DNS (Domain Name System)

- **Definition**: DNS is a protocol used daily to make it easier for humans to access resources over networks like the Internet.
- **Function**: It allows the use of domain names (e.g., `youtube.com`, `google.com`) instead of requiring users to enter IP addresses.
- **CCNA Relevance**: Part of exam topic 4.3 ("Explain the role of DHCP and DNS within the network").
- **CCNA Focus**: You do not need to know the detailed operations of DNS, only its basic purpose.

Lecture Outline

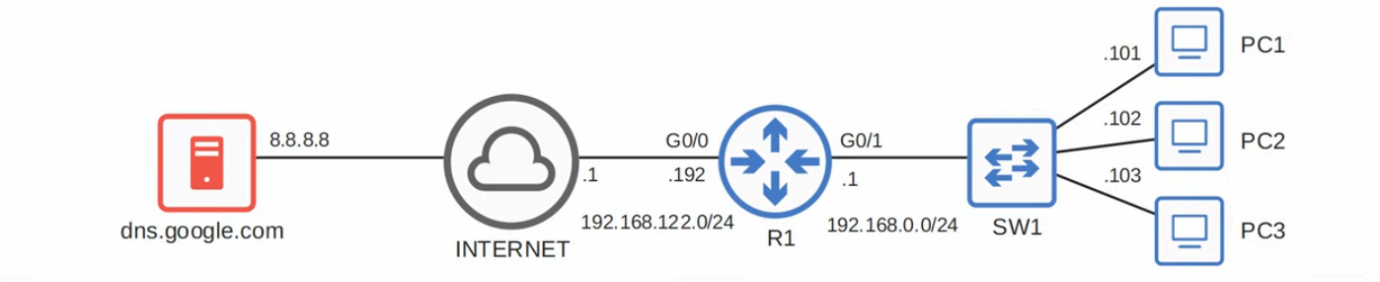
- **Purpose of DNS**: Further explanation of its role.
- **Basic Functions of DNS**: A quick overview of how it works.
- **Configuration in Cisco IOS**: How to configure DNS on Cisco routers.
- **Bonus**: A practice question from Boson Software's ExSim for CCNA is available at the end of the video.

Purpose of DNS

- **Role**: DNS is used to resolve (convert) human-readable names (e.g., `google.com`) to IP addresses.
- **Reasoning**: Machines use addresses (IPv4/IPv6), but names are easier for humans to remember.
- **Process**: When you type a name into a browser, your device asks a DNS server for the corresponding IP address.
- **Configuration**: The DNS server(s) a device uses can be manually configured or learned via DHCP (Dynamic Host Configuration Protocol).

Basic Functions of DNS

- **Network Topology**:

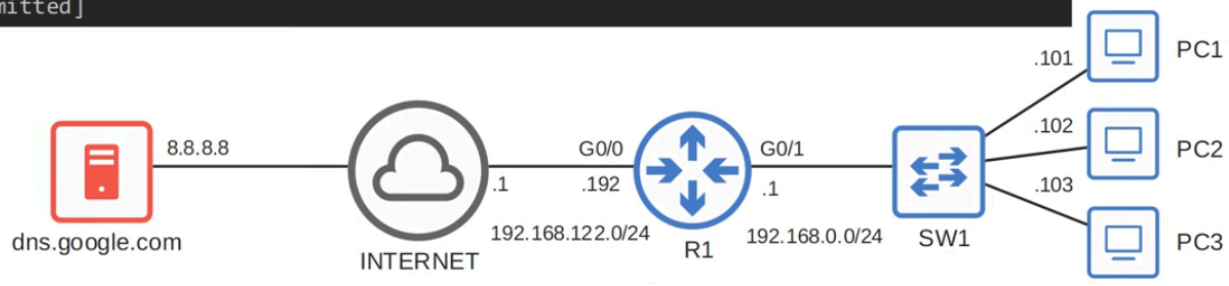


  - Three PCs connected to R1 via SW1.
  - R1 is connected to the Internet.
  - Google's DNS server (IP `8.8.8.8`) is located across the Internet.

Windows CLI: Viewing DNS Configuration

```
C:\Users\user>ipconfig /all

[output omitted]

Ethernet adapter ローカル エリア接続：

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) 82579LM Gigabit Network Connection
   Physical Address. . . . . . . . . : 78-2B-CB-AC-08-67
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 192.168.0.101(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
   DNS Servers . . . . . . . . . . . : 8.8.8.8
   NetBIOS over Tcpip. . . . . . . . : Enabled

[output omitted]
```

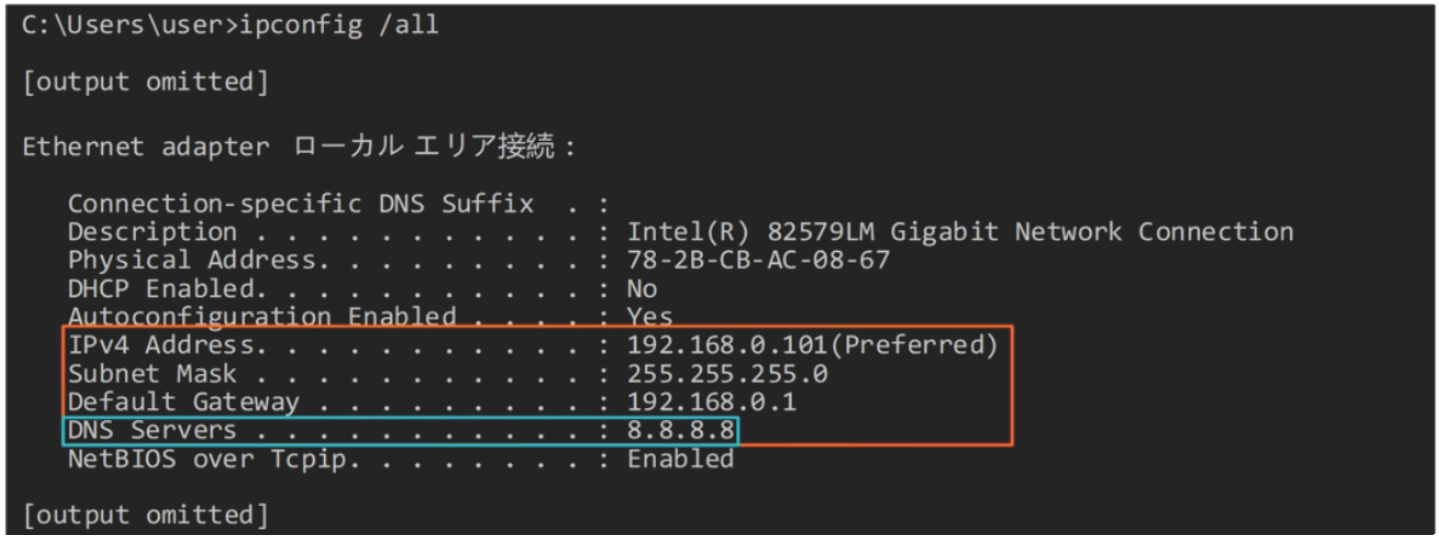
1.10 Verify IP parameters for Client OS (Windows, Mac OS, Linux)



- **Verification Command**: Use the `IPCONFIG /ALL` command in the Windows command prompt to display various info for a PC, including its configured DNS server.

- **Exam Relevance**: The objective to "verify IP parameters for client operating systems" is a CCNA exam topic.

Example:

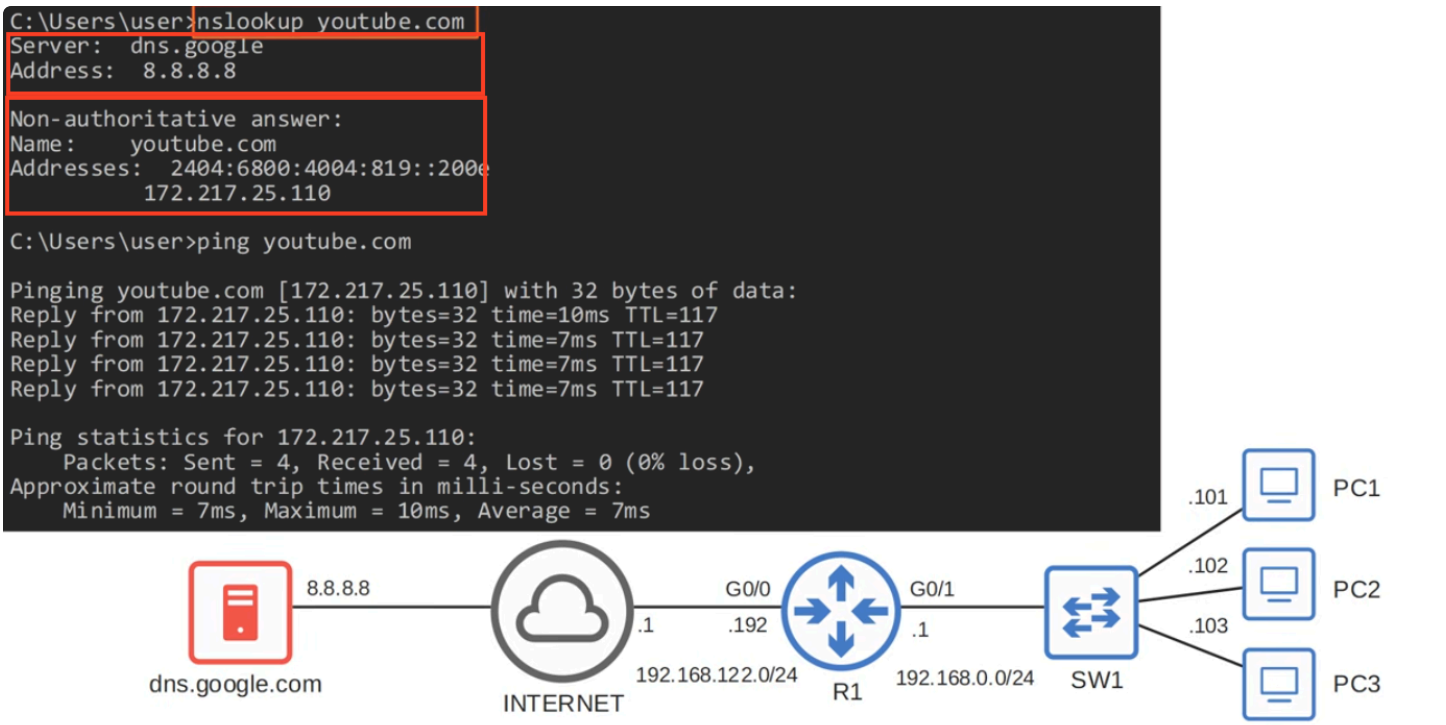


- The professor runs `IPCONFIG /ALL` on PC1 (a Windows PC).
- The output shows that PC1 is configured to use Google's DNS server at `8.8.8.8`.

```
C:\> IPCONFIG /ALL
```

- **Note**: The standard `IPCONFIG` command (without the `/ALL` option) does not display the DNS server information.
- **Configuration Context**: End hosts usually automatically learn which DNS server to use via DHCP. For this demonstration, the DNS server `8.8.8.8` was manually configured on PC1.

Demonstrating DNS: NSLOOKUP Command



- **Command Usage**: Use the `NSLOOKUP` command (stands for "Name Server Lookup") to tell the device to ask its DNS server for the IP address of a specified name.

