实验(二十二)：UDP分段实验

一.实验目的

- 该实验旨在通过分析UDP（用户数据报协议）数据包，深入理解UDP协议的工作机制及其应用特点。具体目标包括：
  - **配置和查看Web服务器**：配置Web服务器并从客户端进行访问，生成UDP数据包。
  - **配置DNS服务器**：配置一个简单的DNS服务器，并使用客户端进行查询以生成UDP数据包。
  - **分析UDP报文**：使用`Packet Tracer`工具分析生成的UDP报文，了解其传输过程及特点。
  - **抓取和分析UDP数据包**：使用`WireShark`工具抓取UDP数据包，查看和解读其报文字段内容。

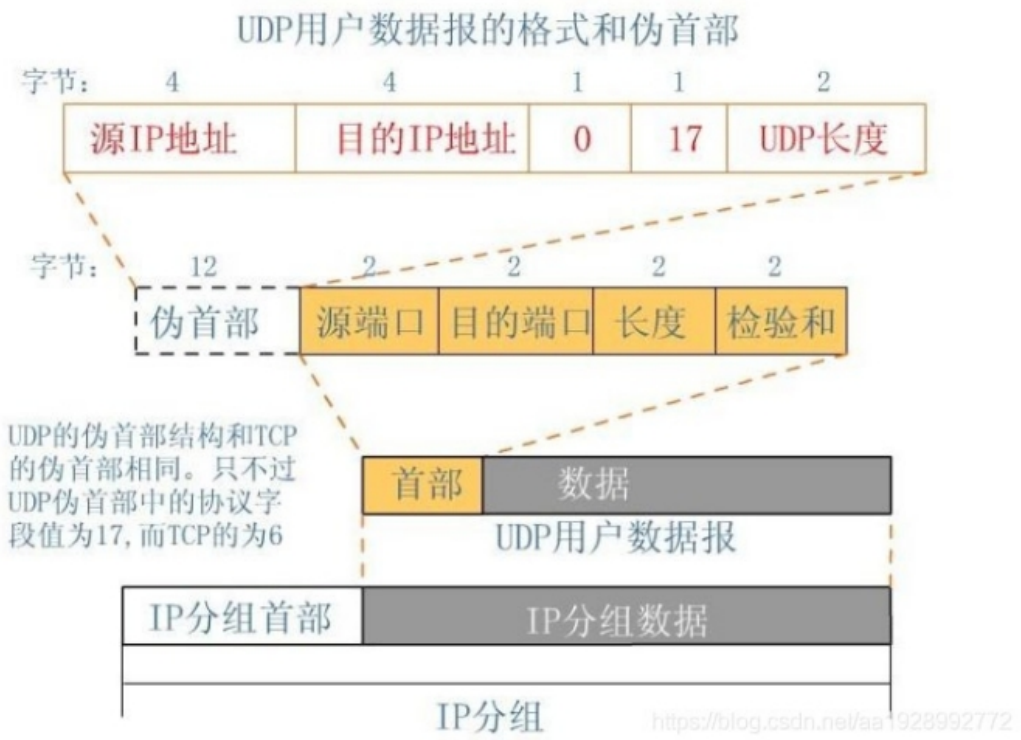
二.实验原理

- UDP概述
  - UDP是传输层的协议，功能即为在IP的数据报服务之上增加了最基本的服务：复用和分用以及差错检测。
  - UDP提供不可靠服务，具有TCP所没有的优势：
  - UDP无连接，时间上不存在建立连接需要的时延。空间上，TCP需要在端系统中维护连接状态，需要一定的开销。此连接装入包括接收和发送缓存，拥塞控制参数和序号与确认号的参数。UCP不维护连接状态，也不跟踪这些参数，开销小。空间和时间上都具有优势。
- UDP应用特点
  - DNS如果运行在TCP之上而不是UDP，那么DNS的速度将会慢很多。HTTP使用TCP而不是UDP，是因为对于基于文本数据的Web网页来说，可靠性很重要。同一种专用应用服务器在支持UDP时，一定能支持更多的活动客户机。分组首部开销小，TCP首部20字节，UDP首部8字节。
