘들어진다. -> 예명을 붙여서 외우기 쉽게 한다.
PTR: IP -> 도메인 이름이다. 사실 A와 완전히 다른 데이터베이스를 사용한다.
(서버에만 가끔 물어본다.)
SOA: Start Of Authority
SRV: NS, MX가 아닌 다른 서버의 정보
->요즘은 서버가 굉장히 다양해져서 일일이 물어보기 힘들어진다.
->SRV로 통합해 사용한다.

TXT: 임의의 텍스트 string 필요하면 기록을 해둠

질문을 받아주는 서버는 어떻게 결정되는가? -> DHCP 서버가 줬다.
>server 168.126.63.1 :이렇게 질문을 던질 서버를 바꿀 수도 있다.

>set querytype=a
>cnn.com
-> cnn.com의 ip주소를 줘!
-> 여러 IP주소를 준다.
-> 많은 서버들이 존재한다.

도메인 이름 ------> IP주소로 바꾸는 프로그램이 있는 것이다.
응용 안에 이러한 역할을 하는 친구가 있다.
-> resolver라고 부른다.



Resolver는 DNS 패킷을 만들고, 알고싶은 Type은 뭐고...
하고 DNS에게 물어본다.

Resolver는 DNS가 준 답(IP주소)중에서 가장 위에 있는 답만을 사용한다.
그래서 DNS 서버는 Round-robin DNS로 IP주소를 돌아가면서 맨위에 올린다.

>set query=ns
>cnn.com
->Name server의 Domain name과 해당하는 IP주소를 준다.

>set query=mx
>cnn.com
->@cnn.com으로 가는 mail들을 처리하는 서버들의 Domain name과 IP주소를 준다.

set query=SOA
>cnn.com
->Primary domain name server의 Domain name과 IP주소

responsible mail addr: 이 zone 파일을 관리하는 사람의 이메일 주소
-> @로 바꿔줘야 함.

serial number: 1이면 안 쓴다는 것이다.
-> 주기적으로 바뀌는 게 정상이다.
-> Zone file의 version number를 의미한다.

refresh: 매 6시간 마다 한번씩 secondary를 업데이트한다.
retry: refresh가 일어나지 않았을 때, 10분마다 primary를 찔러본다.
expire: 7일이 지나도 응답이 없으면 그 primary를 버림
TTL: 6시간이 지나면 그 응답은 무효한 것이다.
6시간 뒤 그 응답이 expire된다.

>set querytype=cname
>cnn.com
->이건 예명이 있어야만 응답이 제대로 온다.

답이 오지 않으면, 그 주소가 본명인 것이다.
예명의 예명의 예명의.... 예명을 만드는 일이 가능하다.

multiple한 서버의 주소들을 하나의 machine인 것처럼 예명을 붙일 수 있다.

SRV: _서비스이름._트랜스포트 프로토콜 명._도메인 이름
-> 이 도메인에서 이 프로토콜을 사용해서 이 서비스를 하는 애가 누구야?

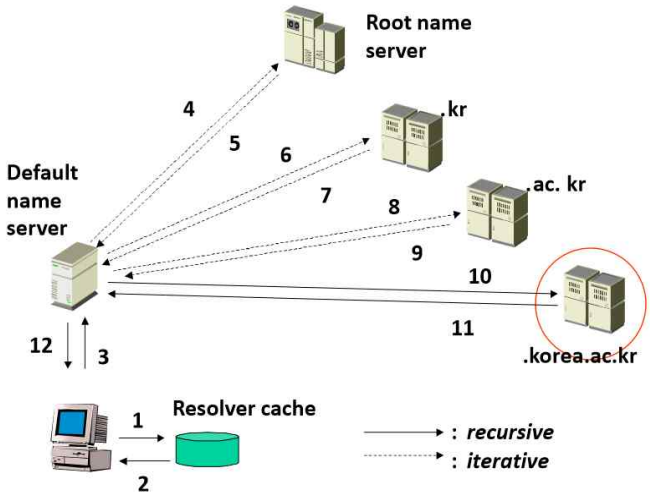
URL을 클릭하는 순간: 일단 type A가 필요하다.
-> resolver에게 물어봄.
(Resolver 안의 cache가 있어서 다시 똑같은 질문을하면 자기가 들고있는 주머니에서 그냥 답을 준다.-TTL이 끝나기 전까지 유효하다.)

