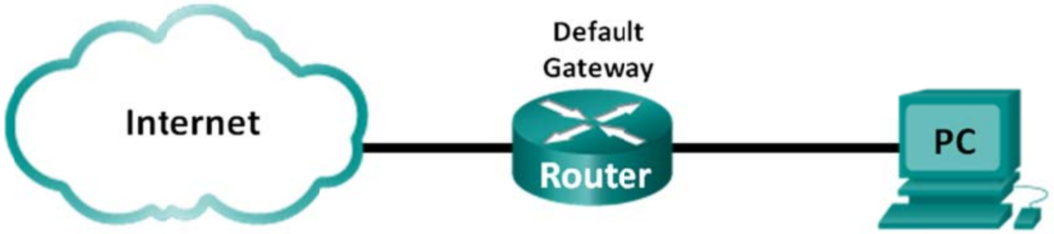
# Lab - Using Wireshark to Examine a UDP DNS Capture

## Topology



## Objectives

**Part 1: Record a PC's IP Configuration Information**

**Part 2: Use Wireshark to Capture DNS Queries and Responses**

**Part 3: Analyze Captured DNS or UDP Packets**

## Background / Scenario

If you have ever used the Internet, you have used the Domain Name System (DNS). DNS is a distributed network of servers that translates user-friendly domain names like www.google.com to an IP address. When you type a website URL into your browser, your PC performs a DNS query to the DNS server's IP address. Your PC's DNS server query and the DNS server's response make use of the User Datagram Protocol (UDP) as the transport layer protocol. UDP is connectionless and does not require a session setup as does TCP. DNS queries and responses are very small and do not require the overhead of TCP.

In this lab, you will communicate with a DNS server by sending a DNS query using the UDP transport protocol. You will use Wireshark to examine the DNS query and response exchanges with the name server.

**Note**: This lab cannot be completed using Netlab. This lab assumes that you have Internet access.

## Required Resources

1 PC (Windows 7, Vista, or XP with a command prompt access, Internet access, and Wireshark installed)

## Part 1: Record a PC's IP Configuration Information

In Part 1, you will use the **ipconfig /all** command on your local PC to find and record the MAC and IP addresses of your PC's network interface card (NIC), the IP address of the specified default gateway, and the DNS server IP address specified for the PC. Record this information in the table provided. The information will be used in the following parts of this lab with packet analysis.

| | |
|---|---|
| IP address | |
| MAC address | |
| Default gateway IP address | |
| DNS server IP address | |

## Part 2: Use Wireshark to Capture DNS Queries and Responses

In Part 2, you will set up Wireshark to capture DNS query and response packets to demonstrate the use of UDP transport protocol while communicating with a DNS server.

a. Click the Windows **Start** button and navigate to the Wireshark program.

   **Note**: If Wireshark is not yet installed, it can be downloaded at http://www.wireshark.org/download.html.

b. Select an interface for Wireshark for capturing packets. Use the **Interface List** to choose the interface that is associated with the recorded PC's IP and Media Access Control (MAC) addresses in Part 1.

c. After selecting the desired interface, click **Start** to capture the packets.

d. Open a web browser and type **www.google.com**. Press Enter to continue.

e. Click **Stop** to stop the Wireshark capture when you see Google's home page.

