

# Lab 8 DNS

DNS (Domain Name System) is the system and protocol that translates domain names to IP addresses and more. It is covered in Chapter 7.1 of the textbook. Review the text section before doing this lab.

## Objective

- Know the format of DNS message.
- Understand architecture and operations of DNS.
- Understand the usage of DNS cache.

## Requirements

You need to install following tools on your computer beforehand:

- **Wireshark:** This lab uses the Wireshark software tool to capture and examine a packet trace. Refer to previous labs for details.
- **nslookup:** *nslookup* is a network administration command-line tool available in many computer operating systems for querying the DNS to obtain domain name or IP address mapping, or other DNS records.
- **dig:** This lab uses *dig* to issue DNS request and observe DNS responses. *dig* is a flexible, command-line tool for querying remote DNS servers. It comes installed on Mac OS. On Window, you can download *dig* from ISC's BIND web site as part of the *bind* download (Check for online instructions to set up *dig* on Windows). On Linux, install *dig* with your package manager. It is normally part of a *dnsutils* or *bindutils* package.
- **Tools for DNS cache:**
  - Windows: "*ipconfig /displaydns*" displays the contents of the DNS cache, which includes both entries preloaded from the local Hosts file and any recently obtained resource records. "*ipconfig /flushdns*" flushes and resets the contents of the DNS client resolver cache.
  - Linux: There is no OS-level DNS caching on Linux unless a caching service such as *Systemd-Resolved*, *DNSMasq*, or *Nscd* is installed and running. Search on Google or check on the course website.
  - MacOS: depends on versions of the OS. Search on Google or check on the course website.

## Exercise

In a typical network, your computer contacts a **local DNS nameserver** to resolve domain names to IP addresses. The local nameserver may be another computer in your company network, a computer at your ISP, or your wireless AP. It exchanges a series of messages with **remote DNS nameservers** all over the Internet to perform

the resolution. The setup is as shown in the Figure 1.

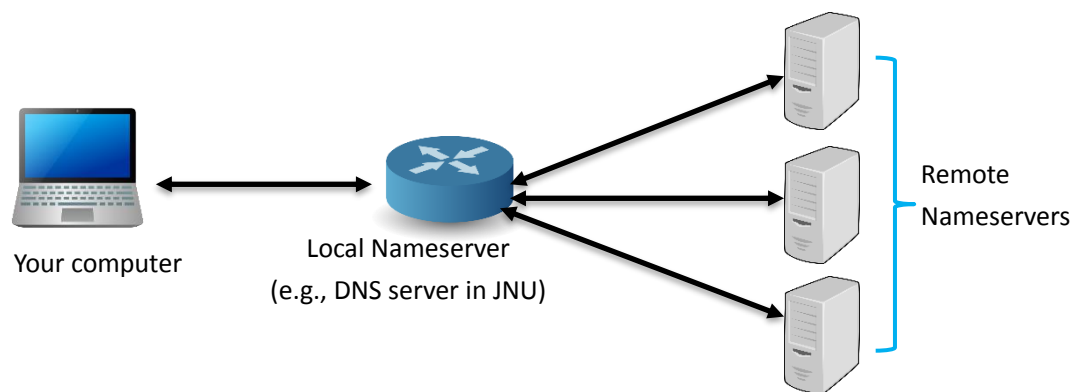


Figure 1: Network topology

### Task 1: Domain Name Resolution with *nslookup*

In this task, we will use *nslookup* to conduct several simple experiments on different types of domains:

1. Generic domain:

Run “*nslookup www.python.org*” command in the command line window and answer following questions:

- What is the corresponding IP address of “*www.python.org*”?
- What is the top-level domain for “*www.python.org*”?

2. Country domain:

Run “*nslookup www.gd.gov.cn*” command and answer the following questions:

- What is the corresponding IP address of “*www.gd.gov.cn*”?
- What are the top level domain, second-level domain and third-level domain of “*www.gd.gov.cn*”?