  - UDP没有拥塞控制，应用层能够更好的控制要发送的数据和发送时间，网络中的拥塞控制也不会影响主机的发送速率。某些实时应用要求以稳定的速度发送，能容 忍一些数据的丢失，但是不能允许有较大的时延（比如实时视频，直播等）
  - UDP提供尽最大努力的交付，不保证可靠交付。所有维护传输可靠性的工作需要用户在应用层来完成。没有TCP的确认机制、重传机制。如果因为网络原因没有传送到对端，UDP也不会给应用层返回错误信息。
  - UDP是面向报文的，对应用层交下来的报文，添加首部后直接乡下交付为IP层，既不合并，也不拆分，保留这些报文的边界。对IP层交上来UDP用户数据报，在去除首部后就原封不动地交付给上层应用进程，报文不可分割，是UDP数据报处理的最小单位。正是如此UDP显得不够灵活，不能控制读写数据的次数和数量。比如我们要发送100个字节的报文，调用一次sendto函数就会发送100字节，对端也需要用recvfrom函数一次性接收100字节，不能使用循环每次获取10个字节，获取十次这样的做法。
  - UDP常用一次性传输比较少量数据的网络应用，如DNS,SNMP等，因为对于这些应用，若是采用TCP，为连接的创建，维护和拆除带来不小的开销。UDP也常用于多媒体应用（如IP电话，实时视频会议，流媒体等）数据的可靠传输对他们而言并不重要，TCP的拥塞控制会使它们有较大的延迟，也是不可容忍的。总之，UDP协议提供不可靠无连接的数据报传输服务。
- UDP报文格式

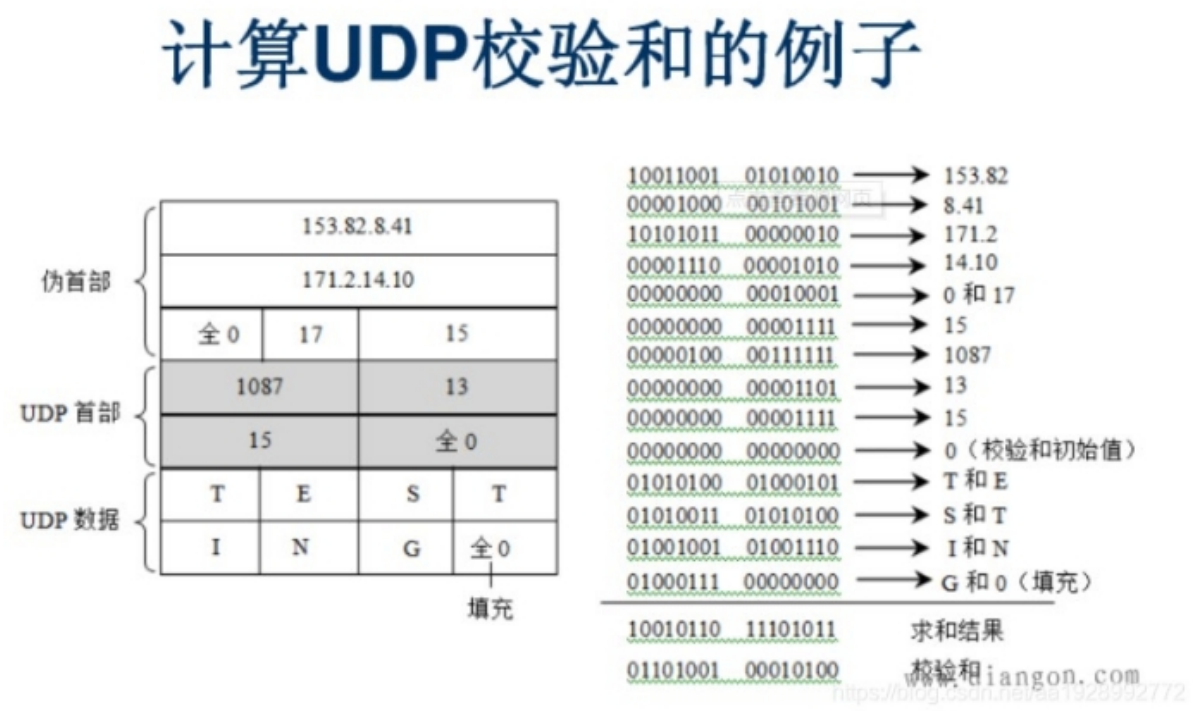
- UDP的首部格式
  - UDP数据报分为首部和用户数据部分，整个UDP数据报作为IP数据报的数据部分封装在IP数据报中，UDP数据报文结构如图所示：