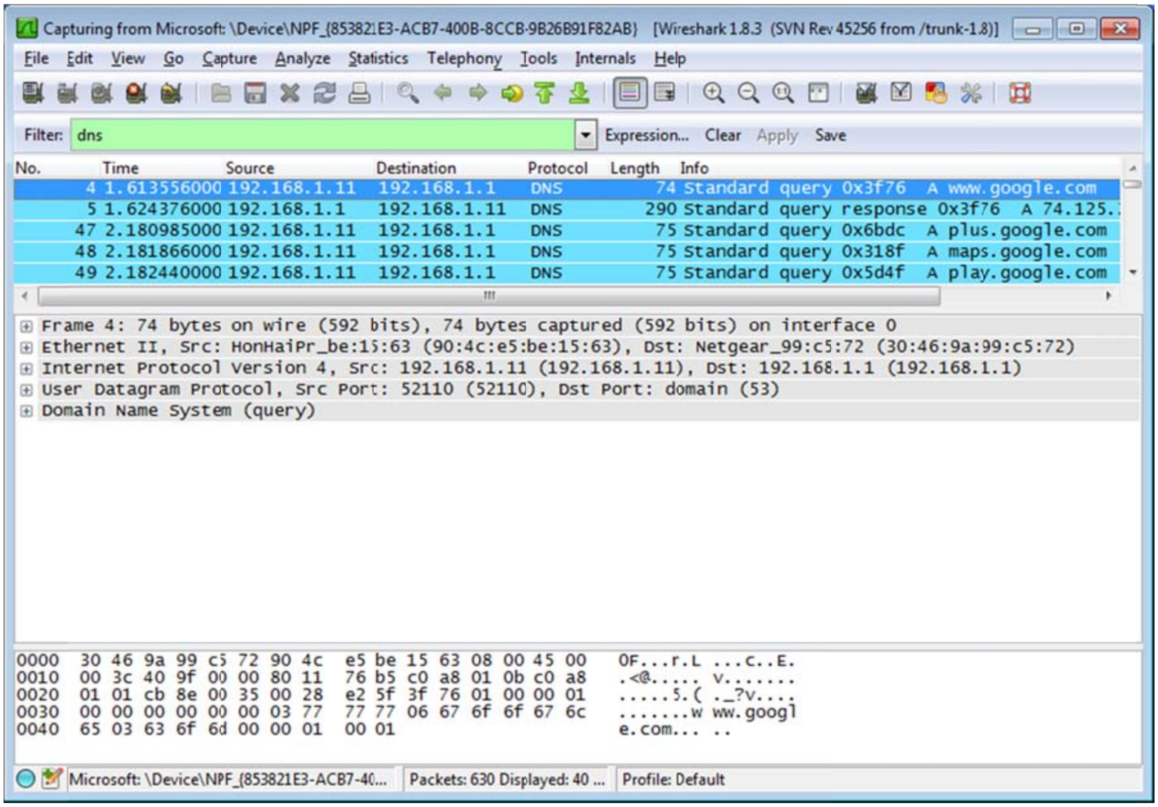
## Part 3: Analyze Captured DNS or UDP Packets

In Part 3, you will examine the UDP packets that were generated when communicating with a DNS server for the IP addresses for www.google.com.

### Step 1: Filter DNS packets.

a. In the Wireshark main window, type **dns** in the entry area of the **Filter** toolbar. Click **Apply** or press Enter.

   **Note**: If you do not see any results after the DNS filter was applied, close the web browser and in the command prompt window, type **ipconfig /flushdns** to remove all previous DNS results. Restart the Wireshark capture and repeat the instructions in Part 2b –2e. If this does not resolve the issue, in the command prompt window, you can type **nslookup www.google.com** as an alternative to the web browser.


Capturing from Microsoft: \Device\NPF_{853821E3-ACB7-400B-8CCB-9B26B91F82AB} [Wireshark 1.8.3 (SVN Rev 45256 from /trunk-1.8)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

Filter: dns

```
No.     Time          Source         Destination    Protocol  Length  Info
      4 1.613556000 192.168.1.11   192.168.1.1     DNS           74 Standard query 0x3f76  A www.google.com
      5 1.624376000 192.168.1.1    192.168.1.11    DNS          290 Standard query response 0x3f76  A 74.125.
     47 2.180985000 192.168.1.11   192.168.1.1     DNS           75 Standard query 0x6bdc  A plus.google.com
     48 2.181866000 192.168.1.11   192.168.1.1     DNS           75 Standard query 0x318f  A maps.google.com
     49 2.182440000 192.168.1.11   192.168.1.1     DNS           75 Standard query 0x5d4f  A play.google.com
```

```
⊞ Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
⊞ Ethernet II, Src: HonHaiPr_be:15:63 (90:4c:e5:be:15:63), Dst: Netgear_99:c5:72 (30:46:9a:99:c5:72)
⊞ Internet Protocol Version 4, Src: 192.168.1.11 (192.168.1.11), Dst: 192.168.1.1 (192.168.1.1)
⊞ User Datagram Protocol, Src Port: 52110 (52110), Dst Port: domain (53)
⊞ Domain Name System (query)
```