Example:

- The professor runs `NSLOOKUP youtube.com` on PC1.
- PC1 is using Google's DNS server at `8.8.8.8`.
- The DNS server responds with:
  - YouTube's IPv4 address: `172.217.25.110`
  - YouTube's IPv6 address: (also displayed in the output)

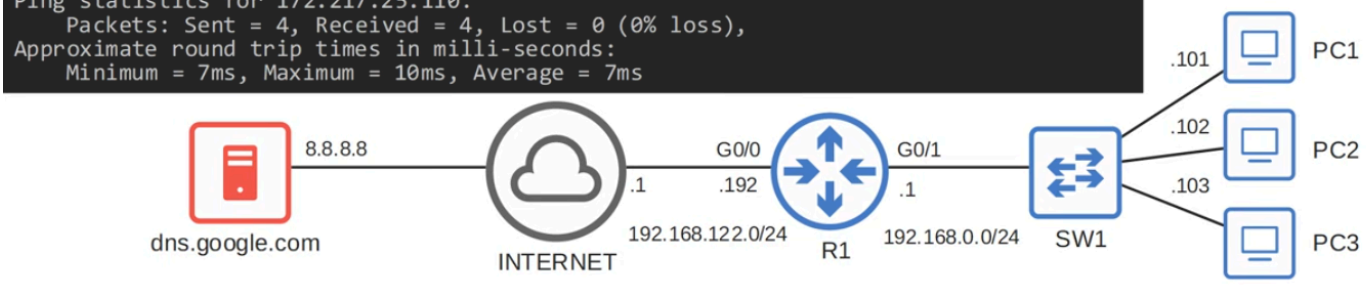
```
C:\> NSLOOKUP youtube.com
```

Demonstrating DNS: Ping with a Name

- **Ping with Name**: You can ping a name directly instead of an IP address. The device will convert the name to an IP address before sending the ping.
- **Automatic DNS Resolution**: You do not need to use `NSLOOKUP` before a ping. If the device doesn't know the correct IP address, it will automatically ask the DNS server.

Example:

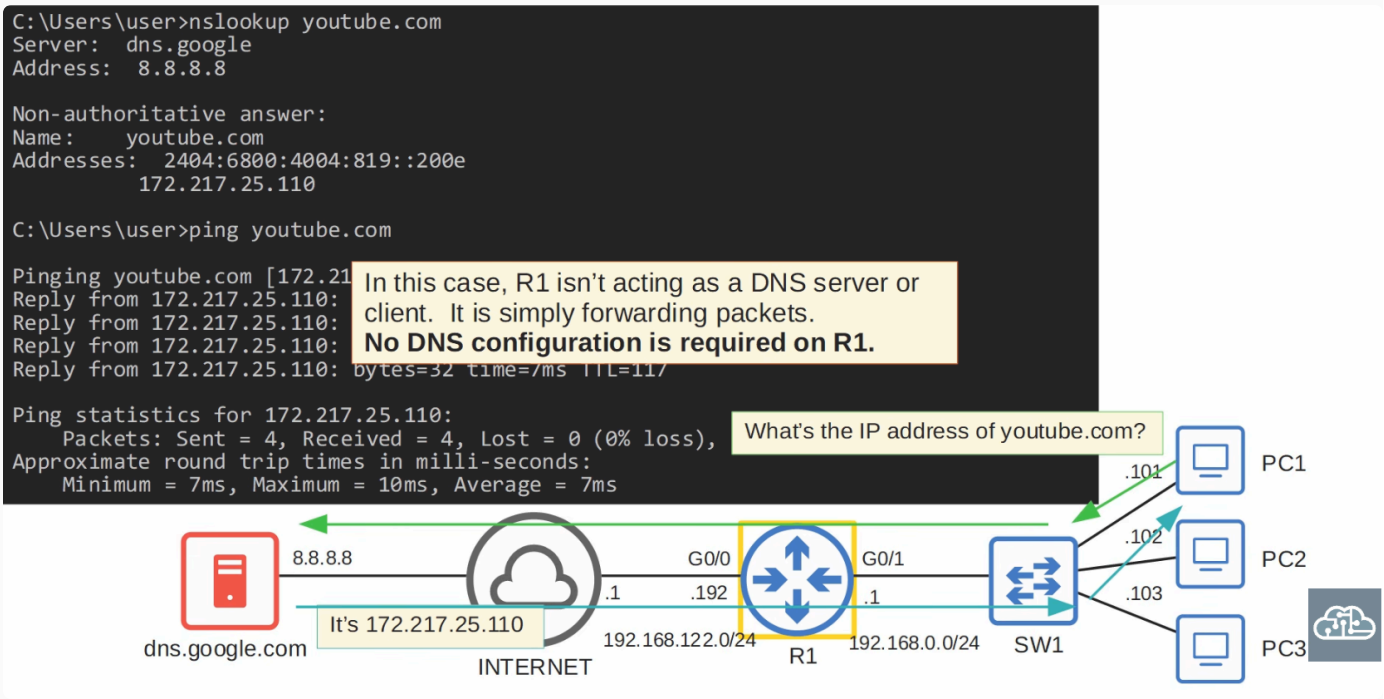
```
C:\Users\user>nslookup youtube.com
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    youtube.com
Addresses:  2404:6800:4004:819::200e
          172.217.25.110

C:\Users\user>ping youtube.com

Pinging youtube.com [172.217.25.110] with 32 bytes of data:
Reply from 172.217.25.110: bytes=32 time=10ms TTL=117
Reply from 172.217.25.110: bytes=32 time=7ms TTL=117
Reply from 172.217.25.110: bytes=32 time=7ms TTL=117
Reply from 172.217.25.110: bytes=32 time=7ms TTL=117

Ping statistics for 172.217.25.110:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 7ms, Maximum = 10ms, Average = 7ms
```



- The professor issues a `ping youtube.com` command on PC1.
- The name `youtube.com` is converted to an IP address, and PC1 successfully pings that IP address.

Visual Demonstration of DNS Query

- **Process**:



1. PC1 sends a DNS query message to its configured DNS server (`8.8.8.8`) to learn the IP address of `youtube.com`.
2. The DNS server replies, telling PC1 that the IP address is `172.217.25.110`.

- **Role of R1**:
  - In this exchange, R1 is **not** acting as a DNS server or client.
  - R1 is simply forwarding packets.
  - **Important Point**: No DNS configuration is required on R1 for the PCs to use DNS. Often, there is no need to configure DNS on routers; they just need to route network traffic.
  - However, a Cisco router **can** act as a DNS server and client, which will be demonstrated later in the video.

Wireshark Capture Analysis: NSLOOKUP Command

- **Traffic Overview**: The capture of the `NSLOOKUP` command shows four messages.

Message 1: First Query

No.	Time	Source	Destination	Protocol	Length	Info
1087	08:55:44.458619	192.168.0.101	8.8.8.8	DNS	71	Standard query 0x0002 A youtube.com
1088	08:55:44.500043	8.8.8.8	192.168.0.101	DNS	87	Standard query response 0x0002 A youtube.com A 172.217.25.110
1089	08:55:44.508888	192.168.0.101	8.8.8.8	DNS	71	Standard query 0x0003 AAAA youtube.com
1115	08:55:44.641775	8.8.8.8	192.168.0.101	DNS	99	Standard query response 0x0003 AAAA youtube.com AAAA 2404:6800:4004:819::200e

```
> Frame 1087: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NPF_{9956EC07-3774-4B11-970D-C8233E7CD172}, id 0
> Ethernet II, Src: Dell_ac:08:67 (78:2b:cb:ac:08:67), Dst: Tp-LinkT_dd:a8:e4 (98:da:c4:dd:a8:e4)
> Internet Protocol Version 4, Src: 192.168.0.101, Dst: 8.8.8.8
> User Datagram Protocol, Src Port: 49286, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0x0002
  v Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v youtube.com: type A, class IN
        Name: youtube.com
        [Name Length: 11]
        [Label Count: 2]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
    [Response In: 1088]
```

- **Source**: `192.168.0.101` (PC1)
- **Destination**: `8.8.8.8` (Google's DNS server)
- **Info**: Standard Query
- **Query Details**: `A youtube.com`

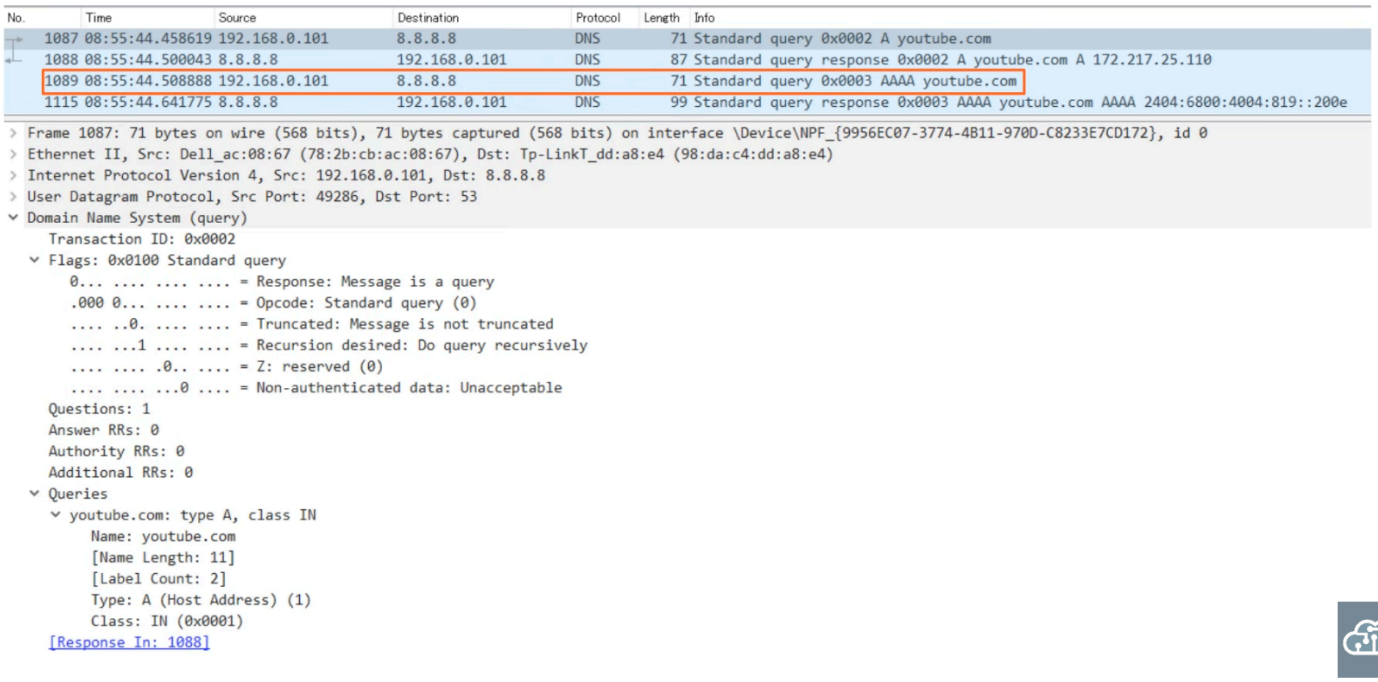
Message 2: First Response

No.	Time	Source	Destination	Protocol	Length	Info
1087	08:55:44.458619	192.168.0.101	8.8.8.8	DNS	71	Standard query 0x0002 A youtube.com
1088	08:55:44.500043	8.8.8.8	192.168.0.101	DNS	87	Standard query response 0x0002 A youtube.com A 172.217.25.110
1089	08:55:44.508888	192.168.0.101	8.8.8.8	DNS	71	Standard query 0x0003 AAAA youtube.com
1115	08:55:44.641775	8.8.8.8	192.168.0.101	DNS	99	Standard query response 0x0003 AAAA youtube.com AAAA 2404:6800:4004:819::200e

```
> Frame 1087: 71 bytes on wire (568 bits), 71 bytes captured (568 bits) on interface \Device\NPF_{9956EC07-3774-4B11-970D-C8233E7CD172}, id 0
> Ethernet II, Src: Dell_ac:08:67 (78:2b:cb:ac:08:67), Dst: Tp-LinkT_dd:a8:e4 (98:da:c4:dd:a8:e4)
> Internet Protocol Version 4, Src: 192.168.0.101, Dst: 8.8.8.8
> User Datagram Protocol, Src Port: 49286, Dst Port: 53
v Domain Name System (query)
    Transaction ID: 0x0002
  v Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v youtube.com: type A, class IN
        Name: youtube.com
        [Name Length: 11]
        [Label Count: 2]
        Type: A (Host Address) (1)
        Class: IN (0x0001)
    [Response In: 1088]
```

