Amrita Viswa Vidyapeetham
Amrita School of Engineering
Dept of Computer Science & Engineering
19CSE301 Computer Networks Lab

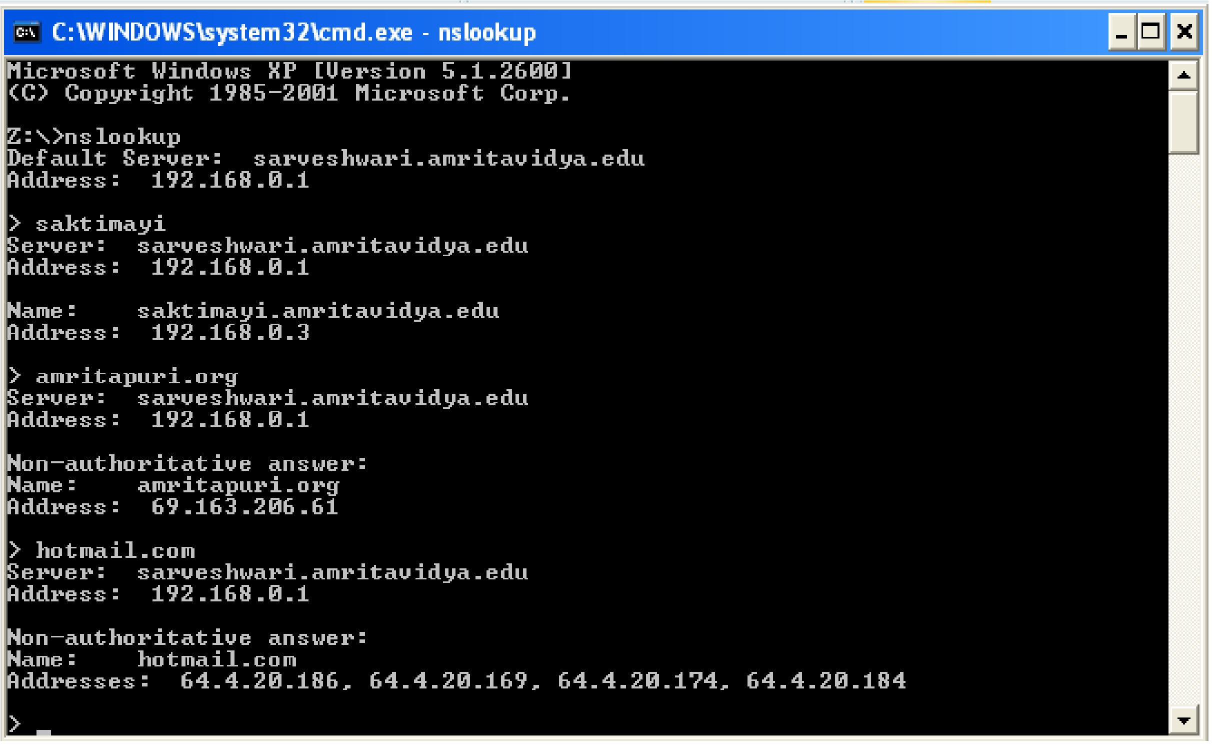
LabSheet3 - Understand how DNS works using the Wireshark

1. nslookup

In this lab, we'll make extensive use of the *nslookup* tool, which is available in most Linux/Unix and Microsoft platforms today. To run *nslookup* in Linux/Unix, you just type the *nslookup* command on the command line. To run it in Windows, open the Command Prompt and run *nslookup* on the command line.

- In it is most basic operation, *nslookup* tool allows the host running the tool to query any specified DNS server for a DNS record.
- The queried DNS server can be a root DNS server, a top-level-domain DNS server, an authoritative DNS server, or an intermediate DNS server (see the textbook for definitions of these terms).
- To accomplish this task, *nslookup* sends a DNS query to the specified DNS server, receives a DNS reply from that same DNS server, and displays the result.