Cache를 관리하는 명령어들:
ipconfig/displaydns->dns cache를 보여준다.
ipconfig/flushdns->dns cache를 지운다.

```
C:₩Users₩USER>ipconfig/displaydns

Windows IP 구성

    www.korea.ac.kr
    ----------------------------------------
    데이터 이름 . . . . . : www.korea.ac.kr
    데이터 유형 . . . . . : 1
    TTL(Time To Live) . . : 2254
    데이터 길이 . . . . . : 4
    섹션 . . . . . : 응답
    (호스트) 레코드 . . . : 163.152.6.10

    데이터 이름 . . . . . : kucc.korea.ac.kr
    데이터 유형 . . . . . : 1
    TTL(Time To Live) . . : 2254
    데이터 길이 . . . . . : 4
    섹션 . . . . . : 추가
    (호스트) 레코드 . . . : 163.152.11.6

    데이터 이름 . . . . . : kuccgx.korea.ac.kr
    데이터 유형 . . . . . : 1
    TTL(Time To Live) . . : 2254
    데이터 길이 . . . . . : 4
    섹션 . . . . . : 추가
    (호스트) 레코드 . . . : 163.152.1.1
```

데이터 길이: 레코드의 byte수를 의미함
데이터 길이: 4->레코드의 크기가 4bytes임을 의미한다.

물어봤을 때 cache에 답이 없으면, default name server에게 물어봄.



Authority Name Server와 Default Name Server는 같을 수도, 다를 수도 있지만, 다르다고 생각하는 것이 더 편하다.

Authority Name Server: Zone file을 관리하는 서버
Default Name Server: 그냥 질문만 받아준다.
그러나, Default Name Server에서 Authority Name Server까지 굳이 갈까?
"권한 없는 응답"은 최종 Authority Name Server로부터 온 응답이 아니라는 것을 의미한다.
대부분의 경우, Default Name Server가 가지고 있는 cache에서 답이 다 나온다.
만일 Default Name Server가 가지고 있는 cache에서 답이 없으면, Root name server에게 간다.
->top level에 대한 정보를 가지고 온다.
Root name server->kr.만 보고 kr이 책임지는 서버가 어디에 있는지를 알려줄테니, 거기에 물어봐!
-> 항상 똑같은 질문을 준다.
마침내 응답을 답으면, 일단 resolver의 cache에 저장하고, 그 다음에 컴퓨터에게 답을 준다.
응답의 종류는 2개이다.
-> 답 자체를 준다. (recursive)
-> 답을 알고 있는 서버의 정체를 준다. (iterative)
상층부는 대부분 iterative하다.

DHCP 서버가 자동으로 DNS 서버의 주소를 받아온다.

13 root servers: 맨 꼭대기, root 서버가 전 세계에 13개 있다.
13 gTLD servers: com. ,net. 과 같은 것들을 책임지는 서버이다.

사실 root에게 올라온 query는 거지같은 쿼리일 가능성이 매우 높다.
resolver에 버그가 있어서 그런 일이 자주 있다.
(처음보는 것, 아니면 거지같은 것이다.)

replication을 이용하여 root name server 같은 데를 가곤한다.
anycast기법을 이용한다.
가까운(hop이 적은) root name server로 간다.
-> 한 대가 파괴되어도 괜찮게 돌아갈 수 있게 해준다.

