

实验 26：DNS 实验

| 姓名 | 学号 | 合作学生 | 指导教师 | 实验地点 | 实验时间 |
|-----|---------|------|------|---------|------------|
| 林继申 | 2250758 | 无 | 陈伟超 | 济事楼 330 | 2024/05/23 |

【实验目的】

DNS 实验的主要目的是通过分析 DNS（域名系统）的工作原理和过程，让学生理解和掌握如何将域名转换为 IP 地址的技术。实验将涉及到理解 DNS 的结构和功能，包括其层次数据库模式和分布式集群工作方式。此外，学生将通过实际操作，如使用 Packet Tracer 和 Wireshark 工具，来观察和分析 DNS 查询和响应的过程。这样的实验能够帮助学生深入了解网络协议的应用层面，以及 DNS 对于互联网通信的重要性。

【实验原理】

一、DNS 概述

DNS（Domain Name System，域名系统）是互联网的一个基础服务，作为将域名和 IP 地址相互映射的分布式数据库，允许用户通过使用容易记忆的域名来访问互联网资源，而不是复杂的数字 IP 地址。这种机制不仅大大简化了用户访问网络服务的方式，还对整个互联网的运行和管理起到了关键作用。

二、DNS 过程

DNS（域名系统）的核心任务是解析域名到相应的 IP 地址，使得其他应用层协议如 HTTP、SMTP、FTP 等可以使用这些 IP 地址进行网络通信。DNS 的过程是一个从用户设备到 DNS 服务器的多步骤查询流程，涉及本地客户端、DNS 缓存、根服务器、顶级域名服务器和权威域名服务器。以下详细描述了 DNS 解析的完整过程：

1. 用户端触发 DNS 查询

- 用户在浏览器地址栏输入一个 URL，例如 `http://www.baidu.com/`。
- 浏览器从 URL 中提取出域名 `www.baidu.com`，并将其传递给操作系统内部的 DNS 解析客户端。

2. 本地 DNS 客户端发起请求

- DNS 客户端首先检查本地缓存是否已经有该域名的解析结果。如果有缓存结果，直接使用该 IP 地址完成后续操作。如果没有，客户端将向配置

的 DNS 服务器发送 DNS 查询请求。

3. DNS 解析过程

- 递归查询：用户的 DNS 客户端通常配置有本地 ISP 的 DNS 服务器。该服务器会代表用户进行递归查询，即全权负责将查询一路跟进到得到答案。
- 迭代查询：本地 DNS 服务器首先查询根 DNS 服务器，根服务器不会直接解析域名，而是指向负责该顶级域（如 .com）的顶级域名服务器。
- 本地 DNS 服务器接着向顶级域名服务器发出查询请求，顶级域名服务器再指向该二级域名（如 baidu.com）的权威 DNS 服务器。
- 本地 DNS 服务器最后向权威 DNS 服务器查询具体的记录（如 A 记录，即 IPv4 地址），权威服务器返回域名对应的 IP 地址。

4. 接收 DNS 响应

- 本地 DNS 服务器收到权威 DNS 服务器提供的 IP 地址后，会将该地址缓存（以减少未来对同一域名的查询时间），然后将解析结果返回给用户的 DNS 客户端。

5. 完成网络连接

- 用户设备的 DNS 客户端接收到 IP 地址后，浏览器使用这个 IP 地址与目标 HTTP 服务器建立 TCP 连接，开始 Web 会话。

这一过程虽在描述中步骤繁多，但实际上通常在几毫秒到几秒钟内完成，取决于网络状况、服务器响应速度和查询路径的长度。DNS 系统的这种设计使得互联网能够以非常动态和分布式的方式运行，同时保持了对数十亿设备和服务的高效寻址能力。

三、DNS 服务体系架构

DNS（域名系统）是一个全球分布式的服务，其主要作用是将人类可读的主机名（如 `www.google.com`）转换为机器可读的 IP 地址（如 `192.168.1.1`）。这个系统的设计允许因特网用户通过域名访问网站，而不需要记忆复杂的数字地址。

1. 应用程序的调用：应用程序（例如 Web 浏览器或电子邮件客户端）需要解析域名时，会调用系统内置的 DNS 解析器。在 Unix-like 系统中，这通常是通过 `gethostbyname()` 函数实现。
2. DNS 客户端查询：DNS 客户端（也称为解析器）向配置的 DNS 服务器发送 DNS

查询报文，这通常使用 UDP 协议，通过端口 53。

3. DNS 服务器：DNS 的服务器体系结构包括多层服务器。
 - 根 DNS 服务器：回答来自下级 DNS 服务器的关于顶级域（如 .com、.net）的查询。
 - 顶级域 DNS 服务器（TLD）：管理域名注册服务，回答关于第二级域名的查询（如 .google.com）。
 - 权威 DNS 服务器：提供最终的域名到 IP 地址的映射。如果权威服务器的数据没有在本地缓存，将由递归服务器查询获取。
4. 递归与缓存：DNS 查询可能涉及多个服务器，从根到顶级，再到权威服务器，直到获取到请求的 IP 地址。为减少延迟和网络负载，DNS 服务器会缓存查询结果，缓存的持续时间由域名的 TTL（生存时间）决定。

四、DNS 分布式集群工作方式

1. 分布式架构：为了避免单点故障和处理上亿台主机发出的 DNS 查询，DNS 采用分布式架构。
 - 可扩展性：分布式设计允许系统水平扩展，以应对持续增长的互联网设备和服务。
 - 容错性：多个服务器分散全球，每个层次的 DNS 服务器都有多个冗余选项，增加了整个系统的稳定性和可靠性。
 - 性能优化：通过地理分散的服务器，减少了用户与 DNS 服务器之间的距离，从而减少了查询的响应时间。
2. 避免集中式设计的问题：
 - 单点故障：集中式系统若出现故障，会影响全球的域名解析服务。
 - 通信容量和维护成本：集中式系统需要处理所有的 DNS 查询和更新操作，难以应对全球范围内的巨大流量和动态更新。

五、DNS 域名

1. 域名称和域命名空间
 - 域命名空间：DNS 的域名称形成了一个层次化的树状结构，这个结构被称为域命名空间。每个节点在这个树中代表一个域名的部分，节点从树的顶部（根域，通常表示为一个点“.”）到具体的主机名。

- 完全限定域名 (FQDN): FQDN 是包括主机名在内的完整域名路径, 它指定了从最高级的根域到具体主机的完整路径。例如, `im.qq.com` 是一个 FQDN, 其中 `im` 是主机名, `qq` 是二级域名, `com` 是顶级域名。

2. DNS 域名称的类别

DNS 中的域名称可以根据其在域名空间中的位置和功能分类:

- 顶级域名 (TLDs): 顶级域名位于 DNS 层次结构的最高层。这些通常分为两大类:
- 通用顶级域名 (gTLDs): 如 `.com`、`.org`、`.net` 等。
- 国家代码顶级域名 (ccTLDs): 每个国家或地区具有特定的两字母代码, 如 `.uk` (英国)、`.de` (德国) 等。
- 二级域名和子域名: 位于顶级域名下面, 例如在 `example.com` 中, `example` 是二级域名。子域名如 `support.example.com` 中的 `support`, 用于进一步细分和组织网站的内容。

3. 域名资源记录 (Resource Records, RR)

DNS 使用一系列的资源记录来描述域内的各种资源, 这些记录包括:

- A 记录: 将域名映射到 IPv4 地址。
- AAAA 记录: 将域名映射到 IPv6 地址。
- NS 记录 (Name Server): 指定该域名由哪个 DNS 服务器控制。
- MX 记录 (Mail Exchange): 指定处理该域名邮件的邮件服务器。
- CNAME 记录 (Canonical Name): 允许将一个域名解析到另一个域名, 而不是一个 IP 地址。

六、DNS 服务工作过程

DNS 服务的工作过程是一个精妙而高效的系统, 它涉及多个步骤和组件, 以确保用户能够快速地将易记的域名转换为计算机能够理解的 IP 地址。这个过程分为递归查询和迭代查询两个主要部分。以下是详细介绍:

1. 客户端发起 DNS 查询: 当用户在浏览器中输入一个域名如 `www.qq.com` 时, 操作系统首先检查本地的 `hosts` 文件是否存在该域名的映射关系。如果存在, 直接使用该映射完成域名解析。
2. 本地 DNS 解析器缓存查询: 如果 `hosts` 文件中没有该域名的记录, 操作系统

的 DNS 解析器将检查自己的缓存是否有这个网址的映射关系。如果缓存中有记录，直接返回这个 IP 地址。

3. 查询本地 DNS 服务器：如果本地缓存也没有找到相应的映射关系，DNS 解析器将向配置的本地 DNS 服务器发起查询。本地 DNS 服务器首先检查它自己的缓存，如果有缓存结果，即使这个结果不具有权威性，也会被返回给客户端以加快解析过程。
4. 递归与迭代查询：如果本地 DNS 服务器本身也无法解析该域名，它将进行以下操作之一：
 - 非转发模式：直接向根 DNS 服务器发起请求，根 DNS 服务器会指向负责该顶级域（如 .com）的 TLD 服务器。本地 DNS 服务器然后向 TLD 服务器查询，获取更具体的权威 DNS 服务器地址（如 qq.com）。这个过程会继续，直至找到具体的 www.qq.com 的记录。
 - 转发模式：如果本地 DNS 服务器配置为转发模式，它将查询请求转发至上一级 DNS 服务器。如果上级服务器也无法解析，则请求可能被转发至更高级别的服务器或直接查询根 DNS 服务器。
5. 返回响应给客户端：无论是通过递归查询还是迭代查询，最终得到的解析结果都会被发送回原始请求的本地 DNS 服务器，然后由此服务器返回给客户端的 DNS 解析器，最后解析器将这一信息提供给浏览器或其他请求应用程序。
6. 缓存结果：为了提高效率，DNS 查询结果通常会在本地 DNS 服务器和客户端的 DNS 解析器中缓存一段时间，这样相同的查询在未来可以直接从缓存中获取答案，减少解析时间。

七、DNS 域名解析顺序

DNS 域名解析顺序是一个层次化和有序的查询过程，涉及从客户端到全球分布的服务器的多个步骤。这个解析顺序确保了域名能够高效且准确地被转换为对应的 IP 地址。以下是 DNS 域名解析的详细步骤：

1. 浏览器缓存：当用户在浏览器中输入一个域名时，浏览器首先检查自己的缓存，看是否已经存储了该域名的 IP 地址。如果找到了，就直接使用这个地址，这是最快的解析方式。
2. 系统缓存：如果浏览器缓存中没有找到域名的记录，解析器接下来会检查操

作系统的 DNS 缓存。系统缓存包括了之前解析过程中获取的域名记录，以及 hosts 文件中的静态指定。

3. 路由器缓存：如果系统缓存中也没有相应的记录，查询请求将转到网络路由器，许多家用路由器也会缓存 DNS 查询结果，以减少网络延迟和频繁的外网 DNS 查询。
4. ISP DNS 缓存：如果本地设备和路由器都无法解析域名，请求会被发送到 ISP 提供的 DNS 服务器。这些服务器拥有广泛的缓存，可能包含了所需的域名解析结果。
5. 根域名服务器：如果 ISP 的 DNS 服务器也无法解析，它会向根域名服务器发出查询。全球共有 13 台根域名服务器，它们知道所有顶级域名服务器的位置。
6. 顶级域名服务器（TLD 服务器）：根服务器会将查询指向负责该域名后缀（如 .com、.net）的顶级域名服务器。顶级域名服务器管理着该后缀下所有域名的分配和解析。
7. 权威域名服务器：顶级域名服务器将查询指向负责该具体域名的权威域名服务器。权威服务器是最终存储域名对应 IP 地址记录的地方。
8. 缓存结果：一旦本地 DNS 服务器从权威服务器接收到 IP 地址，它将此结果缓存起来，用于加速未来的查询。然后，它将 IP 地址返回给发起请求的客户端。
9. 建立连接：客户端拿到 IP 地址后，便开始尝试与该地址对应的服务器建立网络连接，从而访问网站或服务。

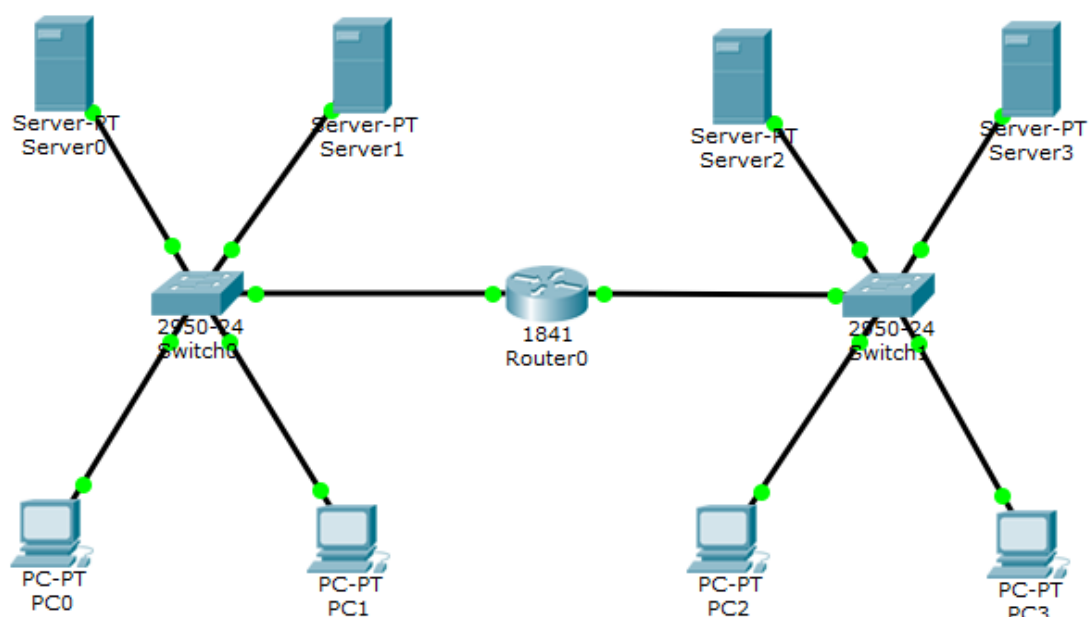
通过这个层次化和分布式的查询机制，DNS 能够在全球范围内提供快速、可靠的域名解析服务，支撑互联网的运作。每一步的缓存都旨在减少延迟，优化性能，同时确保数据的最新和准确性。