3. Reverse domain:

Run “*nslookup 8.8.8.8*” command and answer the following questions:

- What is the corresponding domain name of IP address 8.8.8.8?
- What are the top-level and second-level domain names of reverse domain?

### Task 2: Manually Name Resolution with *dig*

This task will explore how a local nameserver resolves a DNS name. We will let ***your computer pretend to be the local nameserver*** and issue requests to remote nameservers using the *dig* tool.

1. Pick a domain name to resolve, such as [www.jnu.edu.cn](http://www.jnu.edu.cn), [www.gd.gov.cn](http://www.gd.gov.cn) or [www.gznet.edu.cn](http://www.gznet.edu.cn). Find the IP address of one of the **root nameservers** by searching on the web. You need this information to begin the name resolution process, and nameservers are provided with it as part of their configuration.

**Tips:**

- *gznet.edu.cn* is the domain for the Network Center of Southern China for CERNET (教育网华南地区网络中心). JNU campus network is connected


to the CERNET through this center.

- The [Wikipedia article](#) on *root name servers* includes IP address of all root nameservers from *a* to *m*. Any one of these should do, as they hold replicated information. For example, the IP address of “A” root nameserver is [198.41.0.4](#). Following is the figure from our lecture slides “L12-Application Layers-I.pdf”.

**13 root name servers** worldwide, each may contain multiple machines using **anycast**

|  |                |
|--|----------------|
| • A.ROOT-SERVERS.EDU. (NS.INTERNIC.NET)  | 198.41.0.4     |
| • B.ROOT-SERVERS.NET. (NS1.ISI.EDU)      | 192.228.79.201 |
| • C.ROOT-SERVERS.NET. (C.PSI.NET)        | 192.33.4.12    |
| • D.ROOT-SERVERS.NET. (TERP.UMD.EDU)     | 128.8.10.90    |
| • E.ROOT-SERVERS.NET. (NS.NASA.GOV)      | 192.203.23     |
| • F.ROOT-SERVERS.NET. (NS.ISC.ORG)       | 192.5.5.241    |
| • G.ROOT-SERVERS.NET. (NS.NIC.DDN.MIL)   | 192.112.36.4   |
| • H.ROOT-SERVERS.NET. (AOS.ARL.ARMY.MIL) | 128.63.2.53    |
| • I.ROOT-SERVERS.NET. (NIC.NORDU.NET)    | 192.36.148.17  |
| • J.ROOT-SERVERS.NET. (VeriSign)         | 198.41.0.10    |
| • K.ROOT-SERVERS.NET. (RIPE NCC)         | 193.0.14.129   |
| • L.ROOT-SERVERS.NET. (ICANN)            | 198.32.64      |
| • M.ROOT-SERVERS.NET. (WIDE, Japan)      | 202.12.27.33   |

<http://www.root-servers.org/>



A Cisco 7301 router and a Juniper M7i, part of the K root-server instance at AMS-IX (Amsterdam Internet Exchange).

Figure 2: Root name servers from Lecture slides

2. **Close all unnecessary browser tabs and applications.** Browsing web sites will generate DNS traffic as your browser resolves domain names to connect to remote servers. We want to minimize browser activity initially so that we capture only the intended DNS traffic.
3. Issue a request to one root nameserver to perform the first step of the resolution. The following example will use “A” root nameserver:

```
dig @198.41.0.4 www.gznet.edu.cn
```