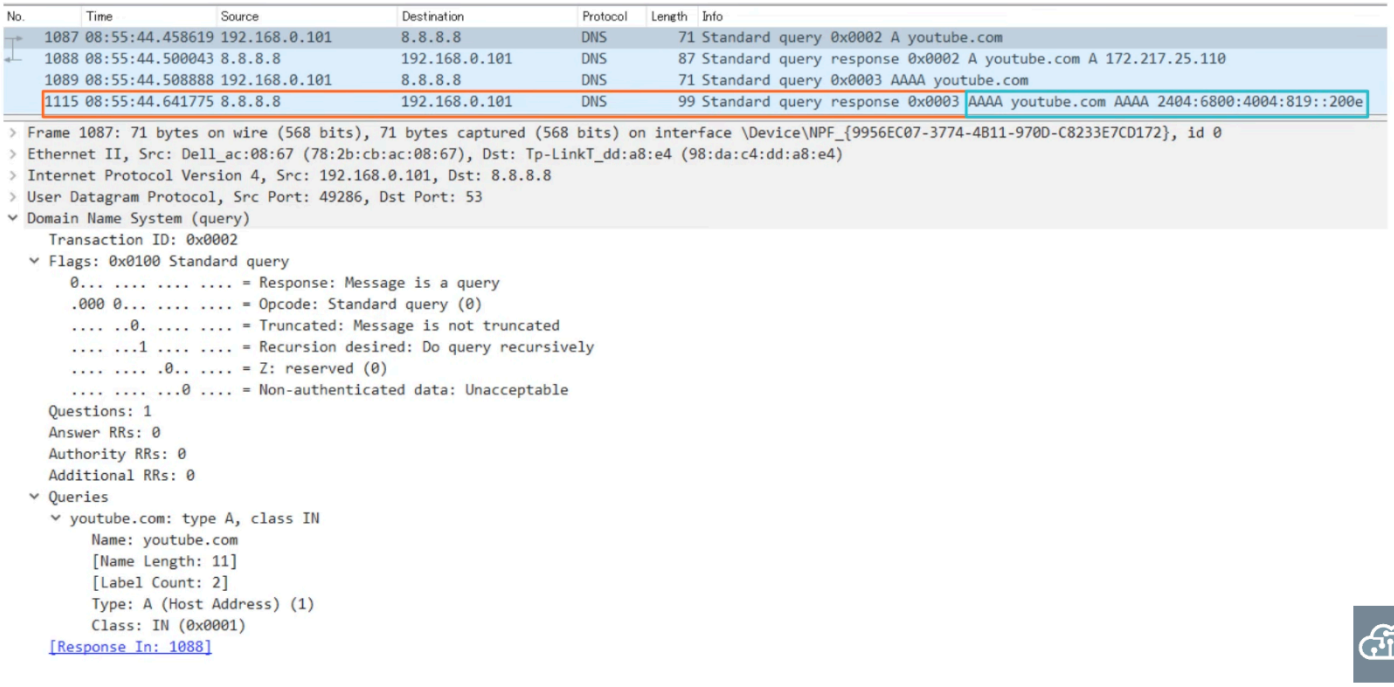
- **Source**: `8.8.8.8` (Google's DNS server)
- **Destination**: `192.168.0.101` (PC1)
- **Info**: Standard Query Response
- **Response Details**: `A youtube.com A 172.217.25.110` (IPv4 address)

Message 3: Second Query



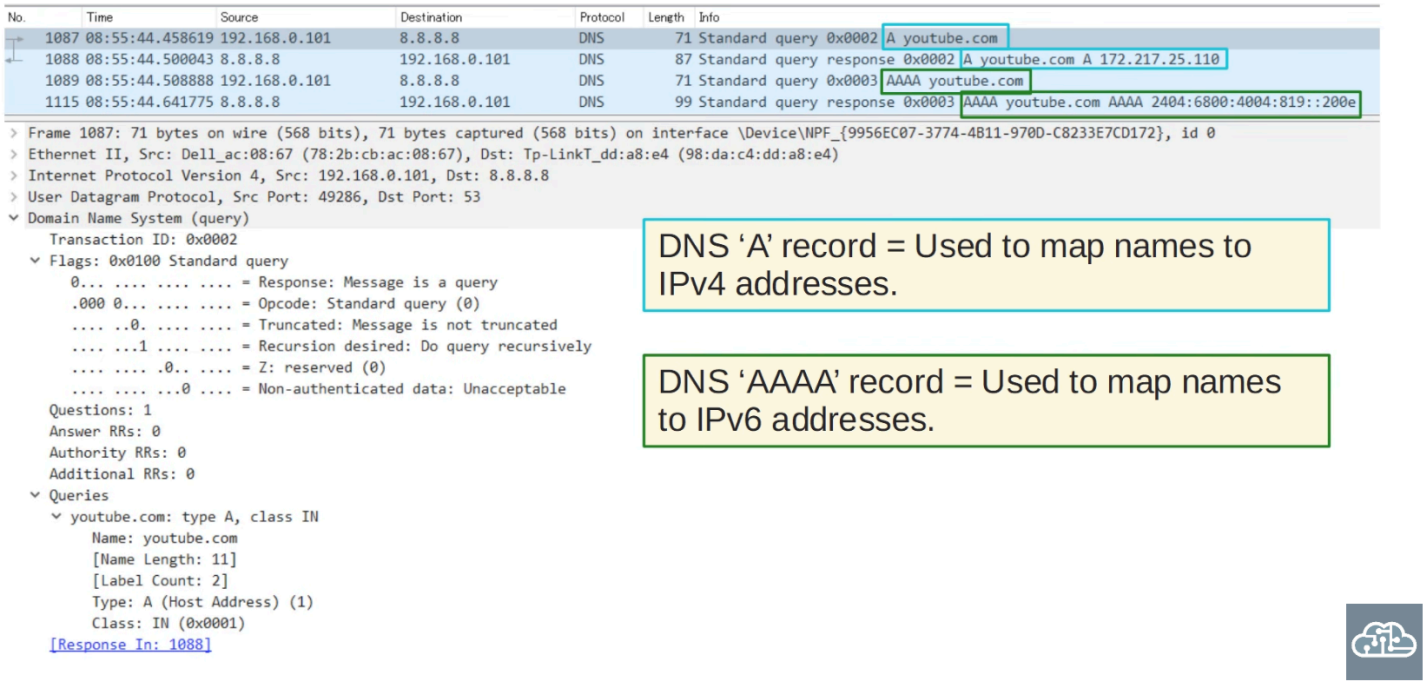
- **Source**: `192.168.0.101` (PC1)
- **Destination**: `8.8.8.8` (Google's DNS server)
- **Info**: Standard Query
- **Query Details**: `AAAA youtube.com` (referred to as "quadruple A")

Message 4: Second Response



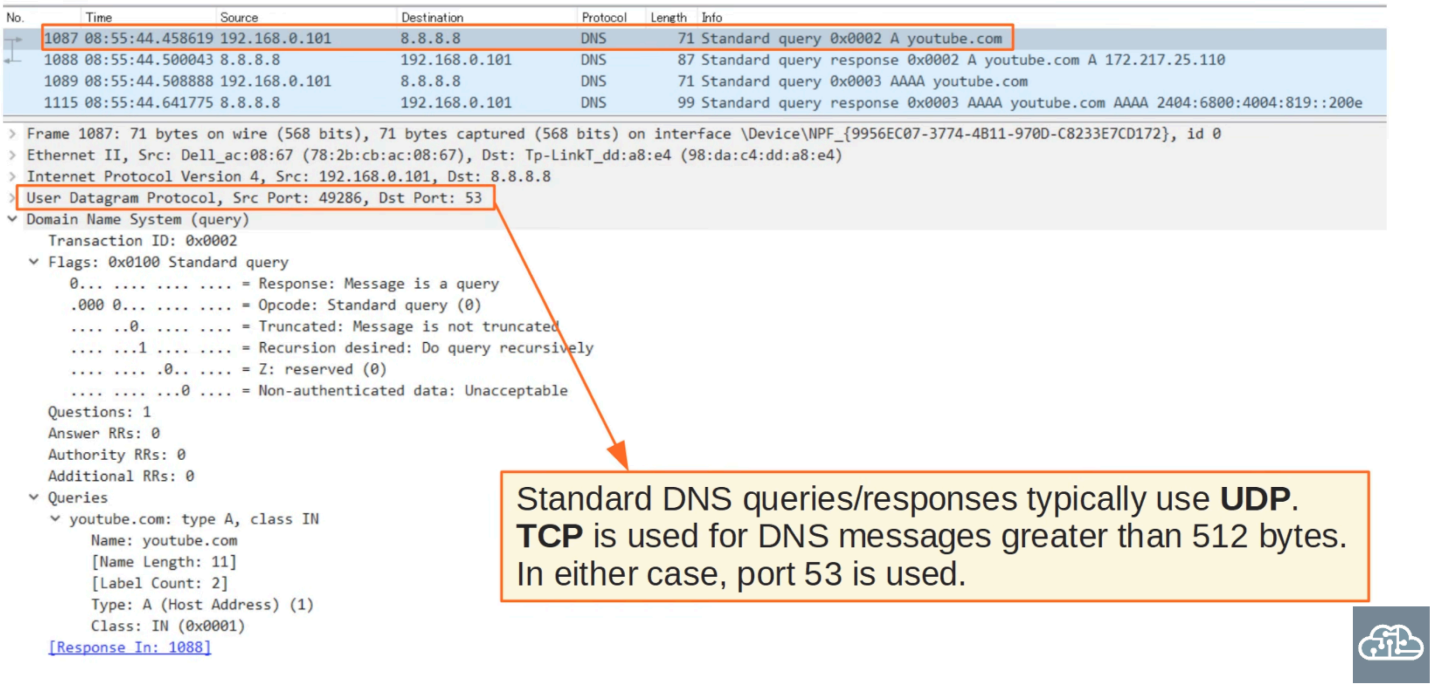
- **Source**: `8.8.8.8` (Google's DNS server)
- **Destination**: `192.168.0.101` (PC1)
- **Info**: Standard Query Response
- **Response Details**: `AAAA youtube.com AAAA` followed by an IPv6 address.