LabSheet3 - Understand how DNS works using the Wireshark



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

Z:\>nslookup
Default Server:  sarveshwari.amritavidya.edu
Address:  192.168.0.1

> saktimayi
Server:  sarveshwari.amritavidya.edu
Address:  192.168.0.1

Name:    saktimayi.amritavidya.edu
Address:  192.168.0.3

> amritapuri.org
Server:  sarveshwari.amritavidya.edu
Address:  192.168.0.1

Non-authoritative answer:
Name:    amritapuri.org
Address:  69.163.206.61

> hotmail.com
Server:  sarveshwari.amritavidya.edu
Address:  192.168.0.1

Non-authoritative answer:
Name:    hotmail.com
Addresses:  64.4.20.186, 64.4.20.169, 64.4.20.174, 64.4.20.184

> _
```

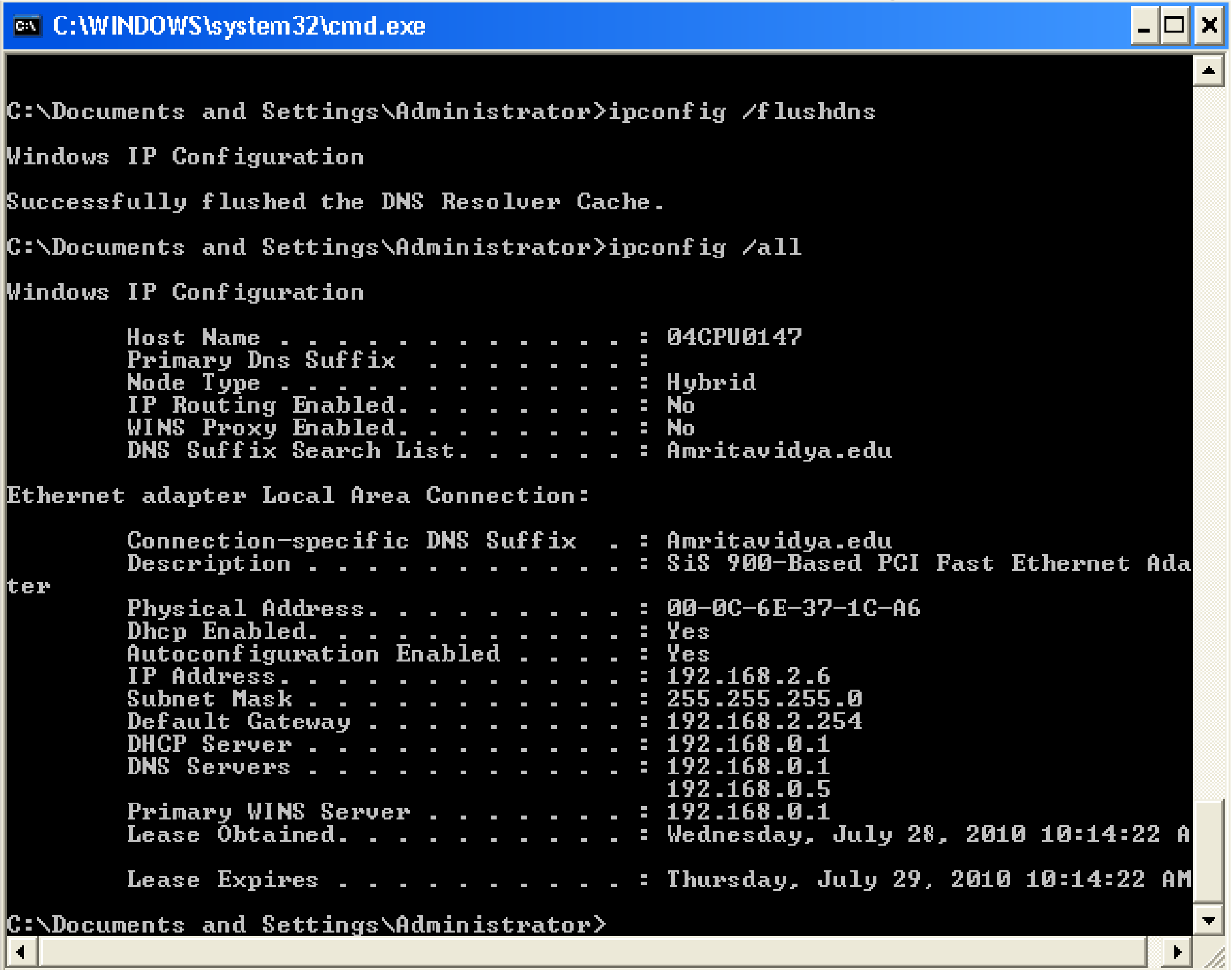
2. ipconfig

ipconfig (for Windows) and ifconfig (for Linux/Unix) are among the most useful little utilities in your host, especially for debugging network issues.
Here we'll only describe ipconfig, although the Linux/Unix ifconfig is very similar.

ipconfig can be used to show your current TCP/IP information, including your address, DNS server addresses, adapter type and so on. For example, if you want to see all this information about your host, simply enter:

