6.033 Computer System Engineering

Spring 2009

Hands-on 4: Internet Domain Name System

Today's hands-on exercise is designed to give you a quick introduction to the Internet's Domain Name System. This is an example of a naming system which all of you use on a daily basis --- in fact you used it to get to this web-page! To prepare for this assignment, please read Appendix 4-A of the class notes, titled "Case study of the Internet Domain Name System". This should give you a good general idea of how the DNS works.

Introduction

In order to help explore the domain name system, there is a tool called `dig`, short for Domain Information Groper. We will be making use of `dig` in this assignment. `dig` should be available on all recent Athena workstations. It should work by default, but if it does not, please try running `add watchmaker` first. If that still does not work, try an Athena Sun workstation.

Here is an example usage of `dig`:

```
athena% dig slashdot.org

; <<>> DiG 9.3.1 <<>> slashdot.org
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 997
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;slashdot.org.                  IN      A

;; ANSWER SECTION:
slashdot.org.           3600    IN      A       216.34.181.45           (*)

;; AUTHORITY SECTION:
slashdot.org.           86399   IN      NS      ns-2.ch3.sourceforge.com.
slashdot.org.           86399   IN      NS      ns-1.ch3.sourceforge.com.
slashdot.org.           86399   IN      NS      ns-1.sourceforge.com.

;; ADDITIONAL SECTION:
ns-1.ch3.sourceforge.com. 172800 IN     A       216.34.181.21
ns-1.sourceforge.com.   172800  IN      A       208.122.22.23
ns-2.ch3.sourceforge.com. 172800 IN     A       216.34.181.22
```

```
;; Query time: 69 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Wed Mar 11 17:32:51 2009
;; MSG SIZE rcvd: 170
```

The output tells us a lot of information about our DNS request and the response to it. At the bottom, we can see that the query was sent to our default server (`127.0.0.1`), and that it took roughly 69 msecs to respond. Most of the information we are interested in is in the `ANSWER` section, marked with a **`(*)`** above. Let's examine that section more closely:

```
        ;; ANSWER SECTION:
        slashdot.org.    3600    IN      A    216.34.181.45
          name          expire  class   type     data (IP)
```

We can see that this result is of type `A`, an address record: it is telling us that the IP address for the name "`slashdot.org`" is `216.34.181.45`. The expiry time field "`3600`" indicates that this record/entry is valid for 3600 seconds (1 hour). You can ignore the "class" field; this is nearly always `IN` for Internet.

The authority section contains records of type `NS`, which give the names of DNS servers that have name records for a particular domain. Here, we can see that three DNS servers (`ns-1.ch3.sourceforge.com.`, `ns-1.sourceforge.com.` and `ns-2.ch3.sourceforge.com.`) are responsible for answering requests for names in the `slashdot.org` domain. (Note that in all of these examples, the exact results you get may be slightly different.)

We can ask a specific server (instead of the default) for information about a host by using the following syntax:

```
athena% dig @amsterdam.lcs.mit.edu slashdot.org

; <<>> DiG 9.3.1 <<>> @amsterdam.lcs.mit.edu slashdot.org
; (1 server found)
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1988
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;slashdot.org.                  IN      A

;; ANSWER SECTION:
slashdot.org.           3600    IN      A       216.34.181.45

...[output truncated]
```

We can also see that these queries are resulting in the *recursive* searches described in section 4.1.1 of the class notes by examining the `flags` line. The `rd` (recursion desired) flag indicates that `dig` requested a recursive lookup, and the `ra` (recursion available) flag indicates that the server permits recursive lookups (some do not).

`dig` only shows us the final result of the recursive search. One way for us to mimic the individual steps of a recursive search is to send a request to a particular DNS server and ask for no recursion. For the former, we can give an `@server` argument to `dig`. For the latter, we can pass the `+norecurs` flag.