DNS Record Types



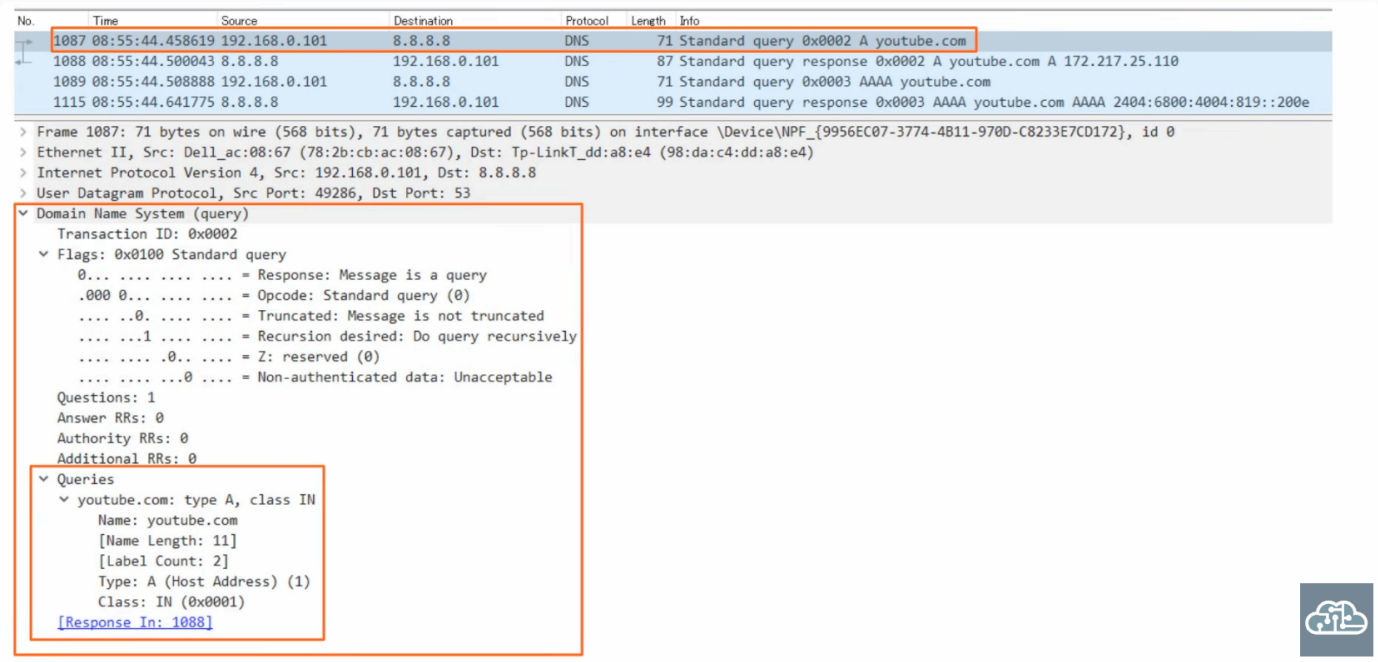
- **DNS 'A' records**: Used to map names to IPv4 addresses.
- **DNS 'Quadruple A' (AAAA) records**: Used to map names to IPv6 addresses.

Protocol Analysis



- **Layer 4 Protocol**: The capture shows the use of **UDP**.
- **Protocol Selection Rule**:
  - Standard DNS queries and responses typically use UDP.
  - TCP is used for DNS messages greater than 512 bytes.

- **Port Number**: DNS uses port **53** for both UDP and TCP.



- **UDP Query Content**: The actual UDP query inside the packet asks for the `A` record of `youtube.com`.

Wireshark Utility

- **Tool Value**: Wireshark is a very useful tool for studying and has real-world, on-the-job uses.
- **Recommendation**: The professor recommends downloading Wireshark to examine the network traffic your computer sends and receives.

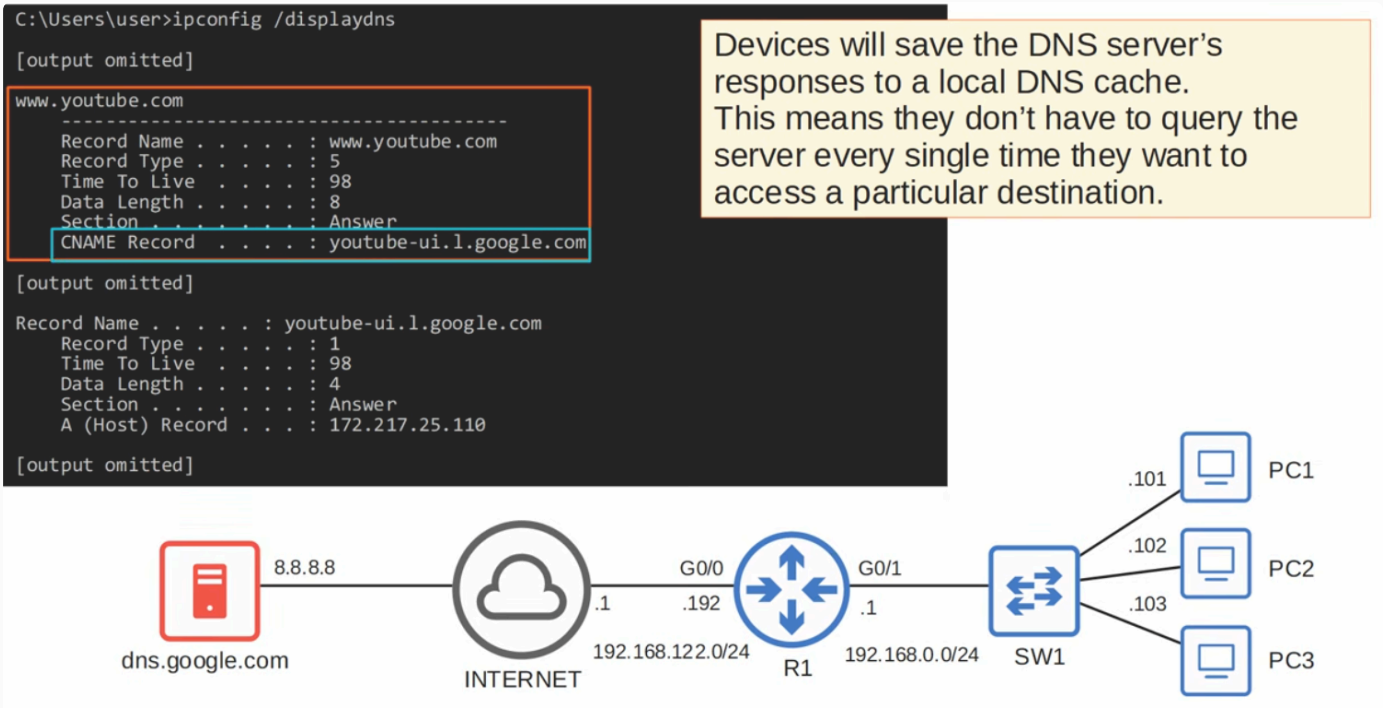
DNS Cache

- **Purpose**: Devices save DNS server responses to a local DNS cache so they don't have to query the server every single time they access a particular destination, saving unnecessary network traffic.

Viewing the DNS Cache

Example:

- The professor uses `IPCONFIG /DISPLAYDNS` to view the DNS cache on PC1.
- The record for `youtube.com` is displayed.

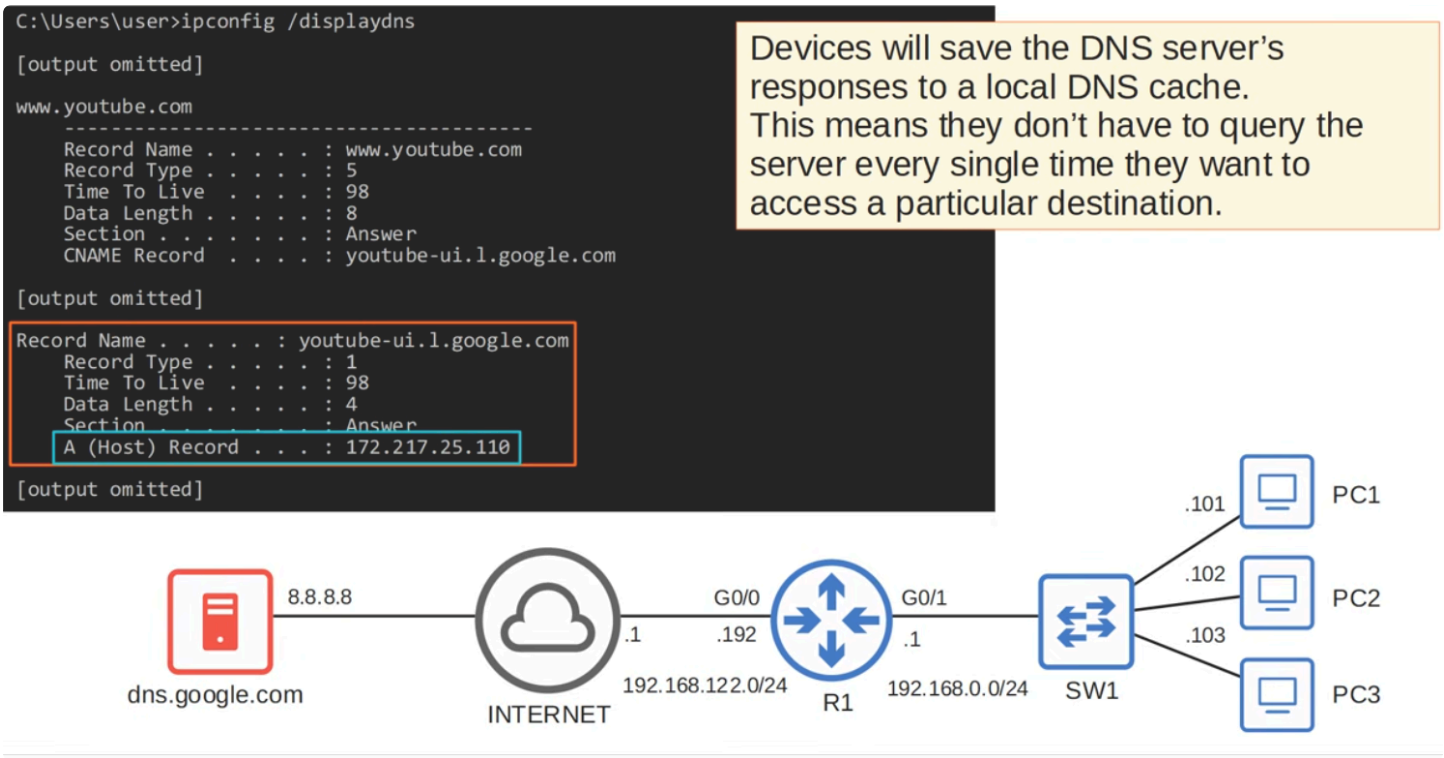


- The record type is not `A` or `AAAA`, but rather `CNAME`, which lists another domain name.

```
C:\> IPCONFIG /DISPLAYDNS
```

- **CNAME Record**: Canonical Name is a type of DNS record that maps a name to another name.