  - UDP首部有8个字节，由4个字段构成，每个字段都是两个字节
    - 源端口： 源端口号，需要对方回信时选用，不需要时全部置0.
    - 目的端口：目的端口号，在终点交付报文的时候需要用到。
    - 长度：UDP的数据报的长度（包括首部和数据）其最小值为8（只有首部）
    - 校验和：检测UDP数据报在传输中是否有错，有错则丢弃。该字段是可选的，当源主机不想计算校验和，则直接令该字段全为0。当传输层从IP层收到UDP数据报时，就根据首部中的目的端口，把UDP数据报通过相应的端口，上交给应用进程。如果接收方UDP发现收到的报文中的目的端口号不正确（不存在对应端口号的应用进程0,），就丢弃该报文，并由ICMP发送“端口不可达”差错报文给对方。

- UDP校验
  - 在计算校验和的时候，需要在UDP数据报之前增加12字节的伪首部，伪首部并不是UDP真正的首部。只是在计算校验和，临时添加在UDP数据报的前面，得到一个临时的UDP数据报。校验和就是按照这个临时的UDP数据报计算的。伪首部既不向下传送也不向上递交，而仅仅是为了计算校验和。这样的校验和，既检查了UDP数据报，又对IP数据报的源IP地址和目的IP地址进行了检验。
  - UDP校验和的计算方法和IP数据报首部校验和的计算方法相似，都使用二进制反码运算求和再取反，但不同的是：IP数据报的校验和只检验IP数据报的首部，但UDP的校验和是把首部和数据部分一起校验



- 发送方，首先是把全零放入校验和字段并且添加伪首部，然后把UDP数据报看成是由许多16位的子串连接起来，若UDP数据报的数据部分不是偶数个字节，则要在数据部分末尾增加一个全零字节（此字节不发送），接下来就按照二进制反码计算出这些16位字的和。将此和的二进制反码写入校验和字段。在接收方，把收到得UDP数据报加上伪首部（如果不为偶数个字节，还需要补上全零字节）后，按二进制反码计算出这些16位字的和。
- 当无差错时其结果全为1,。否则就表明有差错出现，接收方应该丢弃这个UDP数据报。注意：
  - 校验时，若UDP数据报部分的长度不是偶数个字节，则需要填入一个全0字节，但是此字节和伪首部一样，是不发送的。
  - 如果UDP校验和校验出UDP数据报是错误的，可以丢弃，也可以交付上层，但是要附上错误报告，告诉上层这是错误的数据报。
  - 通过伪首部，不仅可以检查源端口号，目的端口号和UDP用户数据报的数据部分，还可以检查IP数据报的源IP地址和目的地址。这种差错检验的检错能力不强，但是简单，速度快。