```
ipconfig /all
```

into the Command Prompt, as shown in the following screenshot.



```
C:\Documents and Settings\Administrator>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Documents and Settings\Administrator>ipconfig /all

Windows IP Configuration

        Host Name . . . . . . . . . . . . : 04CPU0147
        Primary Dns Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Hybrid
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No
        DNS Suffix Search List. . . . . . : Amritavidya.edu

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : Amritavidya.edu
        Description . . . . . . . . . . . : SiS 900-Based PCI Fast Ethernet Ada
ter
        Physical Address. . . . . . . . . : 00-0C-6E-37-1C-A6
        Dhcp Enabled. . . . . . . . . . . : Yes
        Autoconfiguration Enabled . . . . : Yes
        IP Address. . . . . . . . . . . . : 192.168.2.6
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.2.254
        DHCP Server . . . . . . . . . . . : 192.168.0.1
        DNS Servers . . . . . . . . . . . : 192.168.0.1
                                            192.168.0.5
        Primary WINS Server . . . . . . . : 192.168.0.1
        Lease Obtained. . . . . . . . . . : Wednesday, July 28, 2010 10:14:22 A

        Lease Expires . . . . . . . . . . : Thursday, July 29, 2010 10:14:22 AM

C:\Documents and Settings\Administrator>
```

- *ipconfig* is also very useful for managing the DNS information stored in your host.
- We have learned that a host can cache DNS records it recently obtained.
- To see these cached records, after the prompt provide the following command: **ipconfig /displaydns**

Each entry shows the remaining Time to Live (TTL) in seconds. To clear the cache, enter

ipconfig /flushdns

Flushing the DNS cache clears all entries and reloads the entries from the hosts file.

3. Tracing DNS with Wireshark