Example:

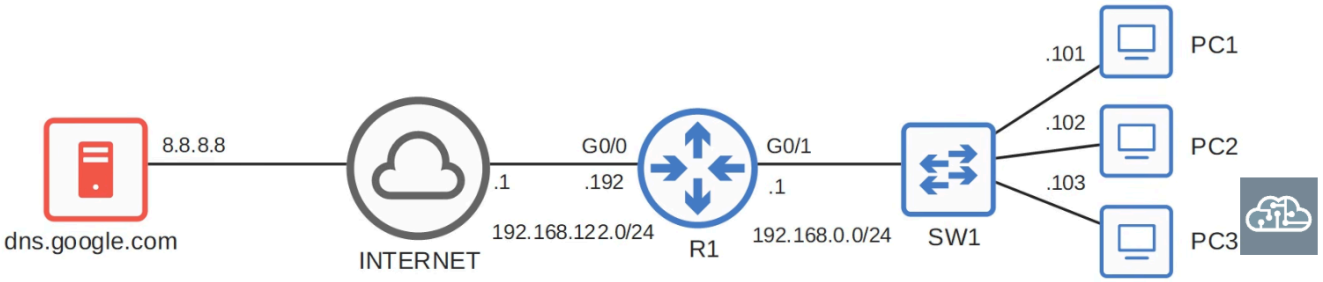


- The professor looks up that other domain name in the DNS cache.
- He finds the `A` record for that name, which contains the same IPv4 address: `172.217.25.110`.
- **CCNA Focus**: For the CCNA, you only need to know the basics of DNS. There are complexities like CNAME, but they are not required for the exam.

Clearing the DNS Cache

Example:

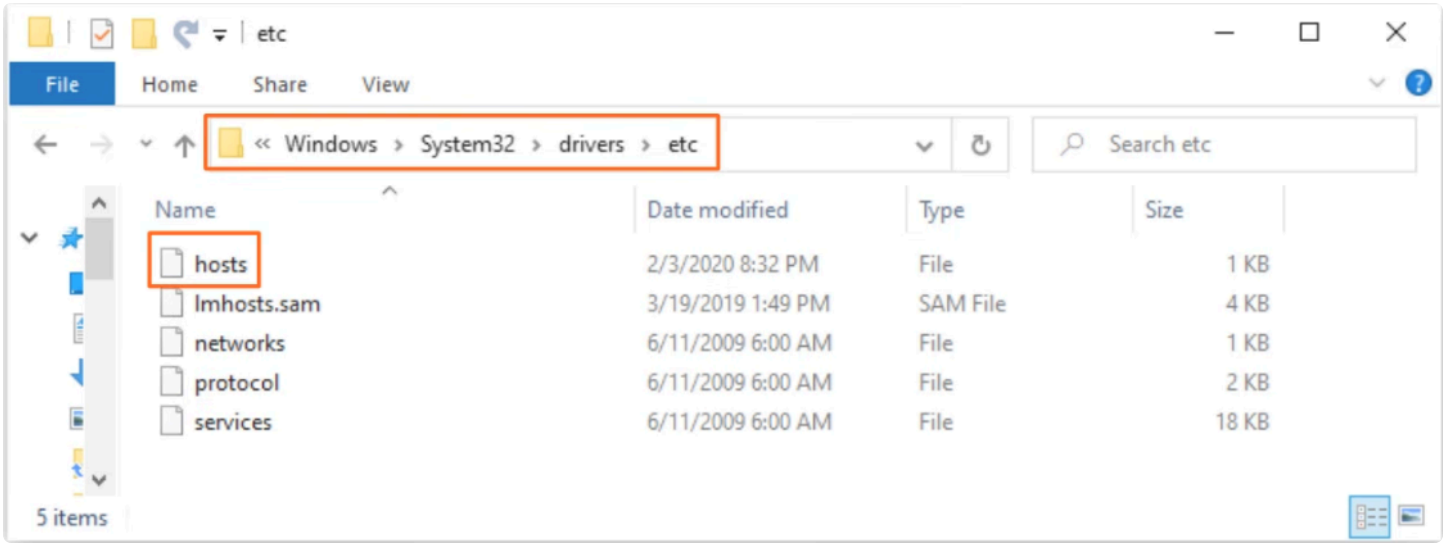
```
C:\Users\user>ipconfig /flushdns

Windows IP Configuration

Successfully flushed the DNS Resolver Cache.

C:\Users\user>ipconfig /displaydns

Windows IP Configuration

C:\Users\user>
```



- The professor uses `IPCONFIG /FLUSHDNS` to clear the DNS cache.
- The command output says the DNS resolver cache was flushed.
- After running `IPCONFIG /DISPLAYDNS` again, nothing is displayed.
- **Result**: If PC1 tries to access `youtube.com` again, it will have to send another DNS query to the DNS server to learn the IP address.

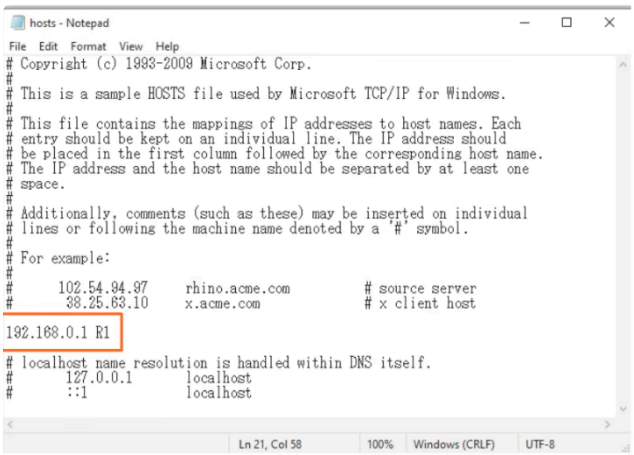
```
C:\> IPCONFIG /FLUSHDNS
```

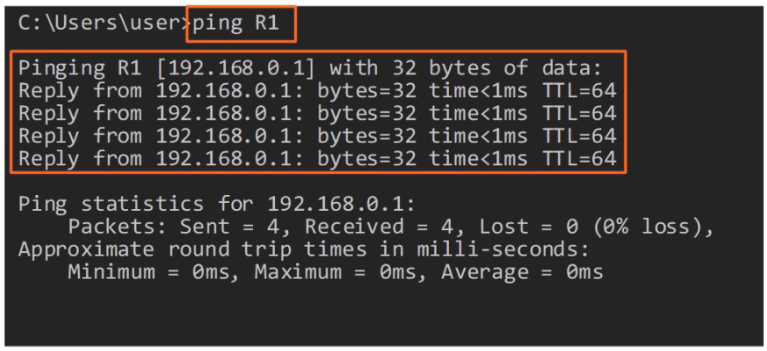
The Hosts File (Alternative to DNS)

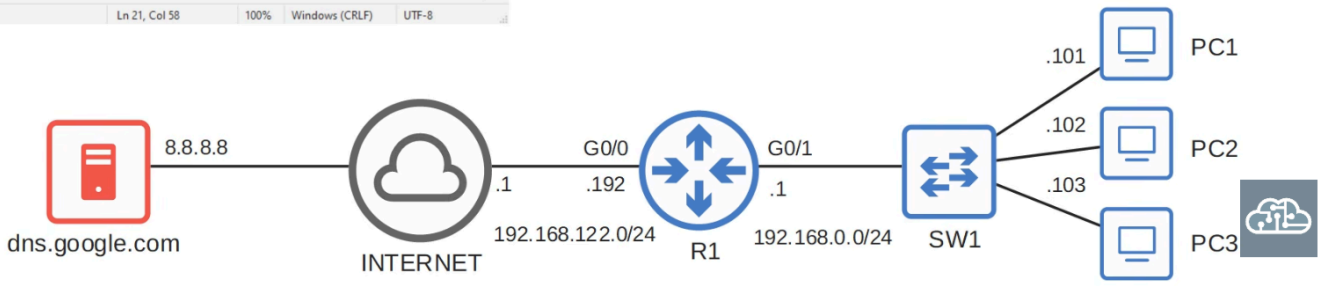


- **Definition**: In addition to a DNS cache, most devices have a **hosts file**, which is simply a list of hosts and IP addresses.
- **Function**: This acts as a simple alternative to DNS.
- **Windows Location**: The file is located in the `C:\Windows\System32\drivers\etc` folder, and the name is simply `hosts`.

Example:







- By default, the hosts file has no hosts listed.
- On PC1, the professor added an entry for R1 simply by typing the IP address, a space, and then the host name.
- Returning to the command prompt, the professor entered `PING R1`.
- PC1 was able to ping R1 because it had an entry in its hosts file.
- **Context**: This isn't DNS, it's a simple alternative. Host files like this were used a long time ago before DNS was invented.
- **Modern Usage**: In modern days, a hosts file might be used in a small network to list some hosts on the local network if necessary, however DNS is a much better solution.

General DNS Process Review

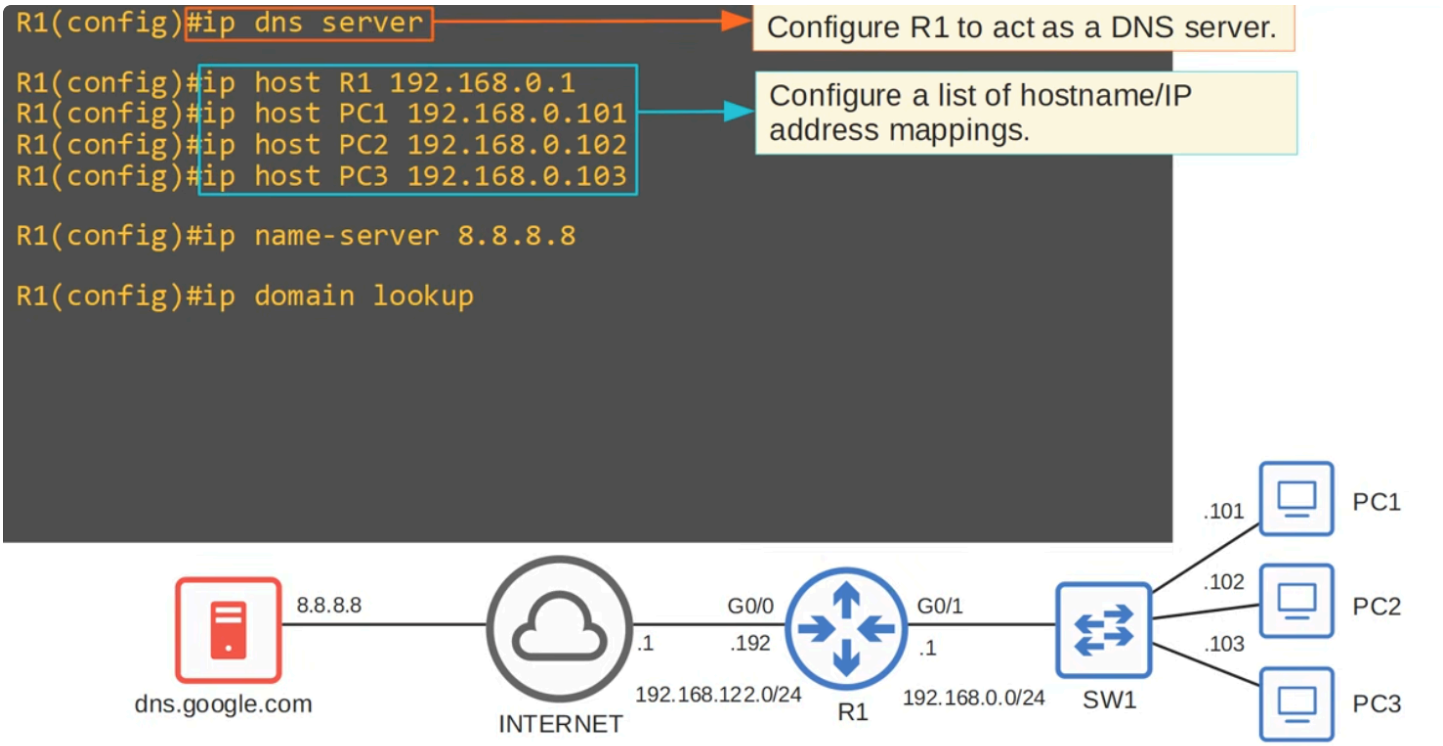
- **Process Summary**:
  1. A client wants to access a destination using a name (e.g., `youtube.com`).
  2. The client needs the IP address to access the website.
  3. The client sends a DNS query to a DNS server.
  4. The DNS server informs the client of the IP address.

