LAB EXERCISE - 4

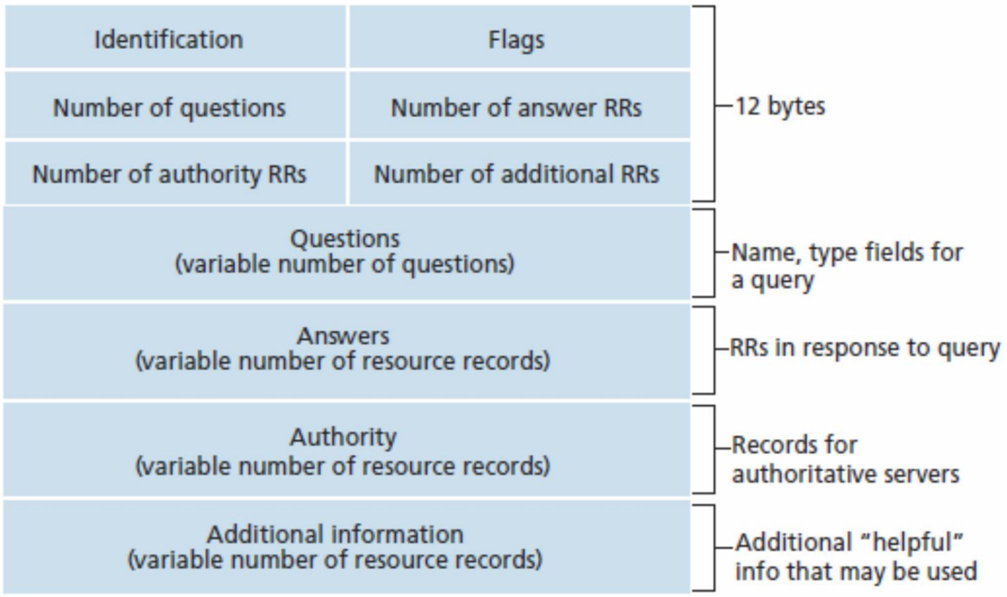
Identification
Flags
Number of questions
Number of answer RRs
Number of authority RRs
Number of additional RRs
12 bytes
Questions (variable number of questions)
Name, type fields for a query
Answers (variable number of resource records)
RRs in response to query
Authority (variable number of resource records)
Records for authoritative servers
Additional information (variable number of resource records)
Additional "helpful" info that may be used

Figure 2.23 ♦ DNS message format

A resource record is a four-tuple that contains the following fields: (Name, Value, Type, TTL) TTL is the time to live of the resource record; it determines when a resource should be removed from a cache. The meaning of Name and Value depend on Type:

- If Type=A, then Name is a hostname and Value is the IP address for the hostname. Thus, a Type A record provides the standard hostname-to-IP address mapping. As an example, (relay1.bar.foo.com, 145.37.93.126, A) is a Type A record.
- If Type=NS, then Name is a domain (such as foo.com) and Value is the hostname of an authoritative DNS server that knows how to obtain the IP addresses for hosts in the domain. This record is used to route DNS queries further along in the query chain. As an example, (foo.com, dns.foo.com, NS) is a Type NS record.
- If Type=CNAME, then Value is a canonical hostname for the alias hostname Name. This record can provide querying hosts the canonical name for a hostname. As an example, (foo.com, relay1.bar.foo.com, CNAME) is a CNAME record.
- If Type=MX, then Value is the canonical name of a mail server that has an alias hostname Name. As an example, (foo.com, mail.bar.foo.com, MX) is an MX record. MX records allow the hostnames of mail servers to have simple aliases. Note that by using the MX record, a company can have the same aliased name for its mail server and for one of its other servers (such as its Web server). To obtain the canonical name for the mail server, a DNS client would query for an MX record; to obtain the canonical name for the other server, the DNS client would query for the CNAME record.

nslookup

To run it in Ubuntu, open the terminal and run *nslookup* on the command line.

In its 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. 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.

Do the following commands

```
nslookup www.mit.edu
nslookup –type=NS mit.edu
nslookup www.mit.edu ns-137.akam.net  (requesting for translation from authoritative server for
                                        www.mit.edu  which is ns-137.akam.net)
```

general syntax of *nslookup* commands.

The syntax is:
nslookup –option1 –option2 host-to-find dns-server

In general, *nslookup* can be run with zero, one, two or more options.

Ipconfig

Ipconfig in windows can be used to show your current network information, including your ipaddress, 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

Use $nmcli dev show | grep DNS to see the dns server ipaddress in ubuntu

Tracing DNS with Wireshark

using *nslookup* and *ifconfig* capture the DNS packets that are generated by ordinary Websurfing

1. Open your browser and empty your browser cache.
2. 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 *ifconfig*. This filter removes all packets that neither originate nor are destined to your host.

3. Start packet capture in Wireshark.

4. With your browser, visit the Web page: http://www.ietf.org

6. Stop packet capture.

nslookup www.mit.edu

1. Start packet capture.
2. Do an *nslookup* on www.mit.edu
3. Stop packet capture

Note: We see for the above experiment that *nslookup* actually sent three DNS queries and received three DNS responses. For the purpose of this assignment, in answering the following questions, ignore the first 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.

Now repeat the previous experiment, but instead issue the command:
nslookup –type=NS mit.edu