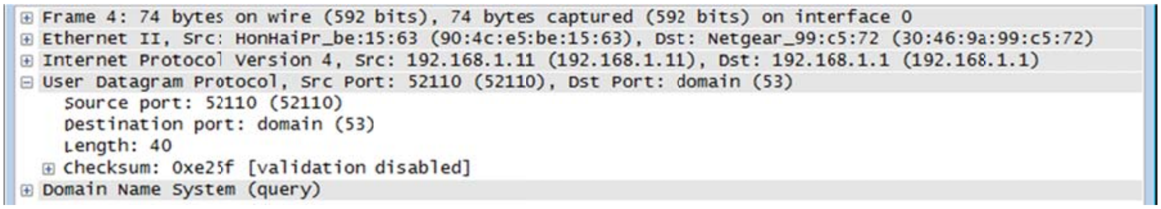
```
0000  30 46 9a 99 c5 72 90 4c  e5 be 15 63 08 00 45 00   0F...r.L ...c..E.
0010  00 3c 40 9f 00 00 80 11  76 b5 c0 a8 01 0b c0 a8   .<@..... v.......
0020  01 01 cb 8e 00 35 00 28  e2 5f 3f 76 01 00 00 01   .....5.( ._?v....
0030  00 00 00 00 00 00 03 77  77 77 06 67 6f 6f 67 6c   .......w ww.googl
0040  65 03 63 6f 6d 00 00 01  00 01                     e.com... ..
```

Microsoft: \Device\NPF_{853821E3-ACB7-40...
Packets: 630 Displayed: 40 ...
Profile: Default

b. In the packet list pane (top section) of the main window, locate the packet that includes "standard query" and "A www.google.com". See frame 4 as an example.

### Step 2: Examine UDP segment using DNS query.

Examine UDP by using a DNS query for www.google.com as captured by Wireshark. In this example, Wireshark capture frame 4 in the packet list pane is selected for analysis. The protocols in this query are displayed in the packet details pane (middle section) of the main window. The protocol entries are highlighted in gray.



```
⊞ Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
⊞ Ethernet II, Src: HonHaiPr_be:15:63 (90:4c:e5:be:15:63), Dst: Netgear_99:c5:72 (30:46:9a:99:c5:72)
⊞ Internet Protocol Version 4, Src: 192.168.1.11 (192.168.1.11), Dst: 192.168.1.1 (192.168.1.1)
⊟ User Datagram Protocol, Src Port: 52110 (52110), Dst Port: domain (53)
    Source port: 52110 (52110)
    Destination port: domain (53)
    Length: 40
  ⊞ Checksum: 0xe25f [validation disabled]
⊞ Domain Name System (query)
```

a. In the packet details pane, frame 4 had 74 bytes of data on the wire as displayed on the first line. This is the number of bytes to send a DNS query to a name server requesting the IP addresses of www.google.com.

b. The Ethernet II line displays the source and destination MAC addresses. The source MAC address is from your local PC because your local PC originated the DNS query. The destination MAC address is from the default gateway, because this is the last stop before this query exits the local network.

   Is the source MAC address the same as recorded from Part 1 for the local PC?

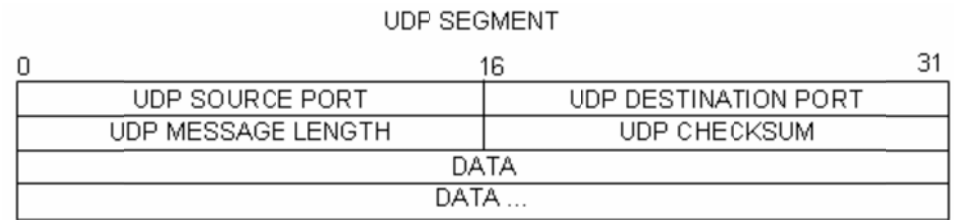
c. In the Internet Protocol Version 4 line, the IP packet Wireshark capture indicates that the source IP address of this DNS query is 192.168.1.11, and the destination IP address is 192.168.1.1. In this example, the destination address is the default gateway. The router is the default gateway in this network.

   Can you pair up the IP and MAC addresses for the source and destination devices?

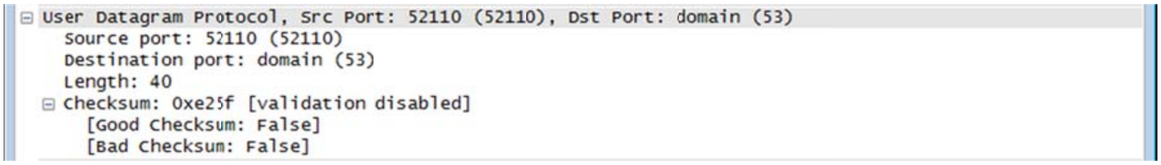
| Device | IP Address | MAC Address |
|---|---|---|
| Local PC | | |
| Default Gateway | | |

   The IP packet and header encapsulates the UDP segment. The UDP segment contains the DNS query as the data.

d. A UDP header only has four fields: source port, destination port, length, and checksum. Each field in UDP header is only 16 bits as depicted below.