Introduction to Cisco IOS DNS Configuration

- **Router Configuration Context**:
  - For hosts to use DNS, **you do not need to configure DNS on the routers**.
  - Routers will simply forward DNS messages like any other packets.
  - However, Cisco routers can be configured to act as DNS servers or clients.
- **Cisco Router as DNS Server**:
  - It is possible but rare.
  - If an "internal" (local network) DNS server is used, it is usually a Windows or Linux server rather than a router.

- **Cisco Router as DNS Client**:
  - This allows the router to execute commands (like `PING`) using names instead of IP addresses.

Configuring a Cisco Router as a DNS Server



- **Command: Enable DNS Server**: Use the `IP DNS SERVER` command in global configuration mode to enable the router to act as a DNS server. If a client queries R1, R1 will reply if it has the matching record.

```
R1(config)# ip dns server
```

- **Command: Configure Host Table**: Use the `IP HOST` command to manually build a host table mapping hostnames to IP addresses.

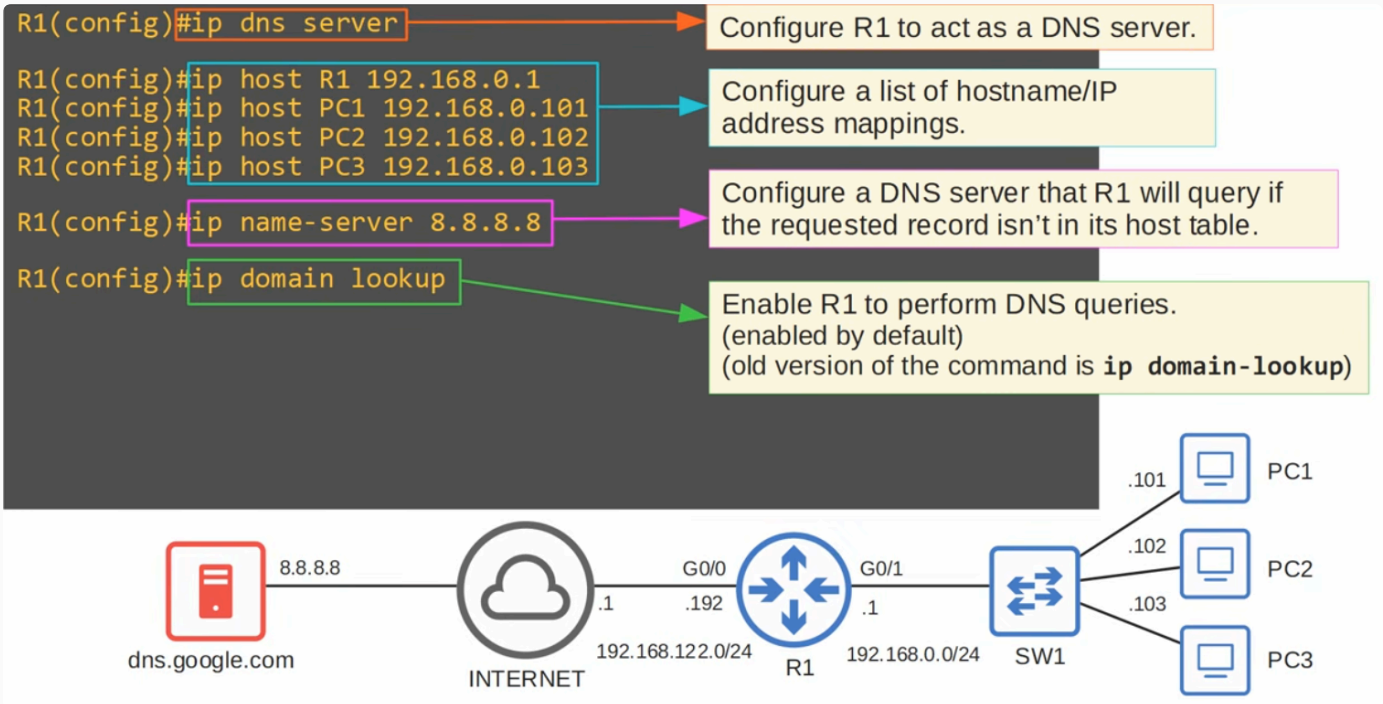
  Example:

  - The professor configured entries for R1, PC1, PC2, and PC3.

```
R1(config)# ip host R1 192.168.0.1
R1(config)# ip host PC1 192.168.0.101
R1(config)# ip host PC2 192.168.0.102
R1(config)# ip host PC3 192.168.0.103
```

- **Command: Configure External Server**: Configure an external DNS server. If R1 does not have a requested record in its own table, it will query this external server.

  Example:



  - The professor configured R1 to use Google's DNS server (`8.8.8.8`).

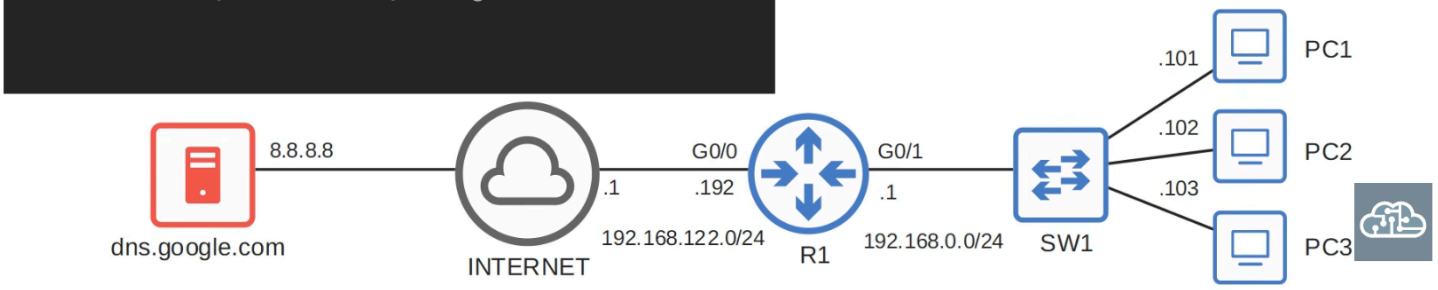
```
R1(config)# ip name-server 8.8.8.8
```

- **Command: Enable DNS Lookup**: Use the `IP DOMAIN LOOKUP` command to allow R1 to send DNS queries. Without this, R1 cannot query the external server.
  - **Note**: This command is **enabled by default**, so it usually does not need to be entered manually.
  - **Syntax Note**: The old version of the command includes a hyphen: `IP DOMAIN-LOOKUP`. Both versions are supported in modern IOS, and you should know both.

```
R1(config)# ip domain lookup
```

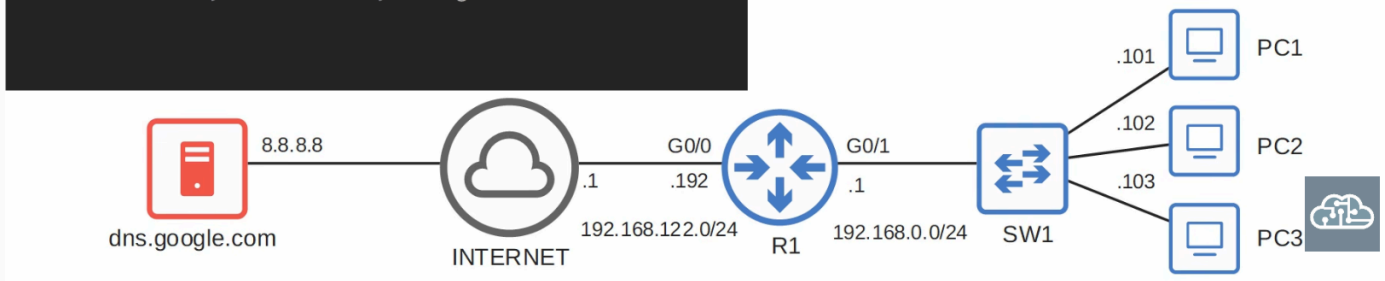
Demonstration: Internal DNS Query (PC1 to PC2)

```
C:\Users\user>ipconfig /all

[output omitted]

   IPv4 Address. . . . . . . . . . . : 192.168.0.101(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
   DNS Servers . . . . . . . . . . . : 192.168.0.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

[output omitted]

C:\Users\user>ping PC2 -n 1

Pinging PC2 [192.168.0.102] with 32 bytes of data:
Reply from 192.168.0.102: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.102:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

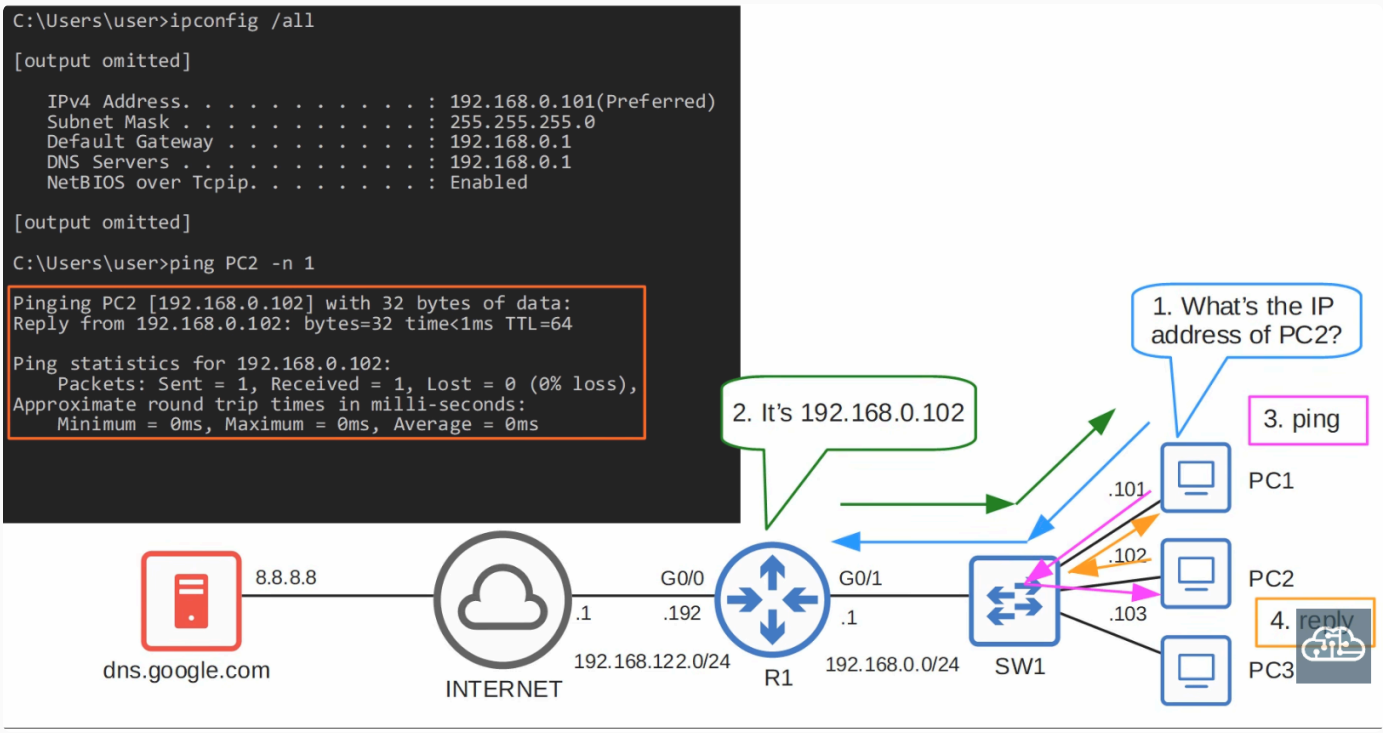


- **Setup**: PC1 is configured to use R1 as its DNS server.