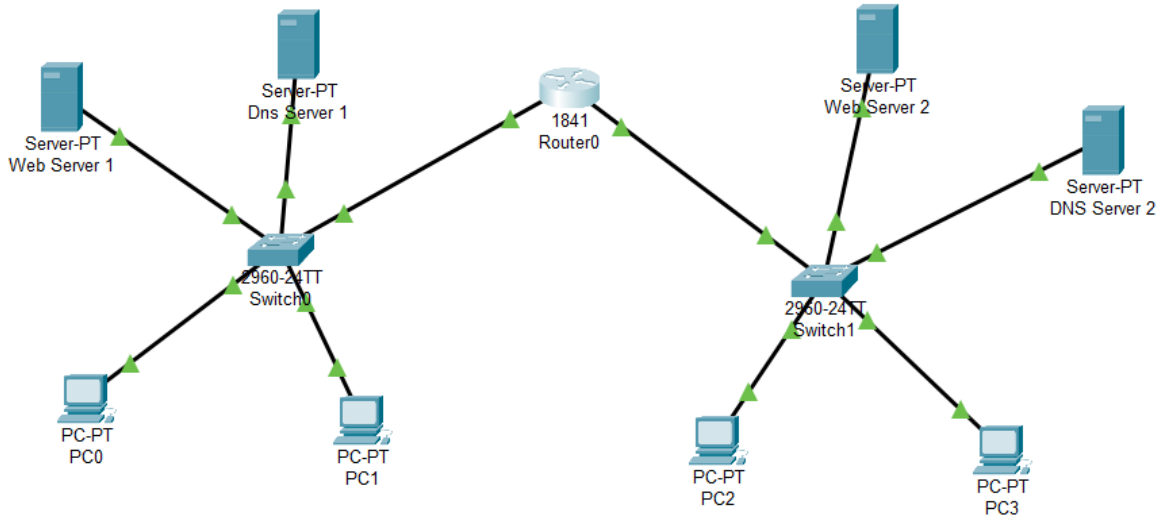


三.实验环境

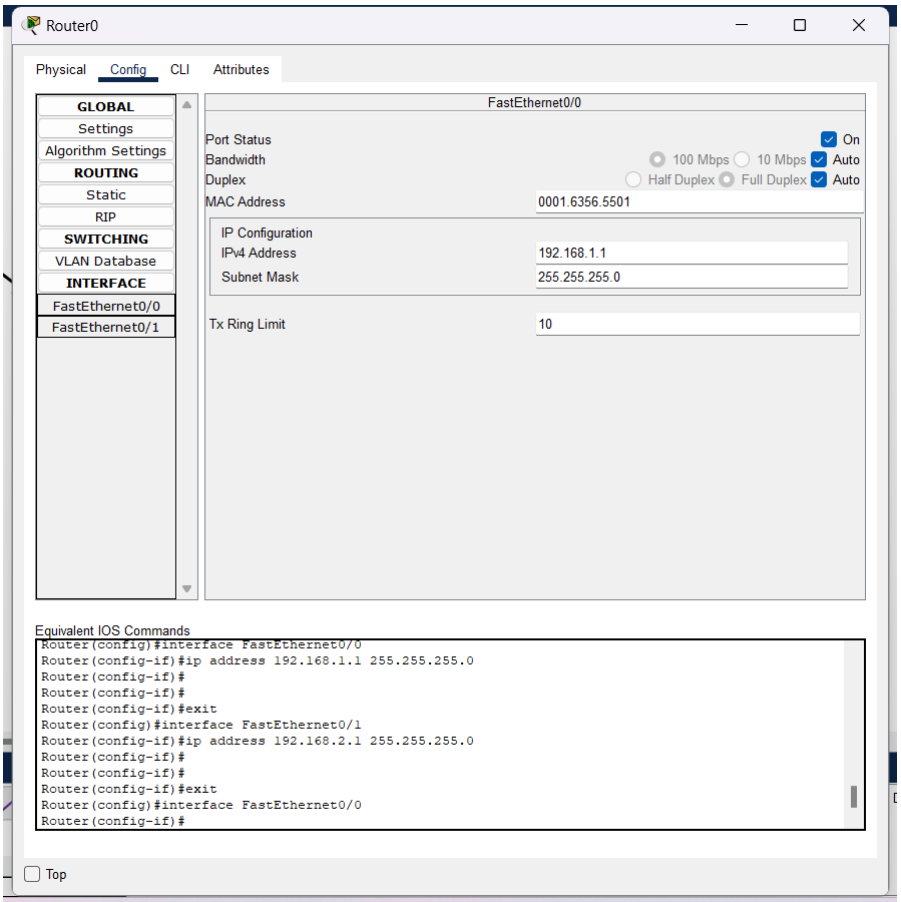
- 操作系统：Windows 11
- 网络环境：局域网
- 软件：Cisco Packet Tracer虚拟实验环境