The reply from the root nameserver does not provide the full name resolution, but it does tell us about *nameservers closer to having the information* for us to contact (see Figure 3). In this case, it is nameservers who know about the “.cn” domain. Multiple nameservers are given as alternative choices, and the reply helpfully includes their IP addresses; we can see both IPv6 and IPv4 addresses.

```
C:\Windows\system32\cmd.exe
C:\Users\Lenovo>dig @198.41.0.4 www.gznet.edu.cn

; <<>> DiG 9.16.9 <<>> @198.41.0.4 www.gznet.edu.cn
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41695
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 8, ADDITIONAL: 11
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.gznet.edu.cn.                IN      A

;; AUTHORITY SECTION:
cn.                172800  IN      NS      c.dns.cn.
cn.                172800  IN      NS      g.dns.cn.
cn.                172800  IN      NS      b.dns.cn.
cn.                172800  IN      NS      ns.cernet.net.
cn.                172800  IN      NS      e.dns.cn.
cn.                172800  IN      NS      f.dns.cn.
cn.                172800  IN      NS      a.dns.cn.
cn.                172800  IN      NS      d.dns.cn.

;; ADDITIONAL SECTION:
c.dns.cn.          172800  IN      A        203.119.27.1
g.dns.cn.          172800  IN      A        66.198.183.65
b.dns.cn.          172800  IN      A        203.119.26.1
ns.cernet.net.     172800  IN      A        202.112.0.44
e.dns.cn.          172800  IN      A        203.119.29.1
f.dns.cn.          172800  IN      A        195.219.8.90
a.dns.cn.          172800  IN      A        203.119.25.1
a.dns.cn.          172800  IN      AAAA     2001:dc7::1
d.dns.cn.          172800  IN      A        203.119.28.1
d.dns.cn.          172800  IN      AAAA     2001:dc7:1000::1

;; Query time: 169 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
;; WHEN: Thu Dec 03 16:21:50 中国标准时间 2020
;; MSG SIZE rcvd: 372

C:\Users\Lenovo>
```

Figure 3: First step resolution using *dig*

4. Now you have nameservers closer to having the answer. Issue a request to one of these nameservers for further resolution.

**Tips:**

- When there are alternatives to choose, you can choose a random one (e.g., We prefer [IPv4 nameservers and select the first one in alphabetical order](#). If this nameserver has multiple IP addresses, [select the numerically smallest IP address](#)). However, the same result will likely be obtained for all IP address since the DNS information is replicated
- Note that future name resolutions are likely to be a much shorter sequence because they can use cached information. For example, if you already know the name of the “.cn” nameserver, when you look up a different domain name in “.cn”, you can start there, or even closer to the final nameserver depending on what you have cached. You do not need to start again at the root

nameserver.