【实验设备】

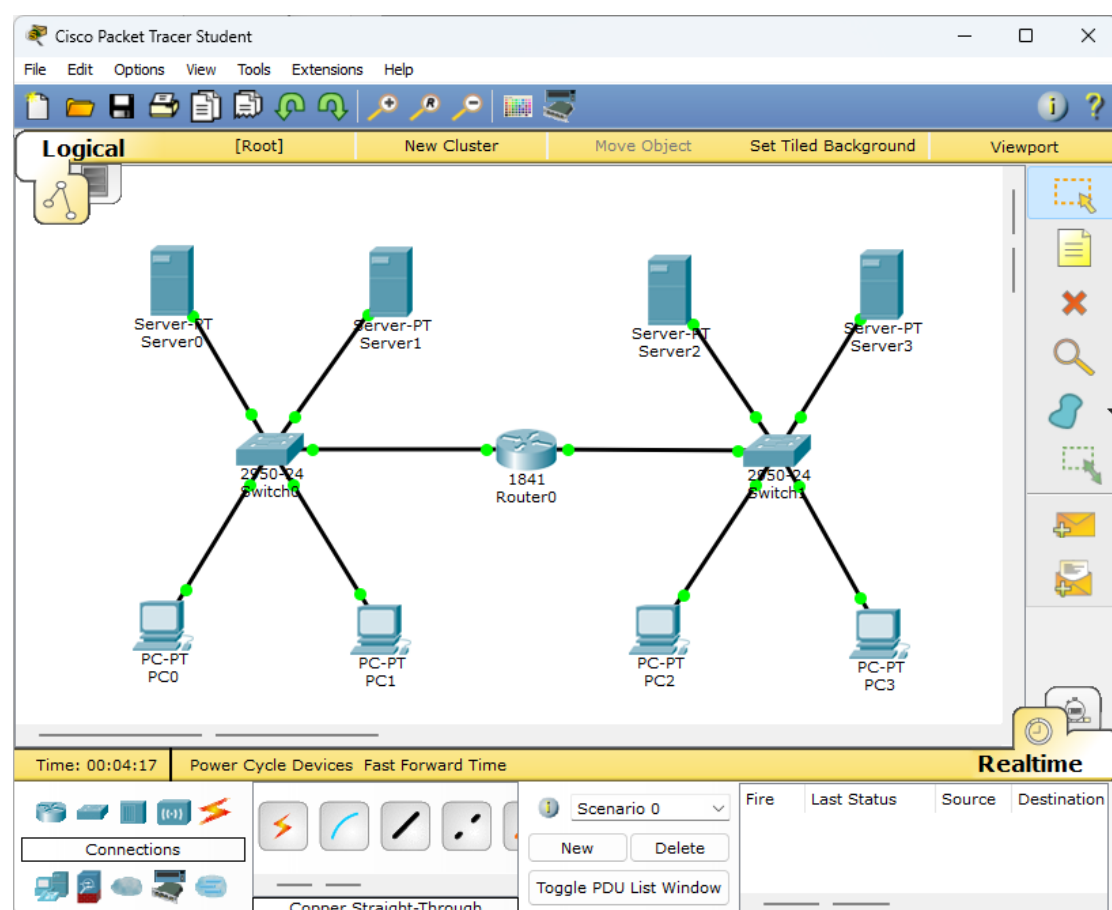
1. 操作系统：Windows 10
2. 网络环境：局域网
3. 应用程序：Cisco Packet Tracer 6.0

【实验步骤】

1. 规划网络地址及拓扑图。



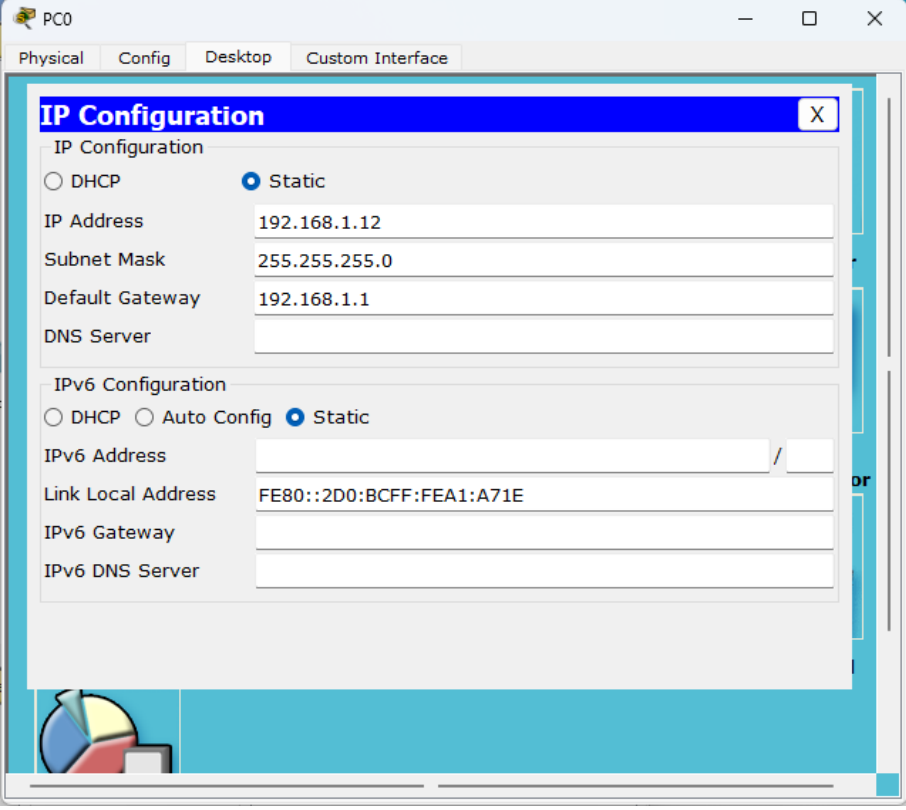
2. 启动 Cisco Packet Tracer, 按照上图连接网络。



3. 为每台 PC 机配置如下静态 IP 地址、子网掩码和默认网关。

| PC | IP Address | Subnet Mask | Default Gateway |
|-----|--------------|---------------|-----------------|
| PC0 | 192.168.1.12 | 255.255.255.0 | 192.168.1.1 |

| | | | |
|-----|--------------|---------------|-------------|
| PC1 | 192.168.1.14 | 255.255.255.0 | 192.168.1.1 |
| PC2 | 192.168.2.12 | 255.255.255.0 | 192.168.2.1 |
| PC3 | 192.168.2.14 | 255.255.255.0 | 192.168.2.1 |



PC0

Physical Config Desktop Custom Interface

IP Configuration

IP Configuration

☐ DHCP ☒ Static

IP Address: 192.168.1.12

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.1.1

DNS Server:

IPv6 Configuration

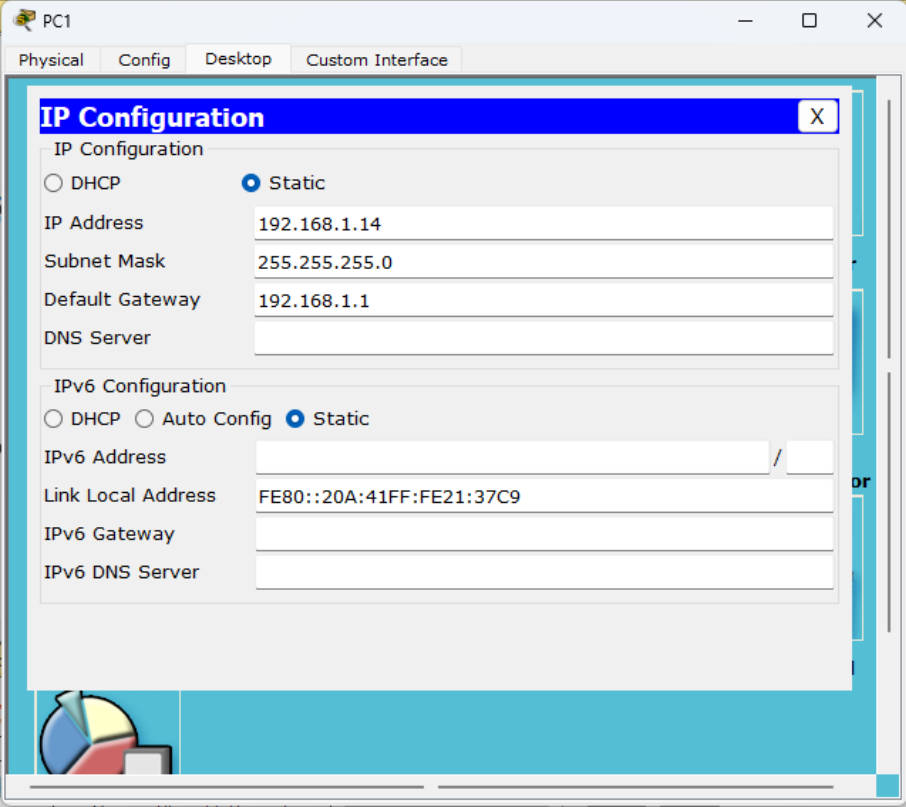
☐ DHCP ☐ Auto Config ☒ Static

IPv6 Address: /

Link Local Address: FE80::2D0:BCFF:FEA1:A71E

IPv6 Gateway:

IPv6 DNS Server:



PC1

Physical Config Desktop Custom Interface

IP Configuration

IP Configuration

☐ DHCP ☒ Static

IP Address: 192.168.1.14

Subnet Mask: 255.255.255.0

Default Gateway: 192.168.1.1

DNS Server:

IPv6 Configuration

☐ DHCP ☐ Auto Config ☒ Static

IPv6 Address: /

Link Local Address: FE80::20A:41FF:FE21:37C9

IPv6 Gateway:

IPv6 DNS Server:

PC2

Physical Config Desktop Custom Interface

IP Configuration

IP Configuration

☐ DHCP ☒ Static

IP Address 192.168.2.12

Subnet Mask 255.255.255.0

Default Gateway 192.168.2.1

DNS Server

IPv6 Configuration

☐ DHCP ☐ Auto Config ☒ Static

IPv6 Address /

Link Local Address FE80::2D0:BCFF:FE42:870

IPv6 Gateway

IPv6 DNS Server

PC3

Physical Config Desktop Custom Interface

IP Configuration

IP Configuration

☐ DHCP ☒ Static

IP Address 192.168.2.14

Subnet Mask 255.255.255.0

Default Gateway 192.168.2.1

DNS Server

IPv6 Configuration

☐ DHCP ☐ Auto Config ☒ Static

IPv6 Address /

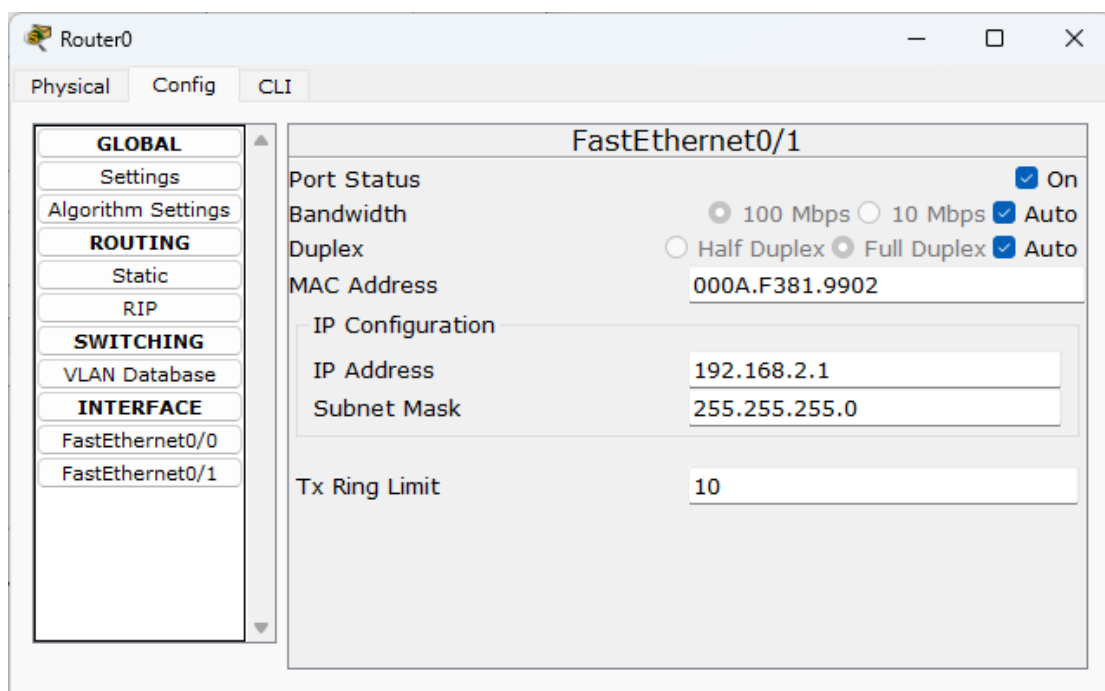
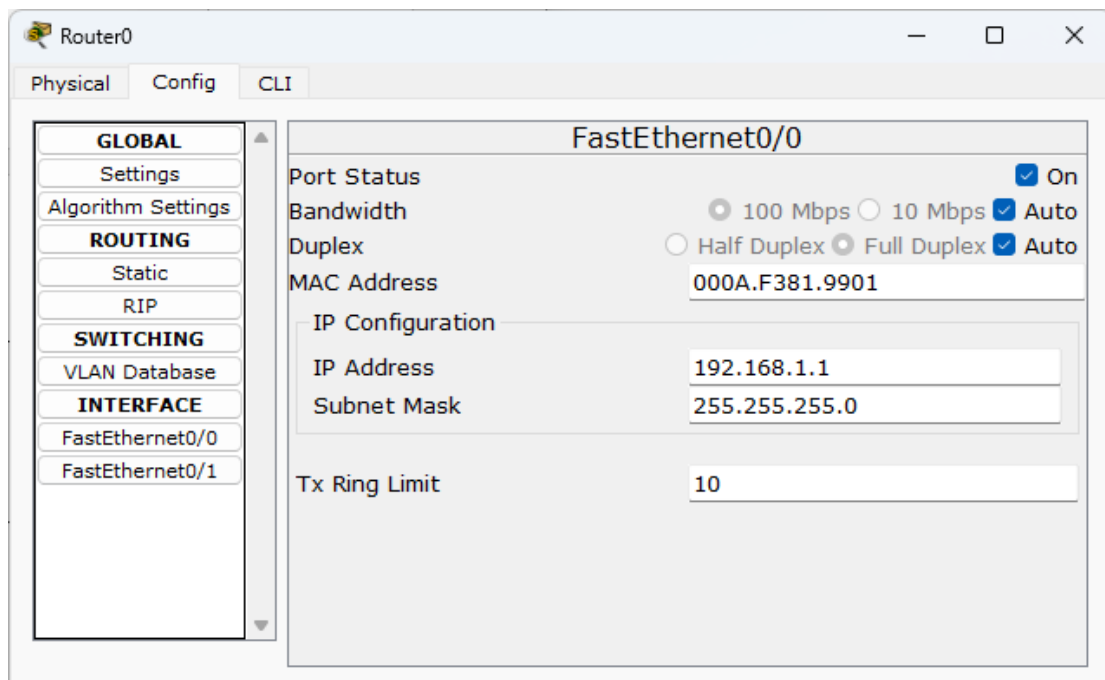
Link Local Address FE80::201:97FF:FE77:E26E

IPv6 Gateway

IPv6 DNS Server

4. 参照先前实验（实验 18：动态 IP 地址分配 DHCP 实验）配置路由器 Router0 的接口，可以通过在 CLI 中输入以下命令进行配置，也可以通过图形化界面进行配置。

```
interface FastEthernet0/0
ip address 192.168.1.1 255.255.255.0
interface FastEthernet0/1
ip address 192.168.2.1 255.255.255.0
```



5. 在路由器 Router0 配置 DHCP。

- 在 CLI 输入以下命令配置路由器 DHCP 左边网络。

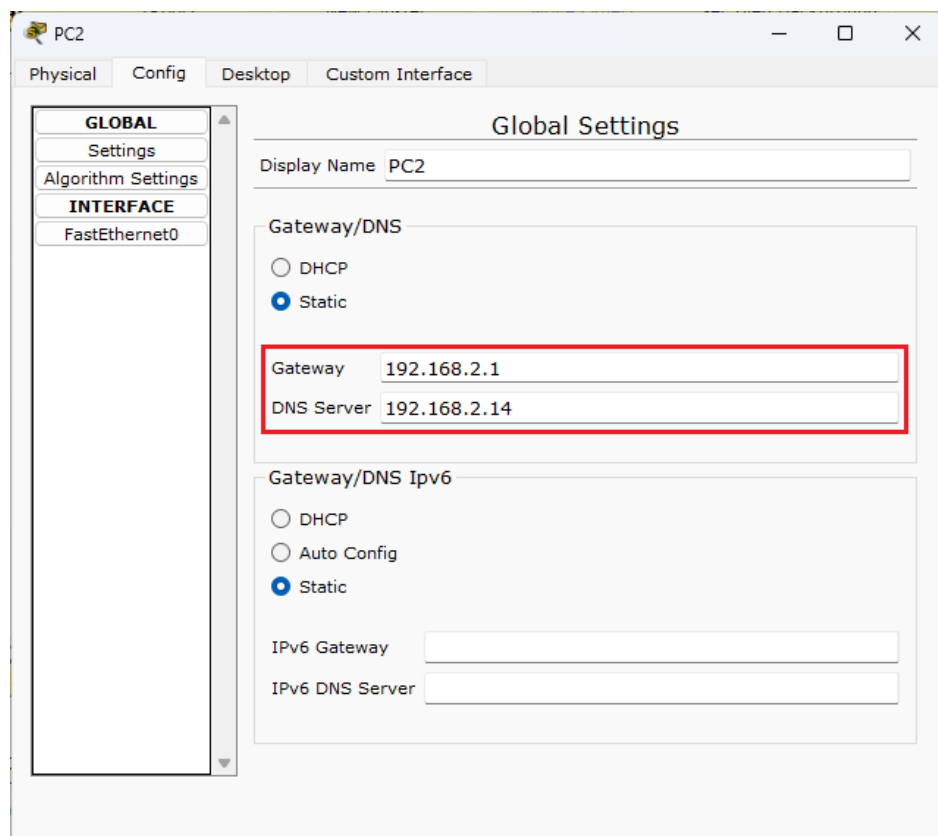
```
ip dhcp excluded-address 192.168.1.0 192.168.1.10
ip dhcp pool myleftnet
network 192.168.1.0 255.255.255.0
default-router 192.168.1.1
option 150 ip 192.168.1.3
dns-server 192.168.1.2
```