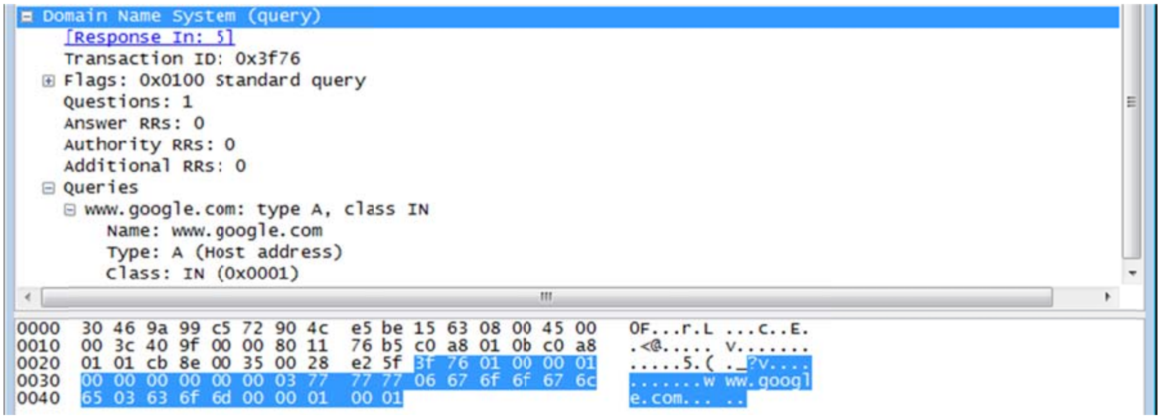


   Expand the User Datagram Protocol in the packet details pane by clicking the plus (+) sign. Notice that there are only four fields. The source port number in this example is 52110. The source port was randomly generated by the local PC using port numbers that are not reserved. The destination port is 53. Port 53 is a well-known port reserved for use with DNS. DNS servers listen on port 53 for DNS queries from clients.



```
⊟ User Datagram Protocol, Src Port: 52110 (52110), Dst Port: domain (53)
    Source port: 52110 (52110)
    Destination port: domain (53)
    Length: 40
  ⊟ Checksum: 0xe25f [validation disabled]
      [Good Checksum: False]
      [Bad Checksum: False]
```

In this example, the length of this UDP segment is 40 bytes. Out of 40 bytes, 8 bytes are used as header. The other 32 bytes are used by DNS query data. The 32 bytes of DNS query data is highlighted in the following illustration in the packet bytes pane (lower section) of the Wireshark main window.



```
⊟ Domain Name System (query)
    [Response In: 5]
    Transaction ID: 0x3f76
  ⊞ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ⊟ Queries
    ⊟ www.google.com: type A, class IN
        Name: www.google.com
        Type: A (Host address)
        Class: IN (0x0001)

0000  30 46 9a 99 c5 72 90 4c  e5 be 15 63 08 00 45 00   0F...r.L ...c..E.
0010  00 3c 40 9f 00 00 80 11  76 b5 c0 a8 01 0b c0 a8   .<@..... v.......
0020  01 01 cb 8e 00 35 00 28  e2 5f 3f 76 01 00 00 01   .....5.( ._?v....
0030  00 00 00 00 00 00 03 77  77 77 06 67 6f 6f 67 6c   .......w ww.googl
0040  65 03 63 6f 6d 00 00 01  00 01                     e.com... ..
```

The checksum is used to determine the integrity of the packet after it has traversed the Internet.

The UDP header has low overhead because UDP does not have fields that are associated with three-way handshake in TCP. Any data transfer reliability issues that occur must be handled by the application layer.