Now that we are familiar with nslookup and ipconfig, we're ready to get down to some serious business. Let's first capture the DNS packets that are generated by ordinary Web surfing activity.

• Use ipconfig to empty the DNS cache in your host.

• Open your browser and empty your browser cache. (With Internet Explorer, go to Tools menu and select Internet Options; then in the General tab select Delete Files.)

• Open Wireshark and enter "ip.addr == your_IP_address" into the filter, where you obtain your_IP_address (the IP address for the computer on which you are running Wireshark) with ipconfig. This filter removes all packets that neither originate nor are destined to your host.

• With your browser, visit Web pages in internet.
• Stop packet capture.

1. Explain the working of the DNS protocol[DNS Request Query and DNS Response message] briefly with typed answers and answer highlighted screenshots for the above capture

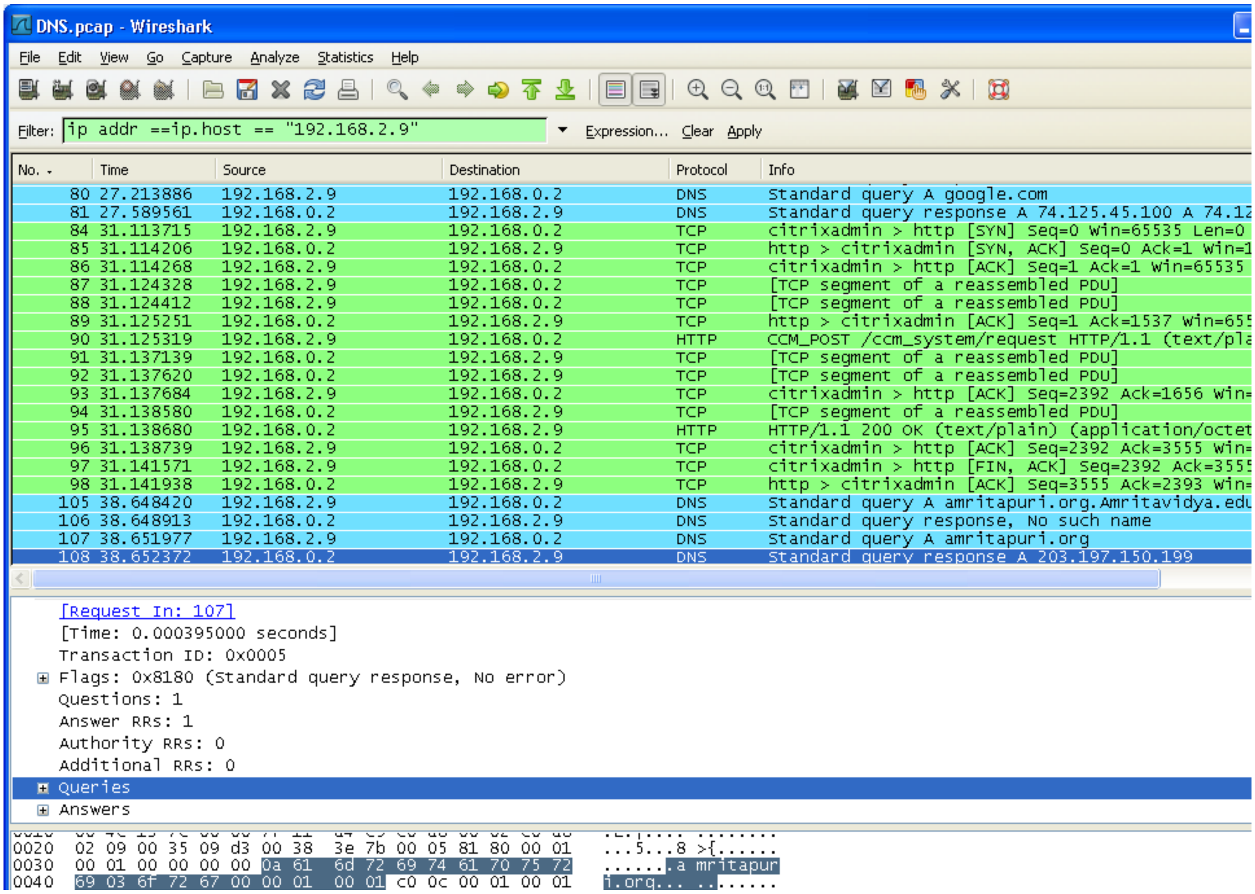
a. Locate the DNS query and response messages. Are they sent over UDP or TCP?
b. What is the destination port for the DNS query message? What is the source port of DNS response message?
c. To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same?
d. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?
e. Examine the DNS response message. How many "answers" are provided? What does each of these answers contain?
f. Before retrieving each image/object in your web page, does your host issue new DNS queries?

2. Now let's play with nslookup.

- Start packet capture.
- Do an nslookup on amritapuri.org, google.com etc.
- Stop packet capture.

You should get a trace that looks something like the following:

LabSheet3 - Understand how DNS works using the Wireshark


DNS.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Expression... Clear Apply

Filter: ip addr ==ip.host == "192.168.2.9"