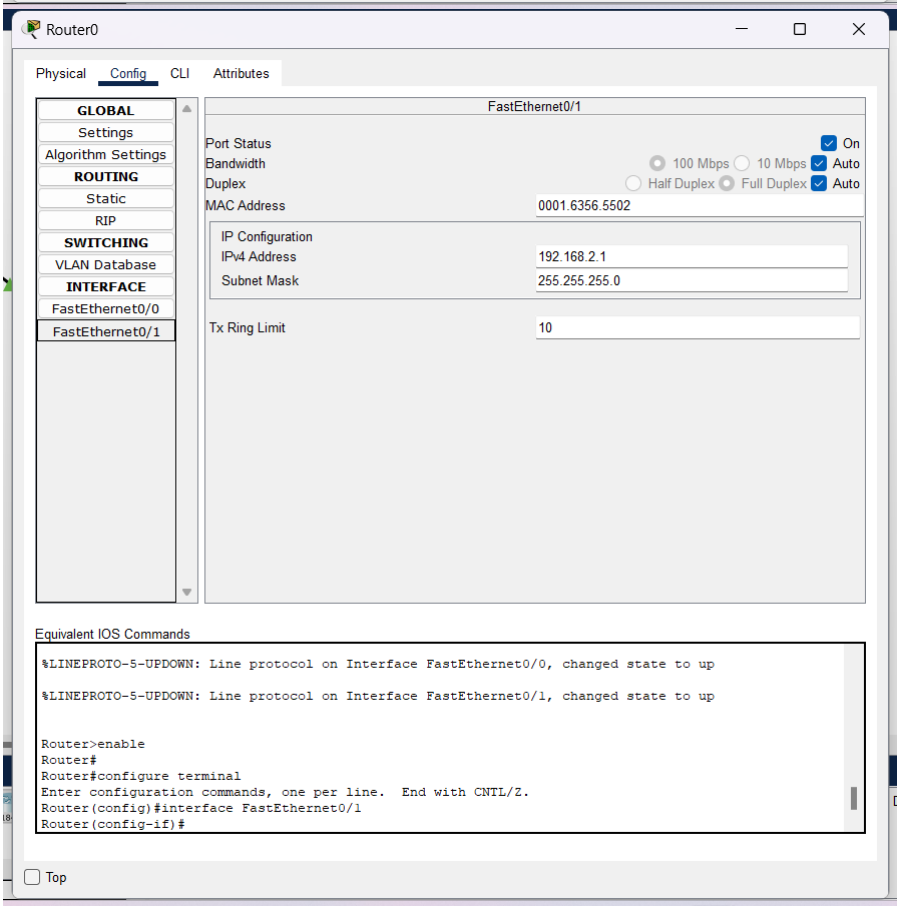
四.实验步骤

- 规划网络地址及拓扑图如下图所示