Record your Wireshark results in the table below:

| | |
|---|---|
| **Frame Size** | |
| **Source MAC address** | |
| **Destination MAC address** | |
| **Source IP address** | |
| **Destination IP address** | |
| **Source Port** | |
| **Destination Port** | |

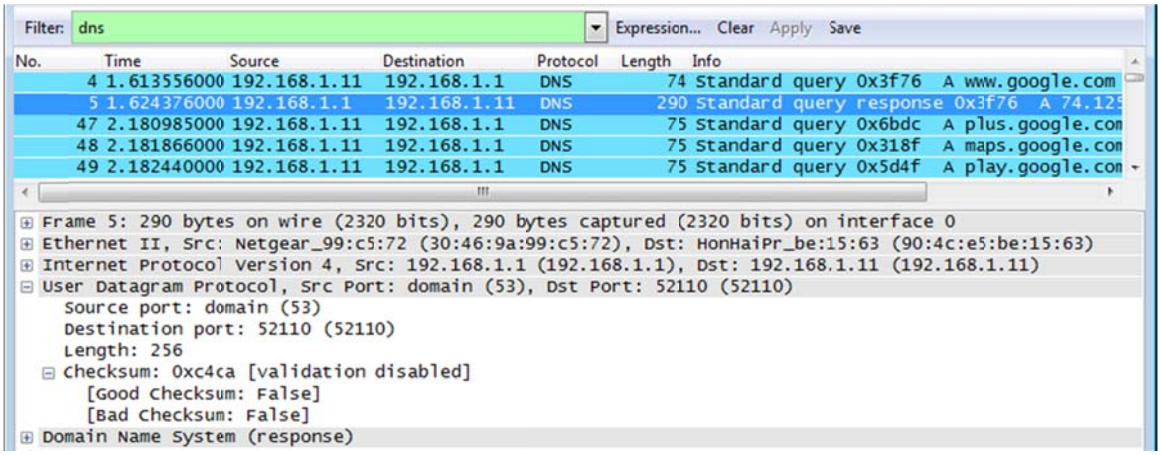
Is the source IP address the same as the local PC's IP address recorded in Part 1?

Is the destination IP address the same as the default gateway noted in Part 1?

### Step 3: Examine UDP using DNS response.

In this step, you will examine the DNS response packet and verify that DNS response packet also uses UDP.

a. In this example, frame 5 is the corresponding DNS response packet. Notice the number of bytes on the wire is 290 bytes. It is a larger packet as compared to the DNS query packet.



```
No.    Time        Source        Destination   Protocol Length Info
     4 1.613556000 192.168.1.11  192.168.1.1   DNS          74 Standard query 0x3f76  A www.google.com
     5 1.624376000 192.168.1.1   192.168.1.11  DNS         290 Standard query response 0x3f76  A 74.125
    47 2.180985000 192.168.1.11  192.168.1.1   DNS          75 Standard query 0x6bdc  A plus.google.com
    48 2.181866000 192.168.1.11  192.168.1.1   DNS          75 Standard query 0x318f  A maps.google.com
    49 2.182440000 192.168.1.11  192.168.1.1   DNS          75 Standard query 0x5d4f  A play.google.com

⊞ Frame 5: 290 bytes on wire (2320 bits), 290 bytes captured (2320 bits) on interface 0
⊞ Ethernet II, Src: Netgear_99:c5:72 (30:46:9a:99:c5:72), Dst: HonHaiPr_be:15:63 (90:4c:e5:be:15:63)
⊞ Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 192.168.1.11 (192.168.1.11)
⊟ User Datagram Protocol, Src Port: domain (53), Dst Port: 52110 (52110)
    Source port: domain (53)
    Destination port: 52110 (52110)
    Length: 256
  ⊟ Checksum: 0xc4ca [validation disabled]
      [Good Checksum: False]
      [Bad Checksum: False]
⊞ Domain Name System (response)
```

b. In the Ethernet II frame for the DNS response, from what device is the source MAC address and what device is the destination MAC address?

c. Notice the source and destination IP addresses in the IP packet. What is the destination IP address? What is the source IP address?