```
No. -  Time       Source        Destination   Protocol  Info
 80 27.213886   192.168.2.9   192.168.0.2   DNS    Standard query A google.com
 81 27.589561   192.168.0.2   192.168.2.9   DNS    Standard query response A 74.125.45.100 A 74.12
 84 31.113715   192.168.2.9   192.168.0.2   TCP    citrixadmin > http [SYN] Seq=0 Win=65535 Len=0
 85 31.114206   192.168.0.2   192.168.2.9   TCP    http > citrixadmin [SYN, ACK] Seq=0 Ack=1 Win=1
 86 31.114268   192.168.2.9   192.168.0.2   TCP    citrixadmin > http [ACK] Seq=1 Ack=1 Win=65535
 87 31.124328   192.168.2.9   192.168.0.2   TCP    [TCP segment of a reassembled PDU]
 88 31.124412   192.168.2.9   192.168.0.2   TCP    [TCP segment of a reassembled PDU]
 89 31.125251   192.168.0.2   192.168.2.9   TCP    http > citrixadmin [ACK] Seq=1 Ack=1537 Win=655
 90 31.125319   192.168.2.9   192.168.0.2   HTTP   CCM_POST /ccm_system/request HTTP/1.1 (text/pla
 91 31.137139   192.168.0.2   192.168.2.9   TCP    [TCP segment of a reassembled PDU]
 92 31.137620   192.168.0.2   192.168.2.9   TCP    [TCP segment of a reassembled PDU]
 93 31.137684   192.168.2.9   192.168.0.2   TCP    citrixadmin > http [ACK] Seq=2392 Ack=1656 Win=
 94 31.138580   192.168.0.2   192.168.2.9   TCP    [TCP segment of a reassembled PDU]
 95 31.138680   192.168.0.2   192.168.2.9   HTTP   HTTP/1.1 200 OK (text/plain) (application/octet
 96 31.138739   192.168.2.9   192.168.0.2   TCP    citrixadmin > http [ACK] Seq=2392 Ack=3555 Win=
 97 31.141571   192.168.2.9   192.168.0.2   TCP    citrixadmin > http [FIN, ACK] Seq=2392 Ack=3555
 98 31.141938   192.168.0.2   192.168.2.9   TCP    http > citrixadmin [ACK] Seq=3555 Ack=2393 Win=
105 38.648420   192.168.2.9   192.168.0.2   DNS    Standard query A amritapuri.org.Amritavidya.edu
106 38.648913   192.168.0.2   192.168.2.9   DNS    Standard query response, No such name
107 38.651977   192.168.2.9   192.168.0.2   DNS    Standard query A amritapuri.org
108 38.652372   192.168.0.2   192.168.2.9   DNS    Standard query response A 203.197.150.199
```

```
    [Request In: 107]
    [Time: 0.000395000 seconds]
    Transaction ID: 0x0005
  ⊞ Flags: 0x8180 (Standard query response, No error)
    Questions: 1
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  ⊞ Queries
  ⊞ Answers
```

```
0020   02 09 00 35 09 d3 00 38  3e 7b 00 05 81 80 00 01   ...5...8 >{......
0030   00 01 00 00 00 00 0a 61  6d 72 69 74 61 70 75 72   .......a mritapur
0040   69 03 6f 72 67 00 00 01  00 01 c0 0c 00 01 00 01   i.org... ........
```

We see from the above screenshot that *nslookup* actually sent two/three DNS queries and received two/three DNS responses.

For the purpose of this assignment, in answering the following questions ignore the first one/two sets of queries/responses, as they are specific to *nslookup* and are not normally generated by standard Internet applications.

You should instead focus on the last query and response messages. Again, answer the following questions for this capture of frames.

Amrita Viswa Vidyapeetham
Amrita School of Engineering
Dept of Computer Science & Engineering
19CSE301 Computer Networks Lab

LabSheet3 - Understand how DNS works using the Wireshark

a. What is the destination port for the DNS query message? What is the source port of DNS response message?
b. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
c. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?
d. Examine the DNS response message. How many "answers" are provided? What does each of these answers contain?