DNS Protocol은 기본적으로 UDP를 사용한다. (특별한 경우에는 TCP를 사용)
-> simple transition이라서 사용한다.
TCP를 쓰는 경우? -> 질문이나 대답이 너무 길어질 때
질문과 응답을 줄여서 IP fragment가 안 일어나게 노력한다. (512B를 안넘기려 함)
port 번호는 53이다.
질문은 512B를 안 넘는데, 응답은 512B를 자주 넘긴다.
512B가 넘으면, TCP를 사용하고 port는 53을 사용하여 다시 똑같은 질문을 보낸다.

UDP를 사용하는 packet은 항상 identifier field가 필요하다.
-> 먼저 질문을 했다고, 먼저 답변이 온다는 보장이 없기 때문!

(학생 질문: Zone file이 있으면 무조건 Authority Name Server이다.)

Identification	Flags
# of questions	# of answer RRs
# of authority RRs	# of additional RRs
Questions (variable #)	
Answers (variable # of RRs)	
Authority (variable # of RRs)	
Additional information (variable # of RRs)	

RR은 referance를 의미한다.

wire shark로보자
Transition ID (Identifier) : 0x6693
Flags
number of questions: question이 여러 개일 수 있다.
number of anwser RRs:
number of authority RRs:
number of additional RRs:

Questions: 질문이 여러 개일 수도 있다.
Answers: 너가 질문한 Domain name은 사실 예명이야.
IP는 이거야.
Authority: Authority name server로 간다. (NS로 준다)
-> 이렇게 해서 recursive한 답을 주는 Authority name server를 언젠가는 찾는다.
-> 나를 괴롭히지 마! 라는 의미이다.
Additional information: 사실 물어본 Domain name의 진짜 이름은 이거다.

QR: 0이면 질문(query), 1이면 답변(response)이다.
Opcode:
Authority Answer: 0이면 cache에서 온 것이다. 1이면 Authority name server에서 온 것이다.

Truncated: 잘렸다.
질문은 그럴 일이 적다.

한편, 512B라는 기준은 어디에서부터 온것인가?
-> 인터넷에서 쓰이는 interface는 적어도 최소 576B은 처리할 수 있어야 한다.
(MTU가 576B까지는 담을 수 있어야 한다.)

IP header가 최소 20B이고, UDP header가 8B정도 된다.
->약 28B정도 된다.
576-28=548
여유 공간으로 적당히 빼고나니, 약 512B정도가 남는다.
이 정도로 설정하니, 576B를 잘 넘기지 않았다.

응답은 512B을 훨씬 넘길 수도 있다.
-> 이러면 잘린다. 512B로 잘라서 잘린 부분만 보냄.
-> host는 이것을 받아서 그냥 버리고, 동일한 질문을 TCP로 보냄.

Recursive Desired: 1이면, “그냥 제발 답만 줘!”라고 떼쓰는 거다.
Recursive Available: 1이면, “그래, 그냥 답만 줄게...”하는 것이다.
-> 상위 DNS 서버는 0으로 설정되어 있다. “안돼! 나 바쁘니까 귀찮게 하지마!”라고 한다.

ResponseCode=0 그냥 0으로 채워져 있으면 정상이다.

DNS Protocol은 오직 FQDN만을 사용한다.
www.cnn.com. 과 같은 FQDN만을 사용한다.
03 77 77 77
-> 03은 이 label은 3글자이다. 라는 것을 알려준다.
즉, 77 77 77. == www. 이라는 뜻이다.

cnn.
-> 03 63 6e 6e

com.
-> 03 63 6f 6d

.
-> 00
(00이 나와버리면, Domain Name의 끝이고, FQDN라는 의미이다.)

패킷사이즈를 줄이기 위하여 포인터를 사용하기도 한다.
-> Domain name이 나와야 할 자리에 “C”가 나오면 포인터이다.
-> 포인터는 header를 제외한 body가 시작되는 부분을 기준으로 센다.
ex) C0 50 -> body로부터 80번째 byte부터 시작이다.