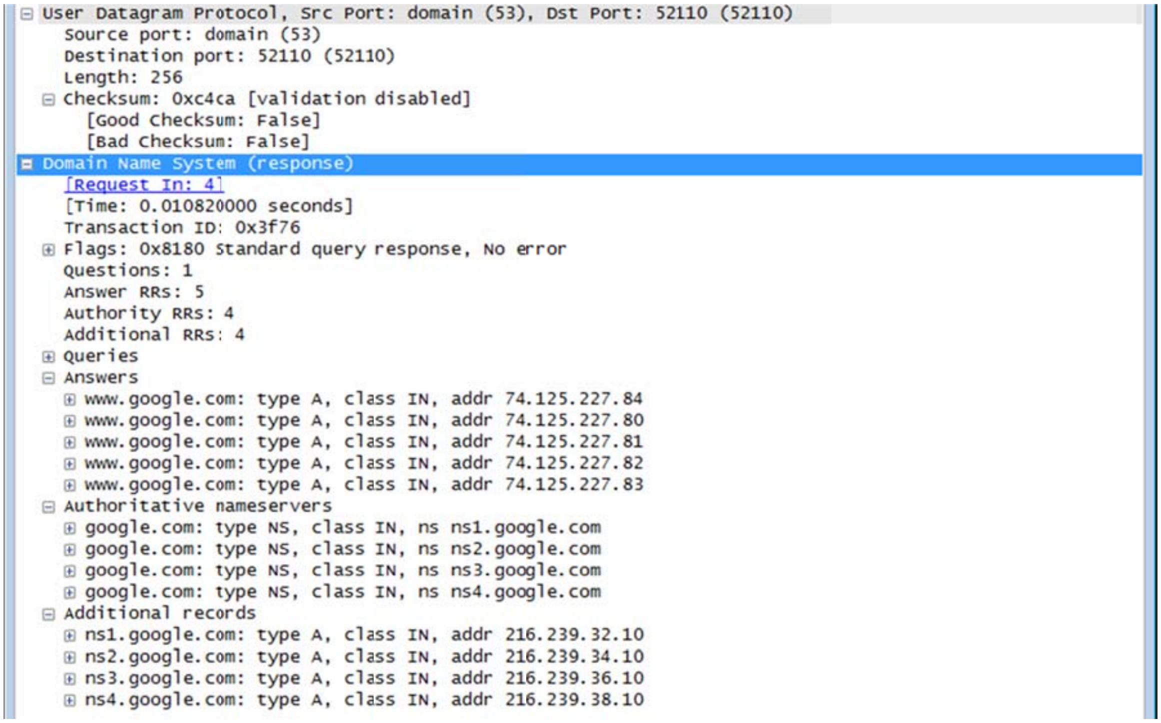
Destination IP address: Source IP address:

What happened to the roles of source and destination for the local host and default gateway?

d. In the UDP segment, the role of the port numbers has also reversed. The destination port number is 52110. Port number 52110 is the same port that was generated by the local PC when the DNS query was sent to the DNS server. Your local PC listens for a DNS response on this port.

The source port number is 53. The DNS server listens for a DNS query on port 53 and then sends a DNS response with a source port number of 53 back to originator of the DNS query.

When the DNS response is expanded, notice the resolved IP addresses for www.google.com in the **Answers** section.



```
⊟ User Datagram Protocol, Src Port: domain (53), Dst Port: 52110 (52110)
    Source port: domain (53)
    Destination port: 52110 (52110)
    Length: 256
  ⊟ Checksum: 0xc4ca [validation disabled]
      [Good Checksum: False]
      [Bad Checksum: False]
⊟ Domain Name System (response)
    [Request In: 4]
    [Time: 0.010820000 seconds]
    Transaction ID: 0x3f76
  ⊞ Flags: 0x8180 Standard query response, No error
    Questions: 1
    Answer RRs: 5
    Authority RRs: 4
    Additional RRs: 4
  ⊞ Queries
  ⊟ Answers
    ⊞ www.google.com: type A, class IN, addr 74.125.227.84
    ⊞ www.google.com: type A, class IN, addr 74.125.227.80
    ⊞ www.google.com: type A, class IN, addr 74.125.227.81
    ⊞ www.google.com: type A, class IN, addr 74.125.227.82
    ⊞ www.google.com: type A, class IN, addr 74.125.227.83
  ⊟ Authoritative nameservers
    ⊞ google.com: type NS, class IN, ns ns1.google.com
    ⊞ google.com: type NS, class IN, ns ns2.google.com
    ⊞ google.com: type NS, class IN, ns ns3.google.com
    ⊞ google.com: type NS, class IN, ns ns4.google.com
  ⊟ Additional records
    ⊞ ns1.google.com: type A, class IN, addr 216.239.32.10
    ⊞ ns2.google.com: type A, class IN, addr 216.239.34.10
    ⊞ ns3.google.com: type A, class IN, addr 216.239.36.10
    ⊞ ns4.google.com: type A, class IN, addr 216.239.38.10
```

## Reflection

What are the benefits of using UDP instead of TCP as a transport protocol for DNS?