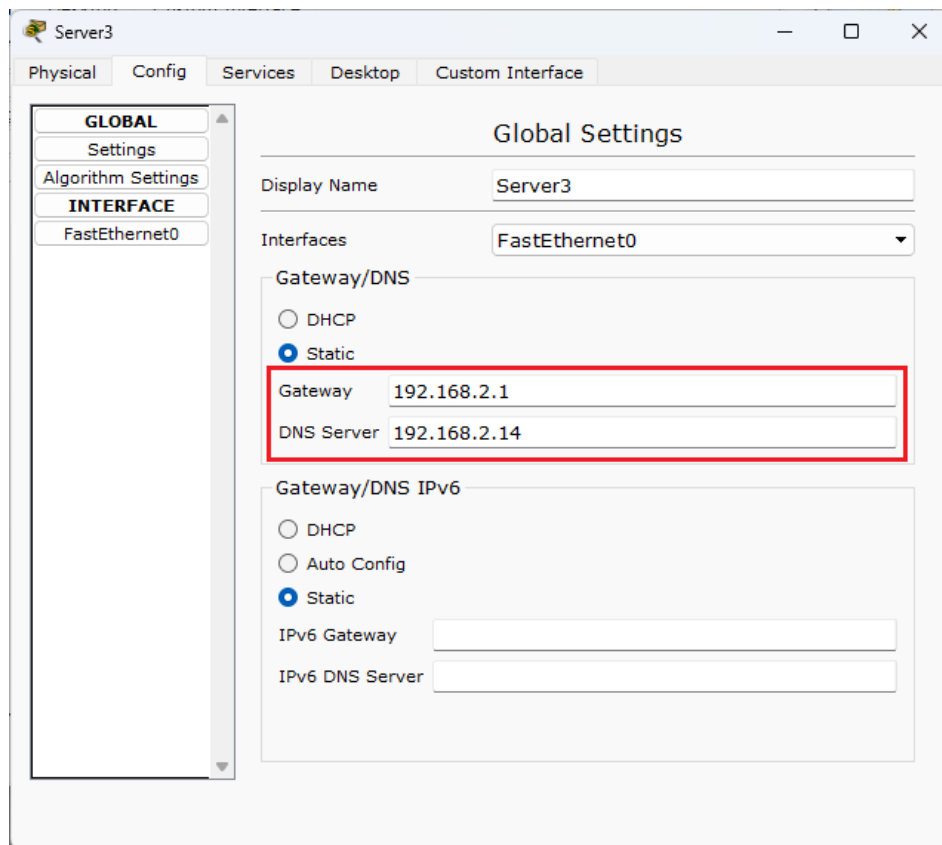
- 在 CLI 输入以下命令配置路由器 DHCP 右边网络。

```
ip dhcp excluded-address 192.168.2.0 192.168.2.10
ip dhcp pool myrightnet
network 192.168.2.0 255.255.255.0
default-router 192.168.2.1
option 150 ip 192.168.2.3
dns-server 192.168.2.2
```

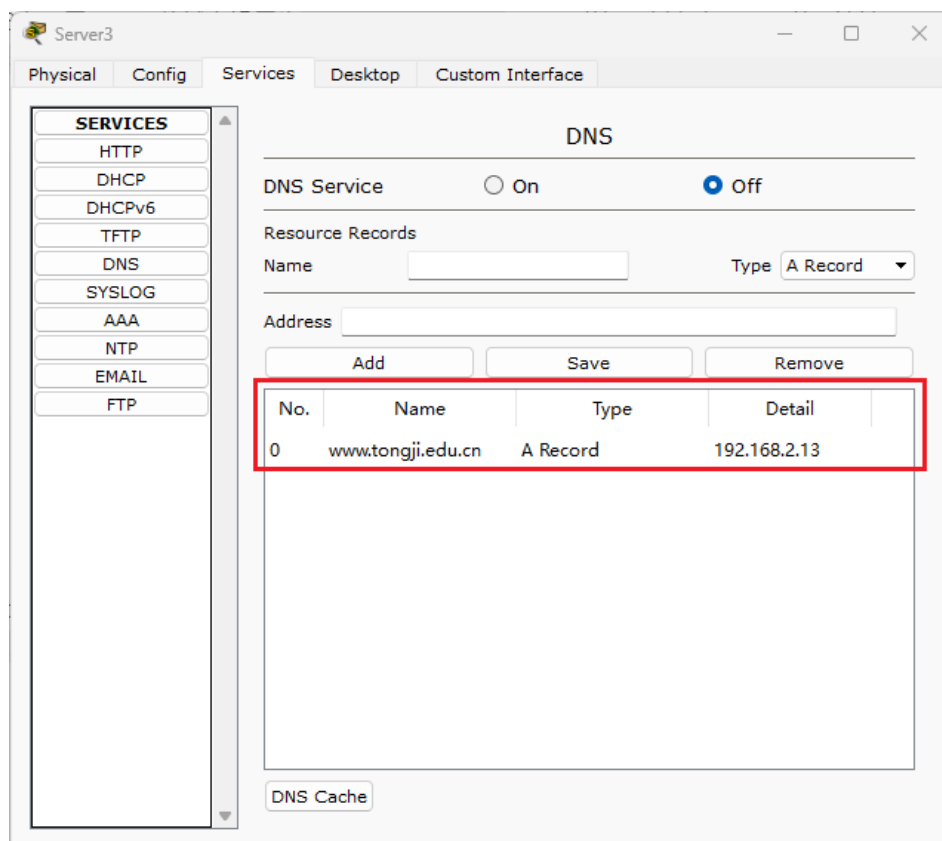
6. 配置 Server 的 Gateway 和 DNS Server。

7. 配置 PC2 的 DNS Server 为 192.168.2.14。设置 PC2 的 DNS Server 与 Server3 的 DNS Server 相同是为了能够通过 Server3 构建映射访问到 Server2。