```
C:\Users\user>ipconfig /all

[output omitted]

   IPv4 Address. . . . . . . . . . . : 192.168.0.101(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1
   DNS Servers . . . . . . . . . . . : 192.168.0.1
   NetBIOS over Tcpip. . . . . . . . : Enabled

[output omitted]

C:\Users\user>ping PC2 -n 1

Pinging PC2 [192.168.0.102] with 32 bytes of data:
Reply from 192.168.0.102: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.0.102:
    Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```



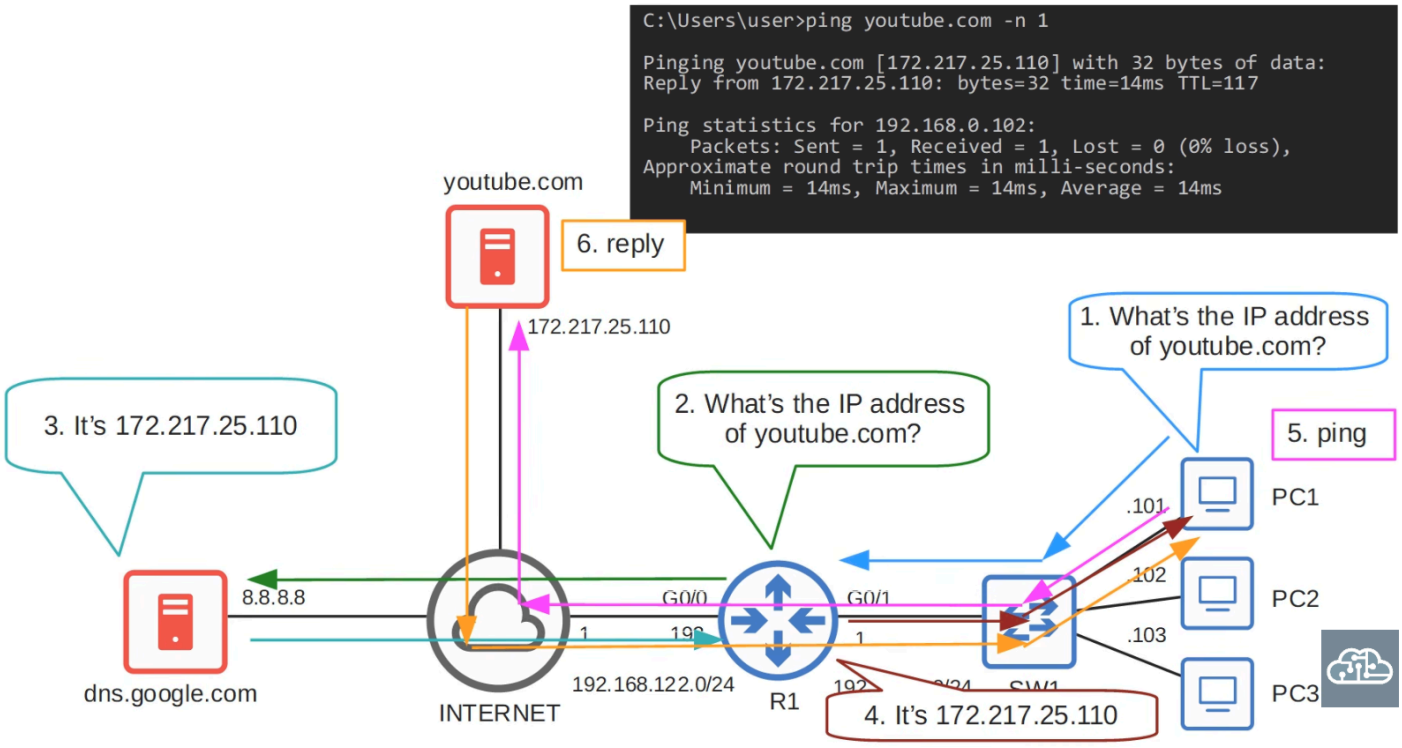
- **Action**: The `PING PC2 -n 1` command is issued on PC1 (using `-n 1` to send only a single ping).
- **Resolution Process**:



1. PC1 does not have an entry for PC2 in its host table, so it must use DNS to learn the IP address.
2. PC1 sends a DNS query to R1.
3. R1 has the entry for PC2 (configured earlier via the `IP HOST` command) and replies to PC1.
4. PC1 sends the ping to PC2; PC2 replies.

- **Caching Result**: PC1 now has an entry for PC2 in its DNS cache. Future pings to PC2 will not require a DNS query because PC1 already knows the IP address.

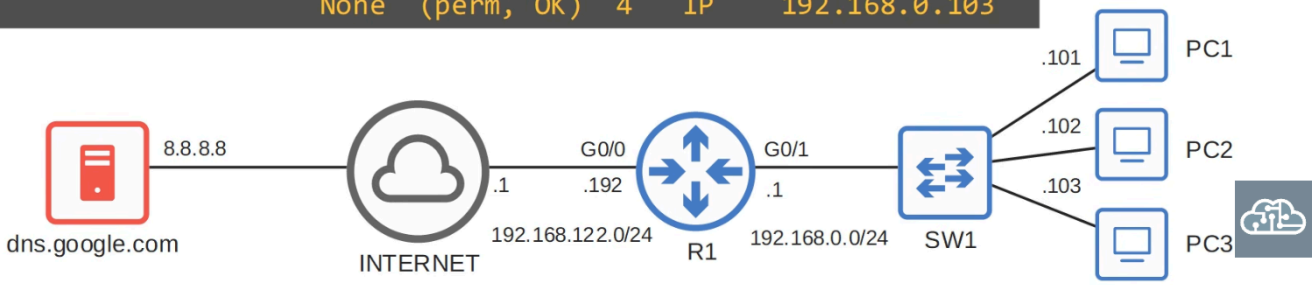
Demonstration: External DNS Query (PC1 to YouTube)



- **Action**: The `PING YOUTUBE.COM -N 1` command is issued on PC1.
- **Resolution Process**:
  1. PC1 sends a DNS query to R1 for the IP address of `youtube.com`.
  2. R1 does **not** have an entry for `youtube.com` in its host table.
  3. R1 acts as a DNS client and sends a query to the configured external server (Google's DNS at `8.8.8.8`).
  4. Google's DNS server replies to R1 with the IP address of `youtube.com`.
  5. R1 acts as a DNS server and replies to PC1 with the IP address.
  6. PC1 sends the ping to YouTube; YouTube replies.
- **Summary**: R1 acts as a DNS server for PC1. If R1 doesn't have the requested record, it acts as a DNS client to an external server (like Google's) to retrieve the information.

Verification: Viewing Host Entries on R1

```
R1#show hosts
Default domain is not set
Name/address lookup uses domain service
Name servers are 8.8.8.8

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined

Host                      Port  Flags      Age Type   Address(es)
youtube.com               None  (temp, OK)  0   IP    172.217.25.110
R1                        None  (perm, OK)  4   IP    192.168.0.1
PC1                       None  (perm, OK)  1   IP    192.168.0.101
PC2                       None  (perm, OK)  4   IP    192.168.0.102
PC3                       None  (perm, OK)  4   IP    192.168.0.103
```

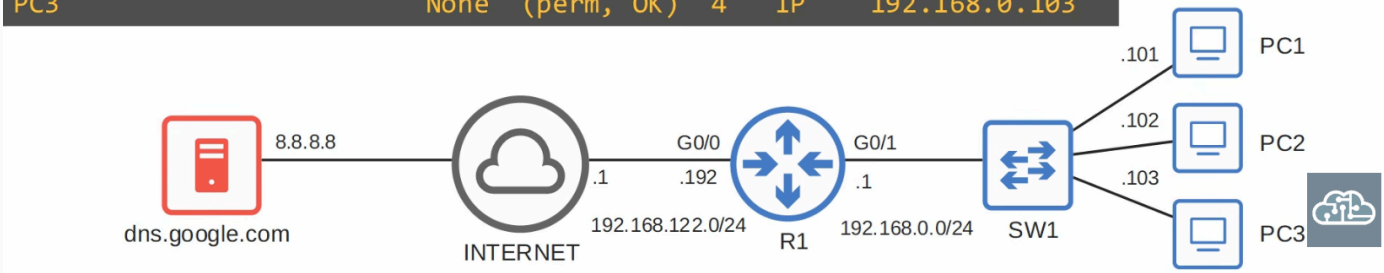


- **Command**: Use `SHOW HOSTS` to view both the manually configured hosts and the hosts learned/cached via DNS.

```
R1# show hosts
```

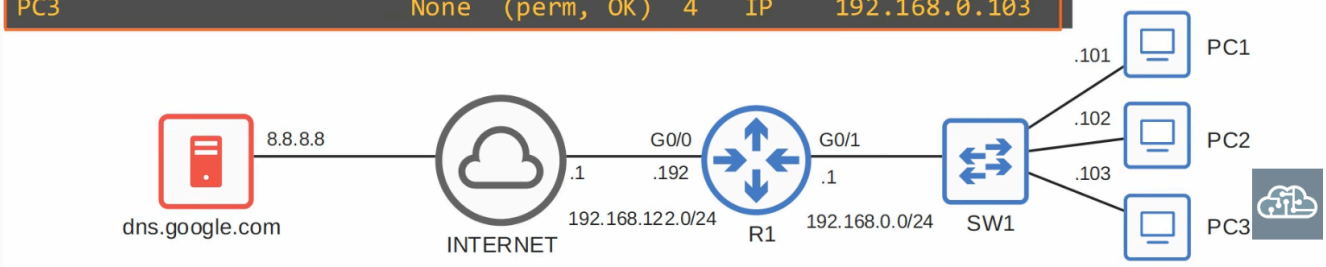
- **Output Analysis**:

```
R1#show hosts
Default domain is not set
Name/address lookup uses domain service
Name servers are 8.8.8.8

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined

Host                      Port  Flags      Age Type   Address(es)
youtube.com               None  (temp, OK)  0   IP     172.217.25.110
R1                        None  (perm, OK)  4   IP     192.168.0.1
PC1                       None  (perm, OK)  1   IP     192.168.0.101
PC2                       None  (perm, OK)  4   IP     192.168.0.102
PC3                       None  (perm, OK)  4   IP     192.168.0.103
```