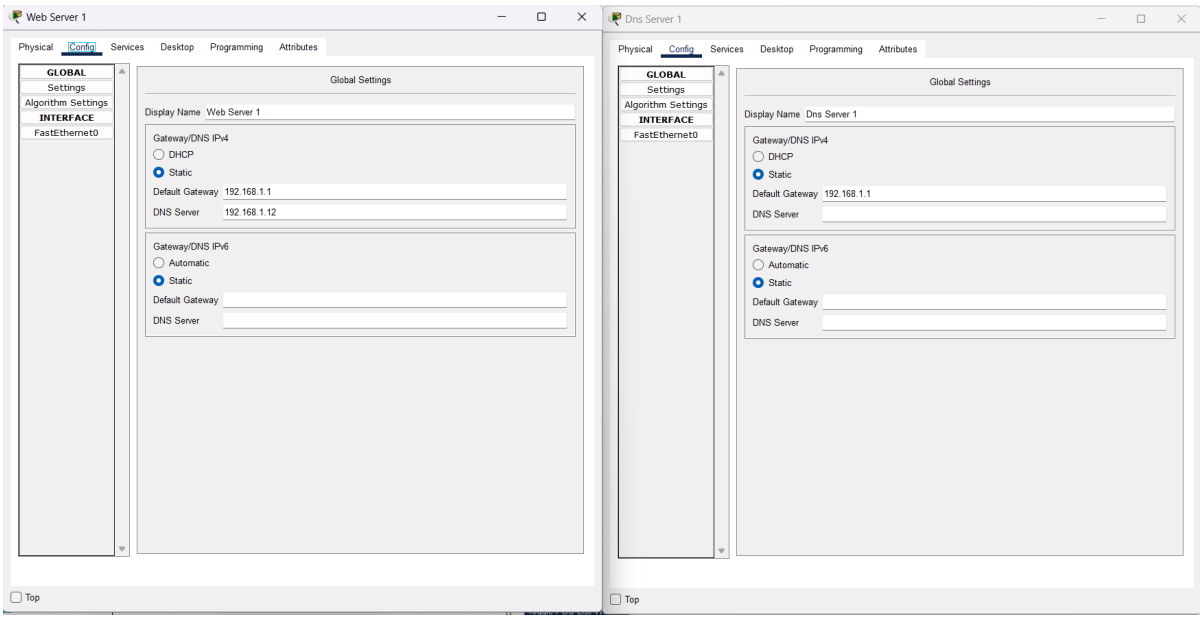


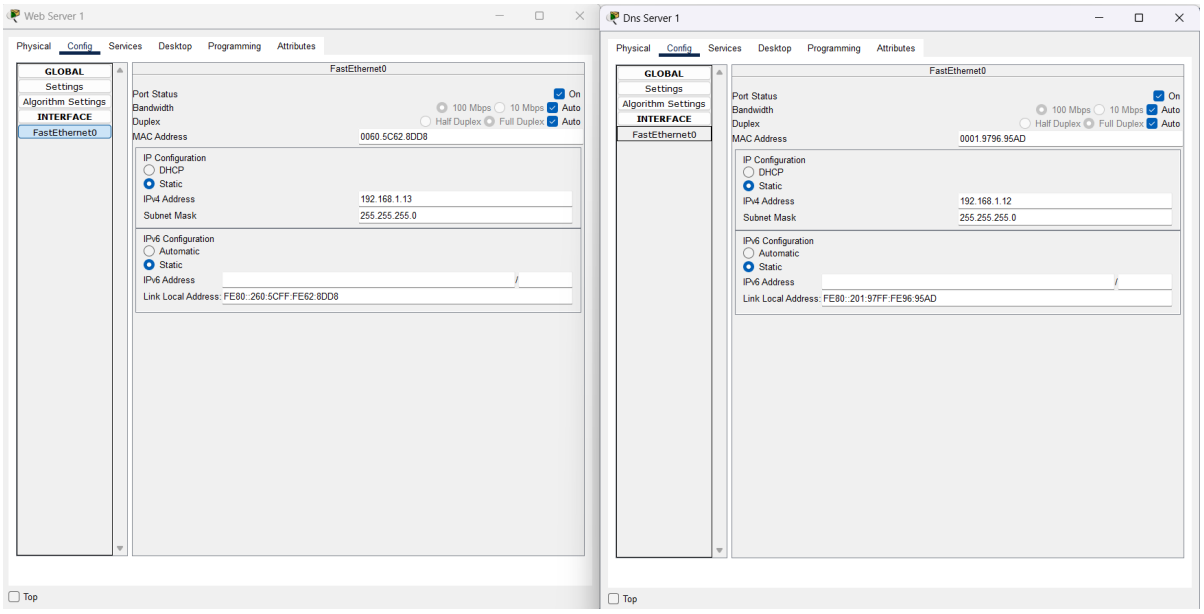
- 配置Router0（以左侧一组为例，右侧同理）