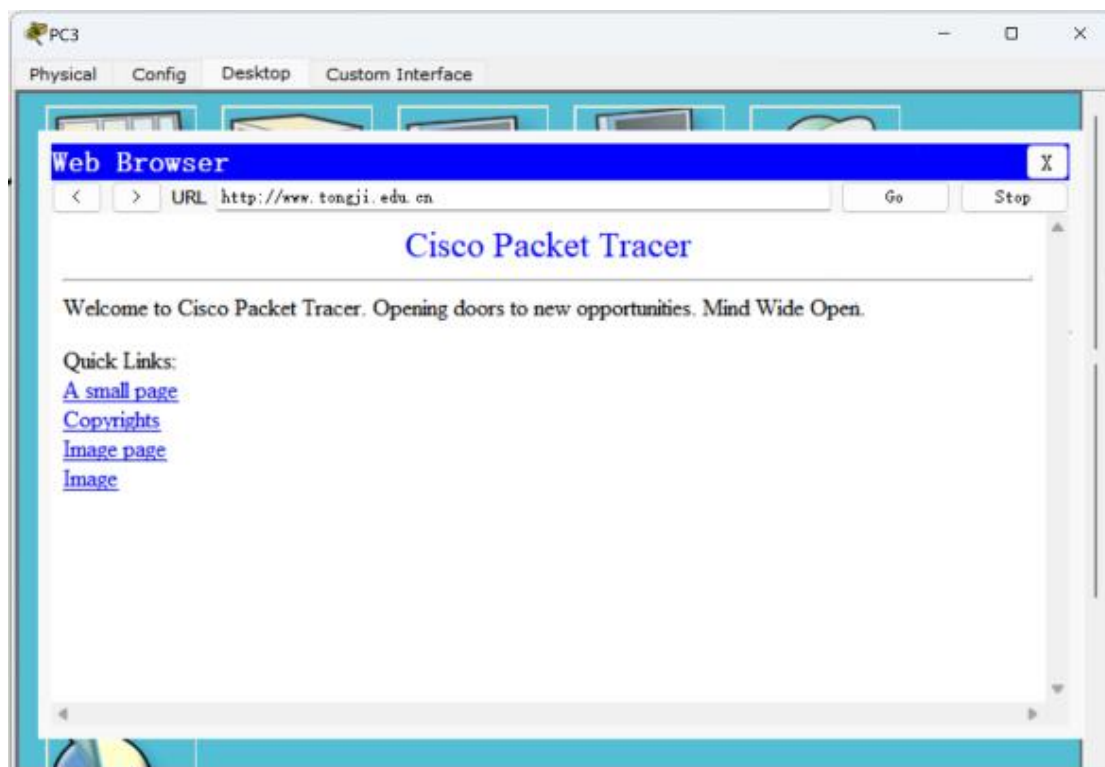




8. 配置 Server3 的 Services, Name 设置为 www.tongji.edu.cn, Detail 设置为 Server2 的 IP 地址 (192.168.2.13)。



9. 从 Realtime 模式切换至 Simulation 模式。
 - 9.1 在 PC3 的 Web Browser 中输入 `http://www.tongji.edu.cn`, 产生 DNS 数据报文, 分析 DNS 报文情况。
 - 9.2 点击 Capture/Forward 单步执行, 也可以点击 Auto Capture/Play 自动执行, 查看相关数据。
 - 9.3 在 Even List 中的 Info 栏可以查看相关信息。



10. 分析在 Packet Tracer 中 DNS 报文情况。
11. 用 WireShark 抓取 DNS 数据包, 查看 DNS 报文字段内容, 并解读。

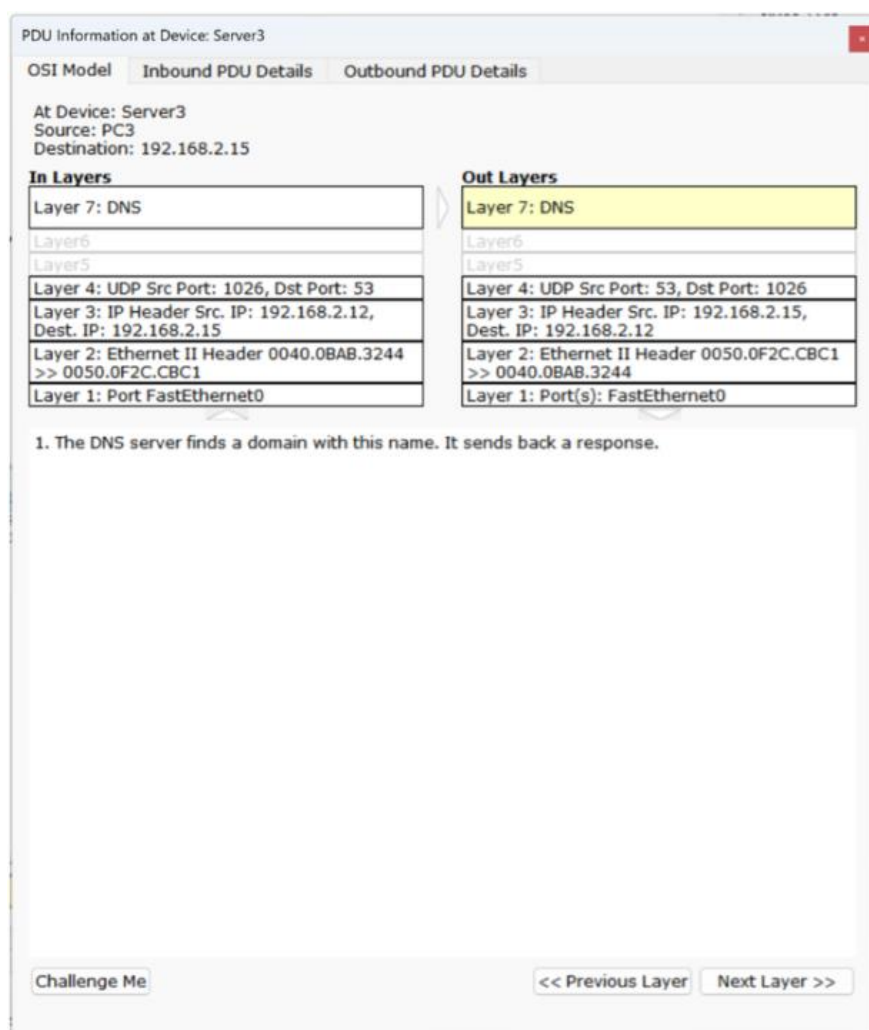
【实验现象】

1. 分析在 Packet Tracer 中 DNS 报文情况。
 - 1.1 PDU Information at Device: Server3 (OSI Model)

在这张 Packet Tracer 的截图中, 我们可以看到 DNS 查询和响应的详细过程, 涵盖了各层的信息。这个过程涉及从一个客户端(PC3)到 DNS 服务器(Server3)的 DNS 查询, 以及服务器对该查询的响应。

 - Inbound PDU Details (入站协议数据单元细节)
 - 源设备: PC3
 - 目的设备: Server3

- 源 IP 地址：192.168.2.12
- 目的 IP 地址：192.168.2.15
- 各层分析：
 - Layer 1（物理层）：Port FastEthernet0
 - Layer 2（数据链路层）：使用 Ethernet II 帧，源 MAC 地址为 0040.0BAB.3244，目的 MAC 地址为 0050.0F2C.CBC1。
 - Layer 3（网络层）：使用 IP 协议，源 IP 地址为 192.168.2.12，目的 IP 地址为 192.168.2.15。
 - Layer 4（传输层）：使用 UDP 协议，源端口 1026，目的端口 53（DNS 服务的标准端口）。
 - Layer 7（应用层）：DNS 层，此处未提供具体的 DNS 查询细节，但通常会包含所查询的域名等信息。