- **Cached Entries**: You can see the cached entry for `youtube.com` that R1 learned from Google's DNS server.

```
R1#show hosts
Default domain is not set
Name/address lookup uses domain service
Name servers are 8.8.8.8

Codes: UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined

Host                      Port  Flags      Age Type   Address(es)
youtube.com               None  (temp, OK)  0   IP     172.217.25.110
R1                        None  (perm, OK)  4   IP     192.168.0.1
PC1                       None  (perm, OK)  1   IP     192.168.0.101
PC2                       None  (perm, OK)  4   IP     192.168.0.102
PC3                       None  (perm, OK)  4   IP     192.168.0.103
```

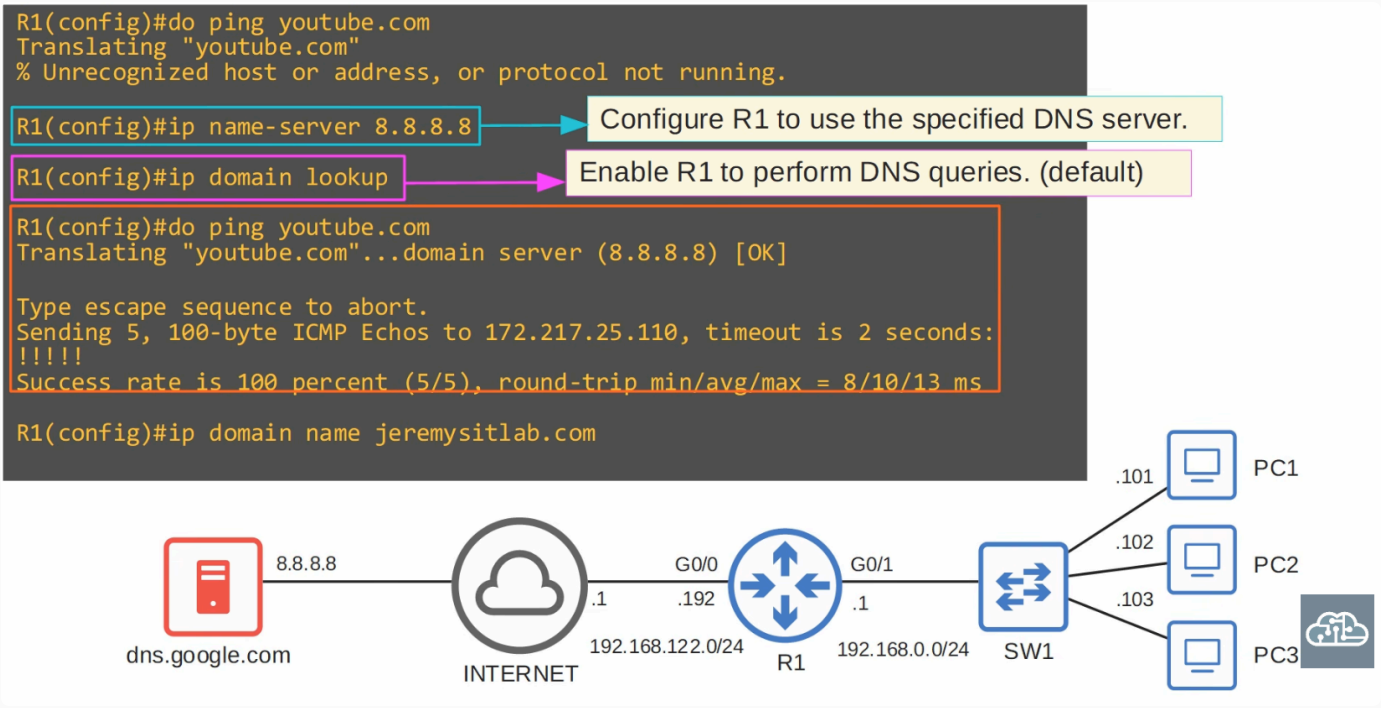


- **Flags Column**: Look for 'temp' (temporary). Because this entry was learned via DNS, it is not permanent. If it expires, it will have to be re-learned.

- **Permanent Entries**: Manually configured entries (using `IP HOST`) are displayed as permanent in the flags column.

Configuring a Cisco Router as a DNS Client

- **Context**: To configure a router only as a DNS client (not a server), you use two commands.
- **Demonstration**:

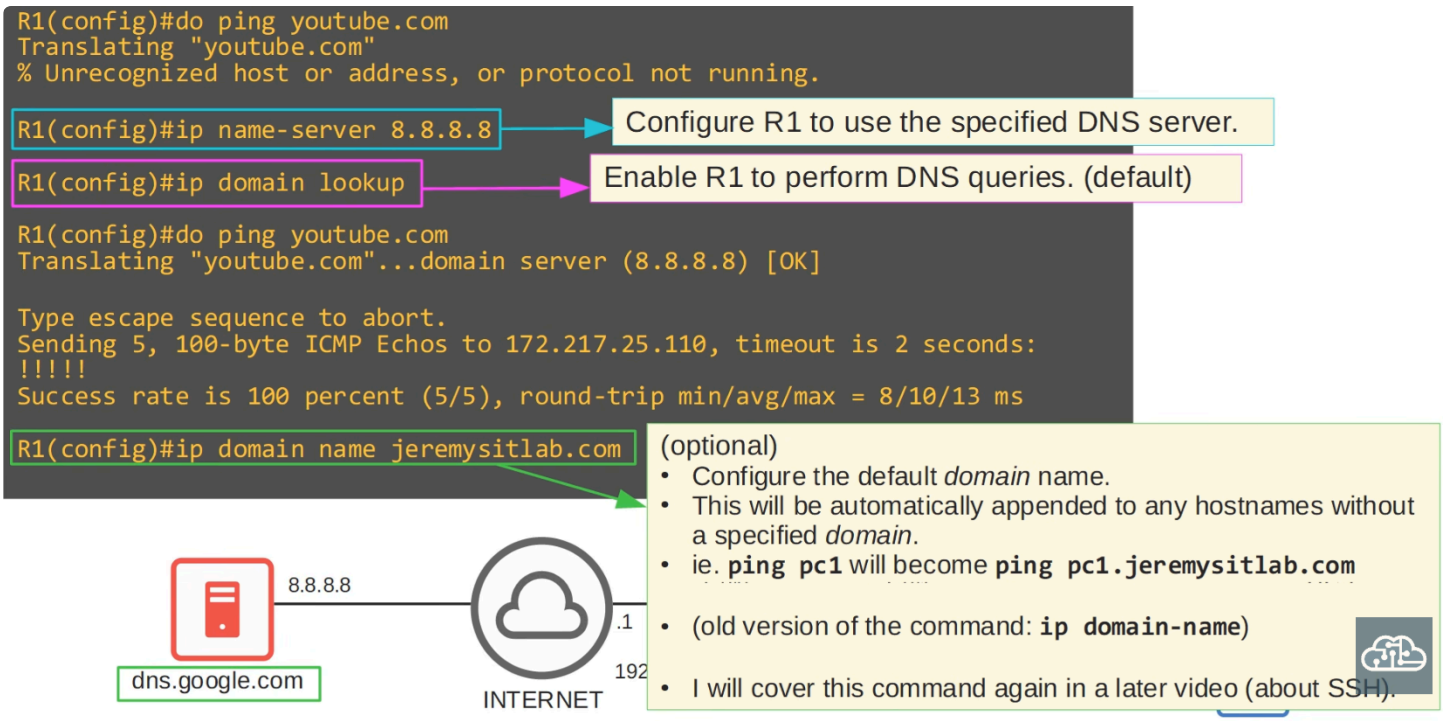


1. The professor deleted all DNS settings from R1.
2. Attempted to `ping youtube.com`, but it failed because R1 could not translate the name to an IP address.
3. Configured R1 to use Google's DNS server using `IP NAME-SERVER 8.8.8.8`.
4. Configured R1 to enable DNS queries using `IP DOMAIN LOOKUP`.
    - **Note**: This command is enabled by default, so you don't actually have to enter it, but make sure you know it.
5. The ping was successful.

- **Important Distinction**: In this configuration, R1 is a DNS client only, **not** a DNS server. If PC1 tries to use R1 as a DNS server, R1 will not reply to PC1's queries.

```
R1(config)# ip name-server 8.8.8.8
R1(config)# ip domain lookup
```

Optional Command: IP DOMAIN NAME



- **Purpose**: Sets a default domain name for the router.
- **Definition**: Domain names like 'google.com' or 'jeremysitlab.com' define a realm or area of administrative control in the Internet.
  - **Example**: 'google.com' tells us that 'mail.google.com', 'dns.google.com', and 'time.google.com' all fall under Google's administration.
- **Function**: When applied, this default domain name is appended to all hostnames without a specified domain name.
  - **Example**: The command `PING PC1` will actually become `PING PC1.JEREMYSITLAB.COM`.

- **Syntax Note**: There is an older version of the command with a hyphen: `IP DOMAIN-NAME`. Both versions are supported in modern IOS.
- **Relevance**: A domain name is required to enable SSH (Secure Shell), which will be covered in a later video.

```
R1(config)# ip domain name jeremysitlab.com
```

Command Review

- **Windows Commands**:
  - `IPCONFIG /ALL`: Displays various info including DNS server.
  - `NSLOOKUP`: Name Server Lookup to query DNS server for IP address.
  - `IPCONFIG /DISPLAYDNS`: View the DNS cache.
  - `IPCONFIG /FLUSHDNS`: Clear the DNS cache.
- **Cisco IOS Commands**:
  - `IP DNS SERVER`: Enable router to act as a DNS server.
  - `IP HOST`: Build a host table with hostname-to-IP mappings.
  - `IP NAME-SERVER`: Configure an external DNS server.
  - `IP DOMAIN LOOKUP` (or `IP DOMAIN-LOOKUP`): Enable DNS queries (enabled by default).
  - `SHOW HOSTS`: View configured and cached host entries.
  - `IP DOMAIN NAME` (or `IP DOMAIN-NAME`): Set a default domain name for the router.

Lecture Summary

- **Purpose of DNS**: Enables the use of easy-to-remember names like `google.com` and `youtube.com` instead of IP addresses.
- **Basic Functions**: DNS clients send queries to DNS servers, asking for the IP address of a particular name, and the server replies with the IP address.

- **Windows Demonstration**: Showed how a PC uses a DNS server to learn the IP address of `youtube.com` and adds it to its DNS cache.
- **Cisco IOS Configuration**: Showed how to configure a Cisco router to be a DNS server and a DNS client.
- **Key Point**: In most cases, you don't need to configure DNS on a router. Usually, a separate DNS server fills that role. However, you still should know how to configure it.