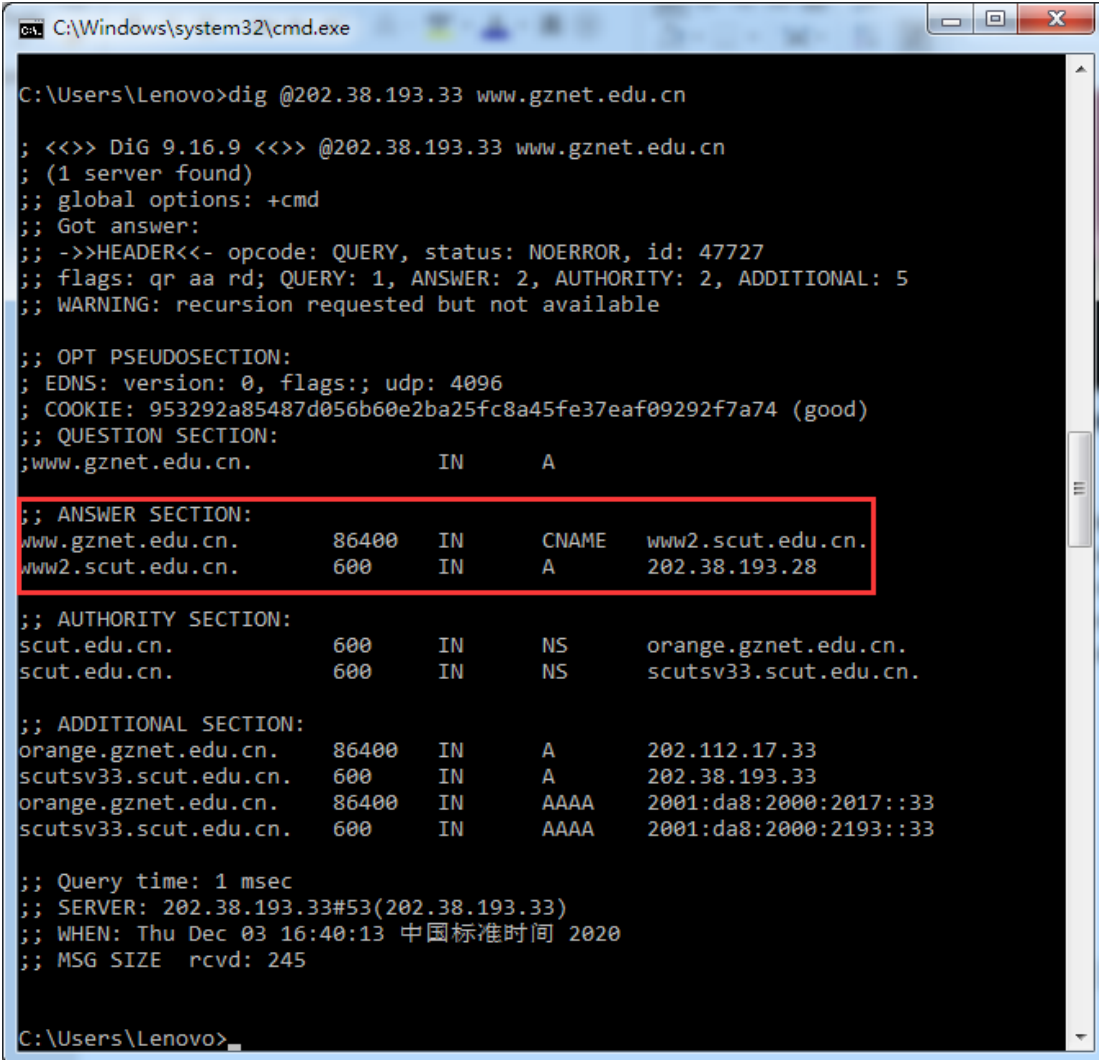
In the above Figure 3, the nameserver at IP address 203.119.27.1, which is authoritative for “.cn”, is the remote nameserver to contact next, i.e.,

*dig @203.119.27.1 www.gznet.edu.cn*

5. Continue the resolution process with *dig* **until you complete the resolution**, i.e., obtain the IP addresses of [www.gznet.edu.cn](http://www.gznet.edu.cn) (see Figure 4).

**Tips:**

- The last name server **may only return a CNAME record** without IP address (i.e., “A” record). **In this case, you need to conduct another query for the domain name in the CNAME record.**
- DNS Records Type: (a) **CNAME (Canonical Name) record** is a type of resource record in DNS that maps one domain name (an *alias*) to another (the canonical name). (b) **A or Address record** (also known as a host record) links a domain to the 32-bit IPv4 address of a computer hosting that domain's services. (c) **AAAA record** links a domain to the 128-bit IPv6 address of a computer hosting that domain's services. (d) **NS (Name Server) record** delegates a DNS zone to use the given authoritative name servers.



```
C:\Windows\system32\cmd.exe
C:\Users\Lenovo>dig @202.38.193.33 www.gznet.edu.cn

; <<>> DiG 9.16.9 <<>> @202.38.193.33 www.gznet.edu.cn
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47727
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 2, ADDITIONAL: 5
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 953292a85487d056b60e2ba25fc8a45fe37eaf09292f7a74 (good)
;; QUESTION SECTION:
;www.gznet.edu.cn.          IN      A

;; ANSWER SECTION:
www.gznet.edu.cn.          86400   IN      CNAME   www2.scut.edu.cn.
www2.scut.edu.cn.          600     IN      A        202.38.193.28

;; AUTHORITY SECTION:
scut.edu.cn.                600     IN      NS       orange.gznet.edu.cn.
scut.edu.cn.                600     IN      NS       scutsv33.scut.edu.cn.

;; ADDITIONAL SECTION:
orange.gznet.edu.cn.        86400   IN      A        202.112.17.33
scutsv33.scut.edu.cn.        600     IN      A        202.38.193.33
orange.gznet.edu.cn.        86400   IN      AAAA     2001:da8:2000:2017::33
scutsv33.scut.edu.cn.        600     IN      AAAA     2001:da8:2000:2193::33

;; Query time: 1 msec
;; SERVER: 202.38.193.33#53(202.38.193.33)
;; WHEN: Thu Dec 03 16:40:13 中国标准时间 2020
;; MSG SIZE rcvd: 245

C:\Users\Lenovo>
```