- Outbound PDU Details（出站协议数据单元细节）

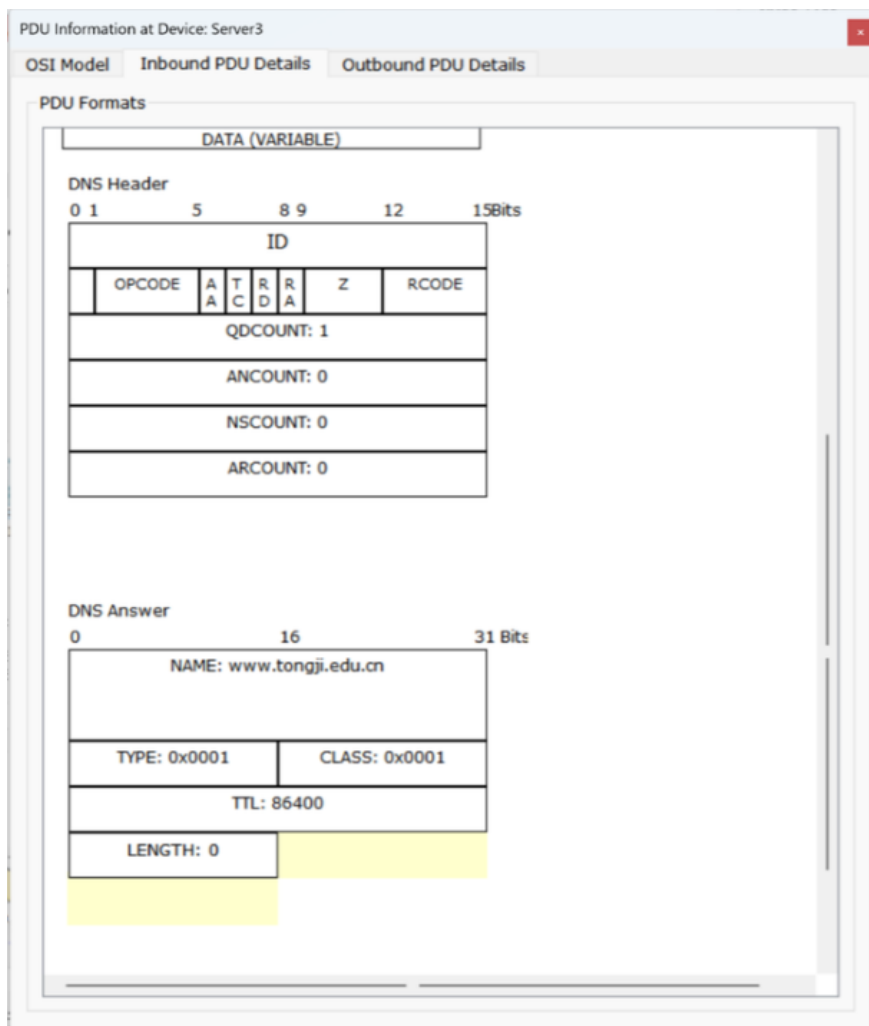
- 源设备: Server3
- 目的设备: PC3
- 源 IP 地址: 192.168.2.15
- 目的 IP 地址: 192.168.2.12
- 各层分析:
 - Layer 1 (物理层): Port FastEthernet0
 - Layer 2 (数据链路层): 使用 Ethernet II 帧, 源 MAC 地址为 0050.0F2C.CBC1, 目的 MAC 地址为 0040.0BAB.3244。
 - Layer 3 (网络层): 使用 IP 协议, 源 IP 地址为 192.168.2.15, 目的 IP 地址为 192.168.2.12。
 - Layer 4 (传输层): 使用 UDP 协议, 源端口 53, 目的端口 1026。
 - Layer 7 (应用层): DNS 层, 说明 DNS 服务器已找到对应的域名, 并发回响应。常见的 DNS 响应会包含请求的域名、解析到的 IP 地址、记录类型 (如 A 记录, AAAA 记录等)、TTL 值等。

1.2 PDU Information at Device: Server3 (Inbound PDU Details)

在这张图中,我们看到的是 Packet Tracer 模拟环境中一个 DNS 查询响应的详细 PDU (协议数据单元) 信息。这个响应显示在设备 Server3 上的出站 PDU 细节。

- DNS 头部
 - ID: 事务 ID, 用于匹配请求和响应。
 - OPCODE: 操作码, 用于指示查询的类型, 这里默认为 0, 表示标准查询。
 - AA (Authoritative Answer): 授权回答, 此处应为 0, 表示响应不是来自权威源。
 - TC (Truncated): 截断, 标识消息是否被截断。
 - RD (Recursion Desired): 期望递归, 请求希望得到一个递归响应。
 - RA (Recursion Available): 递归可用, 服务器可以进行递归查询。
 - Z: 保留位, 必须为 0。
 - RCODE: 响应码, 指示查询的状态, 此处应为 0, 表示无错误。

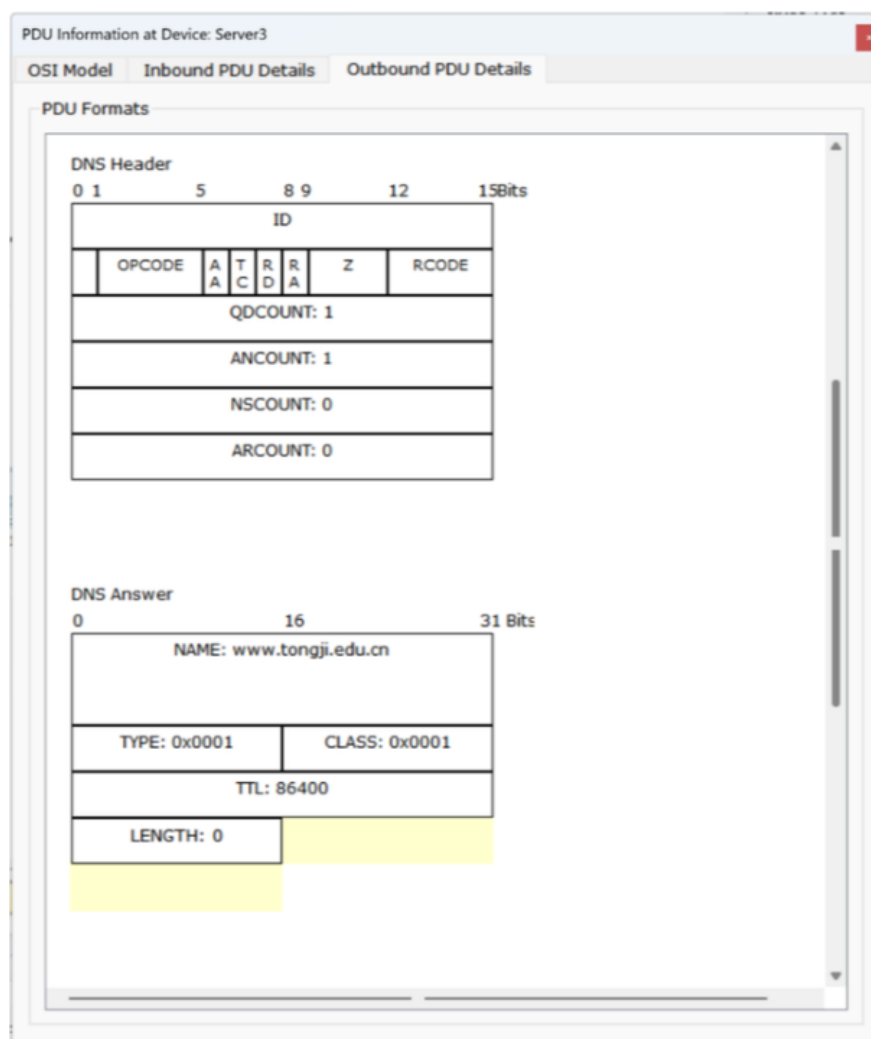
- QDCOUNT: 查询数量, 这里为 1, 表示一个查询请求。
- ANCOUNT: 应答数量, 此处为 0, 表示没有应答资源记录。
- NSCOUNT: 授权名称服务器计数, 此处为 0。
- ARCOUNT: 附加记录计数, 此处也为 0。



● DNS 应答部分

- NAME: 请求的域名, www.tongji.edu.cn。
- TYPE: 查询类型, 这里为 0x0001, 即 A 记录, 请求 IPv4 地址。
- CLASS: 类别, 这里为 0x0001, 即 IN, 表示互联网。
- TTL (Time To Live): 资源记录可以缓存的时间, 这里为 86400 秒 (24 小时)。
- LENGTH: 资源数据的长度, 这里为 0, 表示响应中没有包含实际的 IP 地址。这通常表明 DNS 查询没有成功找到请求的域名对应的 IP 地址, 或者查询的记录不存在。

1.3 PDU Information at Device: Server3 (Outbound PDU Details)



这张图展示的是在 Packet Tracer 中的一个 DNS 查询的响应。图中详细显示了 DNS 查询响应报文的结构，包括 DNS 头部和应答部分的具体内容。

- DNS 头部
 - ID: 用于匹配请求和响应的标识符。
 - Opcode: 操作码，用于指示 DNS 查询的类型，这里为 0，表示标准查询。
 - AA (Authoritative Answer): 授权回答标志，这里没有具体显示是否设置，但通常用于表明响应是否来自拥有此记录权威的 DNS 服务器。
 - TC (Truncated): 截断标志，用于表明消息是否因过长而被截断，这里没有设置。
 - RD (Recursion Desired): 期望递归标志，通常在查询时设置，希