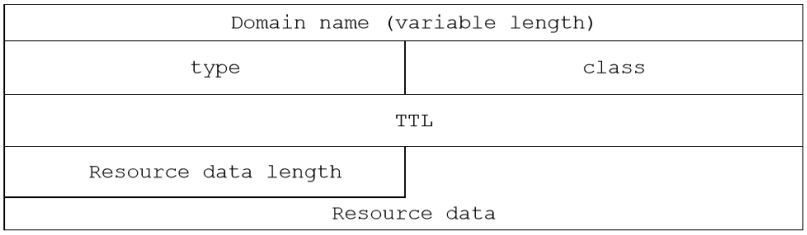
ex)
xxx. A . B . C .
yyy. A . B . C .
일 때, yyy. 다음에 포인터로 A . B . C .를 포인터 가능

Query name (variable length)	
Query type	Query class

```
> Internet Protocol Version 4, Src: 172.30.1.4, Dst: 168.126.63.1
> User Datagram Protocol, Src Port: 56743, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0xffa5
  > Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v chat-pa.clients6.google.com: type A, class IN
        Name: chat-pa.clients6.google.com
        [Name Length: 27]
        [Label Count: 4]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
```

Query: recursive query가 대부분이다.
재전송 -> 질문 안에 Question을 여러 개 넣을 수 있다.

Name: DNS 주소
Type: A, AAAA, 등, 뭘 물어보고 싶은가?
Class: 인터넷은 항상 1번이 된다.




```
▾ Queries
  › cdn.onenote.net: type A, class IN
▾ Answers
  ▾ cdn.onenote.net: type CNAME, class IN, cname cdn.onenote.net.edgekey.net
      Name: cdn.onenote.net
      Type: CNAME (Canonical NAME for an alias) (5)
      Class: IN (0x0001)
      Time to live: 2993
      Data length: 26
      CNAME: cdn.onenote.net.edgekey.net
  ▾ cdn.onenote.net.edgekey.net: type CNAME, class IN, cname e1553.dspg.akamaiedge.net
      Name: cdn.onenote.net.edgekey.net
      Type: CNAME (Canonical NAME for an alias) (5)
      Class: IN (0x0001)
      Time to live: 580
      Data length: 24
      CNAME: e1553.dspg.akamaiedge.net
  ▾ e1553.dspg.akamaiedge.net: type A, class IN, addr 104.75.33.244
      Name: e1553.dspg.akamaiedge.net
      Type: A (Host Address) (1)
      Class: IN (0x0001)
      Time to live: 3
      Data length: 4
      Address: 104.75.33.244
```

Name: Domain Name
Type: A, AAAA 등...
class: 인터넷에서는 1!
TTL:
Data Length:
CNAME (resource data)

Answer:
Authoritative positive
->Normal answer from authoritative server

Positive Answer
-> from non-authoritative server

Referral
->Next server to ask (“authority”)

Negative
-> Even authoritative name server cannot find the answer
(완전 잘못된 Domain Name이라고 해도 cache에 저장된다.)

Microsoft 사건:
DNS 서버가 터졌다고, 바로 접속이 안되었을까?
아니다. 잠깐 괜찮았을 것이다.
Cache에 저장된 값의 TTL이 다 될 때까지는 괜찮았을 것이다.
그러나, 시간이 지나며 Cache에 저장된 Microsoft DNS 서버와 관련된 정보들이 서서히 사라지면서 DNS 번역이 안 되기 시작했을 것이다.

Round-robin DNS
-> Cache는 맨 위에 있는 DNS 주소만을 사용하기 때문에,
DNS 서버는 여러 대의 DNS 주소를 돌아가면서 cache에게 제공하지 않으면,
하나의 Authority Name Server만을 사용하게 되는데, 이러면 여러 대의 Authority Name Server를 설치한 이유가 없게 된다.
따라서 여러 Authority Name Server를 돌아가면서 사용하기 위해 Round-robin DNS를 이용한다.

-> 만약에 똑같은 답이 오면, 그건 그 사이에 몇 바퀴를 돌아서 온 답일 수 있다.

-질문-
Secondary Authority Name Server는 Primary Authority Name Server의 Zone file을 자주 update 해줘야 한다. -> 그래야 간극이 안생긴다.