Figure 4: Obtain Answers using *dig*

6. Now, launch Wireshark and start a capture with a filter of “*dns*” or “*udp port 53*”. Repeat the *dig* commands from the previous steps. This time, you should see the DNS request and reply packets that correspond to your commands captured in the trace window.

**Tips:**

- There may be some background DNS traffic originating from your computer if any process needs to resolve names to make a network connection. We are assuming that there will be little of this traffic.
7. Stop the Wireshark, save the trace and inspect the trace you captured (Check details of both DNS query and response messages).
  8. Draw a figure that shows the sequence of remote nameservers that you contacted and the domain for which they are responsible.
  9. You can also try to trace DNS path automatically using *dig*:
    - Instead of using above steps to resolve a domain manually, *dig* also allows tracing the DNS lookup path by using the “+trace” option, e.g.,  
*dig +trace www.gznet.edu.cn*
    - The “+trace” option makes *iterative queries* to resolve the name lookup. It will query the name servers starting from the root and subsequently traverses down the namespace tree using iterative queries following referrals along the way.

Answer the following questions:

For DNS query:

1. How many bits long is the Transaction ID? Based on this length, take your best guess as to how likely it is that concurrent transactions will use the same transaction ID.
2. Which flag bit and what values signifies whether the DNS message is a query or response?
3. How many bytes long is the entire DNS header?

For DNS response:

4. For the initial response, in what section are the names of the nameservers carried? What is the Type of the records that carry nameserver names?
5. Similarly, in what section are the IP addresses of the nameservers carried, and what is the Type of the records that carry the IP addresses?
6. For the final response, in what section is the IP address of the domain name carried?
7. What's the meaning of CNAME? And what happens if two CNAMEs points to each other?  
(Tips: Check the “Reference Materials” of Lab 8 on course website.)

### Task 3: DNS cache and Host file

#### DNS cache:

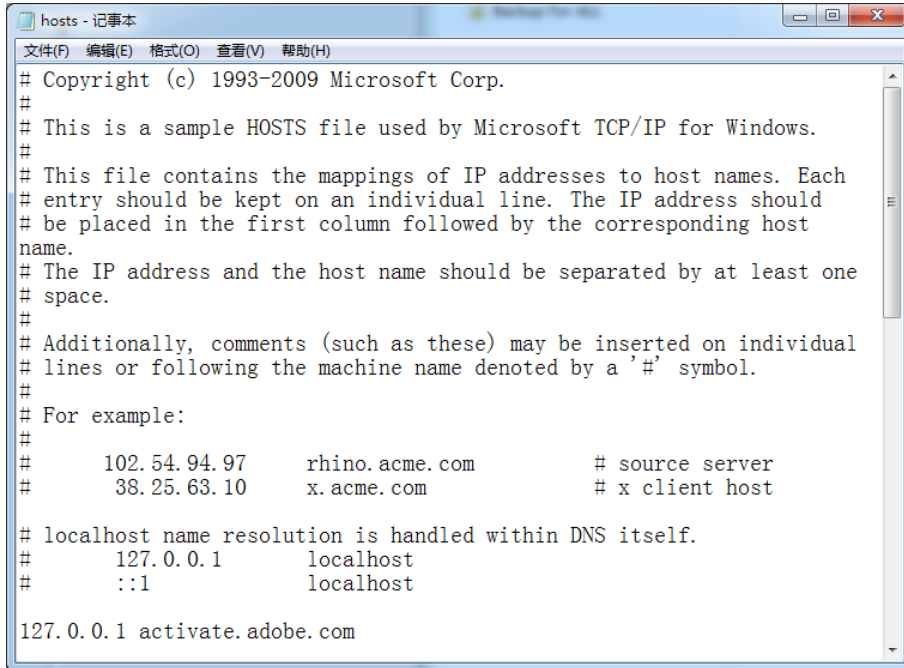
1. On Windows, execute “*ipconfig /flushdns*” command to clear DNS cache.  
**Tips:** Linux and MacOS don’t support *ipconfig* tool, please search on Google or check on course website for solution to flush and display DNS cache.
2. Launch Wireshark to capture packets and setup filter condition to extract both DNS and ICMP protocol, e.g., “*dns or icmp*”.
3. Execute “*ping www.baidu.com*” command, and then execute “*ipconfig /displaydns*” command to display DNS cache. Locate the corresponding records about the domain name in DNS cache.
4. Execute “*ping www.baidu.com*” command again.
5. Stop capturing, analyze the captured trace and the contents in DNS caches.
6. According to your trace, draw a figure of the message interaction process (i.e., sequence of messages) in this task, including both ICMP and DNS query/reply messages.  
**Tips:** There are three nodes in your figure: your computer, the local name server and the remote server of *www.baidu.com*.