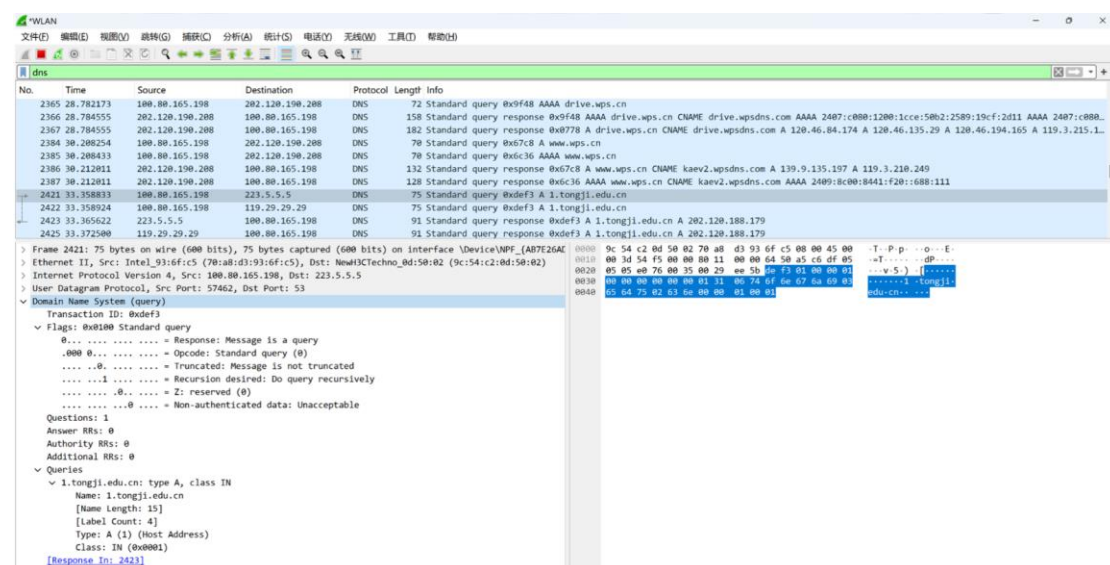
望 DNS 服务器进行完整的递归查询。

- RA (Recursion Available): 递归可用标志, 表示服务器是否支持递归查询, 这里没有设置。
- Z: 保留字段, 必须为 0。
- Rcode: 响应代码, 这里为 0, 表示没有错误。
- QDCOUNT: 查询计数, 这里为 1, 表示有一个查询请求。
- ANCOUNT: 应答计数, 这里为 1, 表示有一个应答。
- NSCOUNT: 授权名称服务器计数, 这里为 0。
- ARCOUNT: 附加记录计数, 这里也为 0。

● DNS 应答部分

- NAME: 请求解析的域名, 这里为 www.tongji.edu.cn。
- TYPE: 请求的类型, 这里为 0x0001, 表示是 A 记录查询, 用于请求 IPv4 地址。
- CLASS: 类别, 这里为 0x0001, 通常表示 IN 类, 即 Internet。
- TTL (Time To Live): DNS 记录的生存时间, 这里为 86400 秒, 表示该记录可以在 DNS 缓存中保留 24 小时。
- LENGTH: 数据长度, 这里为 0, 这通常意味着在响应中没有包含 IP 地址, 可能是因为查找失败或请求的记录在 DNS 服务器上不存在。

2. 用 Wireshark 抓取 DNS 数据包, 查看 DNS 报文字段内容, 并解读。



The screenshot shows a Wireshark capture of a DNS query packet. The packet list at the top shows a query for www.tongji.edu.cn. The packet details pane shows the query structure: Transaction ID: 0xdef3, Flags: 0x0000, Questions: 1. The packet bytes pane shows the raw data of the query.

在这个 Wireshark 截图中, 我们可以看到一个 DNS 查询过程的详细解析。这

份报文捕获展示了从某个设备对域名 1.tongji.edu.cn 进行 DNS 查询的详细过程。下面是报文字段详解：

- 事务 ID (Transaction ID): 0xd9ef3, 这是由客户端生成的唯一 ID, 用于匹配请求和响应。确保响应与请求对应。
- 标志 (Flags): 0x0100, 这是 DNS 报文的标志字段, 具体细分如下:
 - QR (Query/Response): 0, 表明这是一个查询请求。
 - Opcode: 0, 标准查询。
 - AA (Authoritative Answer): 0, 表明回答不是由权威 DNS 服务器直接提供。
 - TC (Truncated): 0, 消息没有被截断。
 - RD (Recursion Desired): 1, 客户端希望 DNS 服务器递归查询。
 - RA (Recursion Available): 0, 响应中不显示是否支持递归。
- 计数器
 - 问题数 (Questions): 1, 指明了查询中包含的问题数量。
 - 回答资源记录数 (Answer RRs): 0, 响应中没有包含答案资源记录。
 - 授权资源记录数 (Authority RRs): 0, 响应中没有权威 DNS 服务器的信息。
 - 附加资源记录数 (Additional RRs): 0, 响应中没有附加资源记录。
- 查询 (Queries)
 - 名称 (Name): 查询的域名是 1.tongji.edu.cn。
 - 类型 (Type): A (1), 请求的是 IPv4 地址的解析。
 - 类 (Class): IN (0x0001), 表示查询属于互联网类别。
- 响应编号 (Response In): 2423, 指向 Wireshark 捕获中的具体一帧 (2423 号帧), 在该帧中可以找到这个查询的响应。这表明响应与此查询相关联, 可以在该帧中查看详细的响应信息。

这个 Wireshark 截图提供了关于一个 DNS 查询的详细技术视图。通过查看 DNS 查询和响应的详细信息能够诊断网络问题, 如域名解析失败或延迟等。通过响应编号, 我们可以进一步分析响应帧来获取更多关于未能解析域名的原因。

【分析讨论】

一、分析在 Packet Tracer 中 DNS 报文情况

在 Packet Tracer 的模拟环境中，DNS 报文的分析揭示了从客户端到 DNS 服务器的详细通信过程。通过配置和实施 DNS 查询，我们能够观察到以下现象：

- **网络配置：**通过设定静态 IP 地址、子网掩码、默认网关以及 DNS 服务器，确保了网络内部各设备能够正确交换信息。
- **DNS 查询过程：**DNS 查询的捕获显示，客户端（如 PC3）向 DNS 服务器（如 Server3）发起请求，查询特定域名（例如 `www.tongji.edu.cn`）的 IP 地址。
- **DNS 响应行为：**服务器对 DNS 查询的响应揭示了是否成功找到相应的域名记录。在特定配置下，可以看到服务器是否能够返回正确的 IP 地址或者是否发生查询失败（如响应中的 LENGTH 字段为 0 时）。

此部分实验强调了 DNS 配置的正确性对网络服务可达性的重要性，同时也展示了在模拟环境中如何调试和解决网络配置问题。

二、用 WireShark 抓取 DNS 数据包并查看 DNS 报文字段内容

使用 WireShark 进行实时数据捕获进一步深化了对 DNS 查询机制的理解，具体包括：

- **详细的头部信息分析：**事务 ID、Flags 和其他 DNS 协议特定字段的观察，提供了查询和响应匹配的方式，确保了数据通信的连贯性。
- **查询响应过程：**WireShark 的捕获数据显示，查询如 `1.tongji.edu.cn` 的 DNS 请求，及其对应的响应或缺乏响应（例如响应中答案资源记录为 0），揭示了可能的配置错误或记录缺失。
- **故障诊断：**分析响应编号如 2423 号帧，可以具体看到 DNS 响应的内容，进而判断 DNS 服务的健康状况以及可能的配置问题。

这一部分实验强调了实际网络环境中 DNS 查询的重要性，并展示了如何使用专业工具进行网络通信的故障诊断和性能分析。