For example, to send a non-recursive query to one of the root servers:

```
athena% dig @a.ROOT-SERVERS.NET www.slashdot.org +norecurs

; <<>> DiG 9.3.1 <<>> @a.ROOT-SERVERS.NET www.slashdot.org +norecurs
;; (1 server found)
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1888
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 12

;; QUESTION SECTION:
;www.slashdot.org.              IN      A

;; AUTHORITY SECTION:
org.                    172800  IN      NS      B0.ORG.AFILIAS-NST.org.
org.                    172800  IN      NS      A0.ORG.AFILIAS-NST.INFO.
org.                    172800  IN      NS      A2.ORG.AFILIAS-NST.INFO.
org.                    172800  IN      NS      D0.ORG.AFILIAS-NST.org.
org.                    172800  IN      NS      C0.ORG.AFILIAS-NST.INFO.
org.                    172800  IN      NS      B2.ORG.AFILIAS-NST.org.

;; ADDITIONAL SECTION:
A0.ORG.AFILIAS-NST.INFO. 172800 IN      A       199.19.56.1
A0.ORG.AFILIAS-NST.INFO. 172800 IN      AAAA    2001:500:e::1
A2.ORG.AFILIAS-NST.INFO. 172800 IN      A       199.249.112.1
A2.ORG.AFILIAS-NST.INFO. 172800 IN      AAAA    2001:500:40::1
B0.ORG.AFILIAS-NST.org. 172800  IN      A       199.19.54.1
B0.ORG.AFILIAS-NST.org. 172800  IN      AAAA    2001:500:c::1
B2.ORG.AFILIAS-NST.org. 172800  IN      A       199.249.120.1
B2.ORG.AFILIAS-NST.org. 172800  IN      AAAA    2001:500:48::1
C0.ORG.AFILIAS-NST.INFO. 172800 IN      A       199.19.53.1
C0.ORG.AFILIAS-NST.INFO. 172800 IN      AAAA    2001:500:b::1
D0.ORG.AFILIAS-NST.org. 172800  IN      A       199.19.57.1
D0.ORG.AFILIAS-NST.org. 172800  IN      AAAA    2001:500:f::1

;; Query time: 84 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
;; WHEN: Wed Mar 11 17:45:41 2009
;; MSG SIZE  rcvd: 436
```

As you can see, the server does not know the answer and instead provides information about the servers most likely to be able to provide authoritative information. In this case, the best the root server knows is the identities of the servers for the `org.` domain.

With this in mind, let's do some simple exercises. Please turn in answers to the questions below for Tuesday's recitation. You should submit answers *only* to the questions asked. In particular, please do not include pages of output from `dig` unless specifically requested.

I. Getting started

- **(1)** Using `dig`, find the IP address of `thyme.lcs.mit.edu`. What is the IP address?
- **(2)** What is the expiration time for this record?
- **(3)** The `dig` answer for `thyme` includes a record of type `CNAME`. In the terminology of chapter 4, what does `CNAME` mean?

- **(4)** What are the IP addresses for `ai` and `ai.` (note the dot at the end) respectively? Use the `host` command for this question. `host` is a simpler tool that performs a DNS lookup and prints only the answer.
- **(5)** Why are the results different? Examining the local machine's `/etc/resolv.conf` file, what can you say about the context of DNS searches for `ai` and `ai.`?

II. Understanding hierarchy

For this problem, you will go through the steps of resolving a particular hostname, by **iterating** through a series of servers, just like a regular server might. Assuming it knows nothing else about a name, a DNS resolver will ask a well-known *root server*. The root servers on the Internet are in the domain `root-servers.net`. One way to get a list of them is with the command:

```
athena% dig . ns
```

- **(6)** Why does this particular command return the names of the root nameservers?
- **(7)** Use `dig` to ask *one* of the root servers the address of `lirone.csail.mit.edu`, *without* recursion. What command do you use to do this?
- **(8)** It is unlikely that these servers actually know the answer so they will *refer* you to a server (or list of servers) that might know more. Go through the hierarchy from the root **without recursion**, following the referrals manually, until you have found the address of `lirone.csail.mit.edu`. What commands did you use to do this, and what is the IP address?

III. Understanding caching

These few queries should show you how your local machine's DNS cache works.

- **(9)** Ask your default server for information, without recursion, about the host `www.dmoz.org`. What command did you use? Does it have the answer in its cache? How do you know? How long did this query take? If this information was cached, please find some other host name that is not cached and do this section with that other host.
- **(10)** Now, ask your default server this same query but *with* recursion. It should return an answer for you. How long did this take?
- **(11)** Finally, ask your default server again without recursion. How long does this request take? Has the cache served its purpose?