Answer the following questions:

1. Will the second *ping* trigger DNS queries?
2. Describe the usage of DNS cache.

#### Host file:

7. You can also check the usage “hosts” file using *ping* and Wireshark.  
**Tips:**
  - “hosts” file: On Windows, it is “Windows\System32\drivers\etc\hosts”. On Linux/MacOS, it is “/etc/hosts”.You can use any text editors to modify its content, e.g., *notepad* or *vim*.  
You may need to use *Administrator* on Windows, or “*sudo*” on Linux.
8. Assign the **correct** IP address for a domain (e.g., [www.baidu.com](http://www.baidu.com) or [www.jnu.edu.cn](http://www.jnu.edu.cn)) in the host file, clear the DNS cache (i.e., Step 1), and then use *ping* and Wireshark to check whether the host file is working (e.g., whether the *ping* works and whether you can capture DNS request for the domain).
9. Assign an **incorrect** IP address for a domain (e.g., [www.baidu.com](http://www.baidu.com) or [www.jnu.edu.cn](http://www.jnu.edu.cn)) in the host file, and then check whether you can *ping* the domain.  
**Tips:**
  - As to the incorrect IP address for the domain, you can use 127.0.0.1 (i.e., your local computer) or IP address of another computer.
  - Check the IP address that responses *ping*, i.e. the source IP address of the ICMP Echo reply.



```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host
# name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com       # source server
#       38.25.63.10       x.acme.com           # x client host
#
# localhost name resolution is handled within DNS itself.
#       127.0.0.1         localhost
#       ::1               localhost
#
127.0.0.1 activate.adobe.com
```

Figure 5: *hosts* file on Windows

Answer the following questions:

- What's the priority of *hosts* file and DNS cache on your Operating System?  
(Tips: Different OS may use different policies. And on Linux, you can control the priority with “/etc/nsswitch.conf”.)

#### Task 4: Explore on your own (Not required in the lab report)

We encourage you to explore DNS with *dig* on your own once you have completed this lab. Some ideas:

- Look up other types of DNS records, such as MX to find the mail server for a domain, and AAAA to find the IPv6 address of a domain.
- Google provides an alternate DNS nameserver system that you may use called “Google Public DNS”. Look it up, and follow the configuration instructions to test it out. Experiment to see if this DNS service is faster than your existing DNS arrangement.
- Reverse DNS lookups determine the domain name associated with an IP address. They are often used as a security check. Read about and perform some reverse DNS lookups.
- DNSSEC is a set of security extensions for DNS. It uses additional DNS record types to return key and signature information so that nameservers can check the authenticity of responses. Read about DNSSEC and perform some DNSSEC lookups using *dig*. You will need to add “+dnssec” to turn on the flag requesting security.