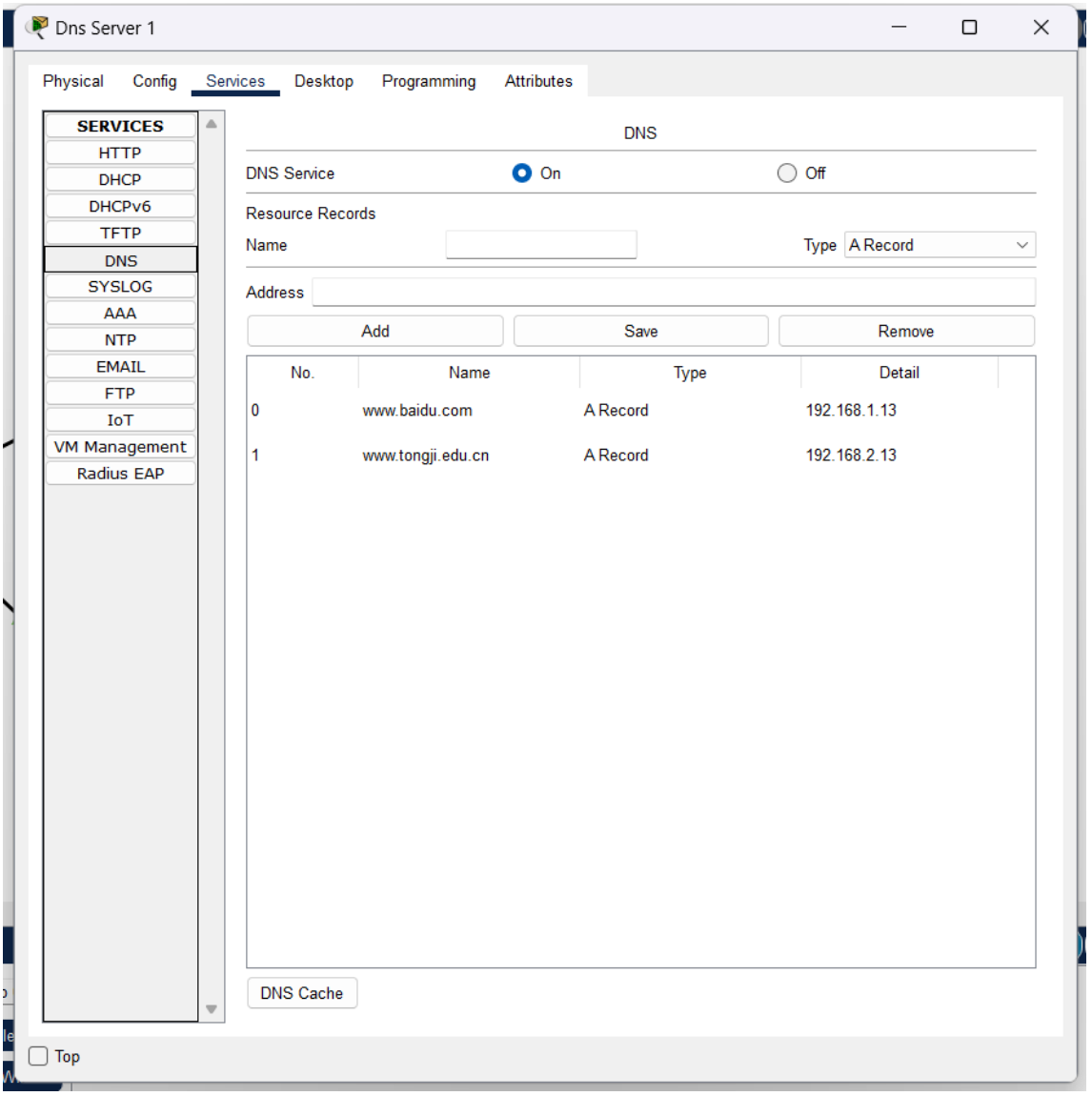


- 配置Web Server和DNS Server（以左侧一组为例，右侧同理）

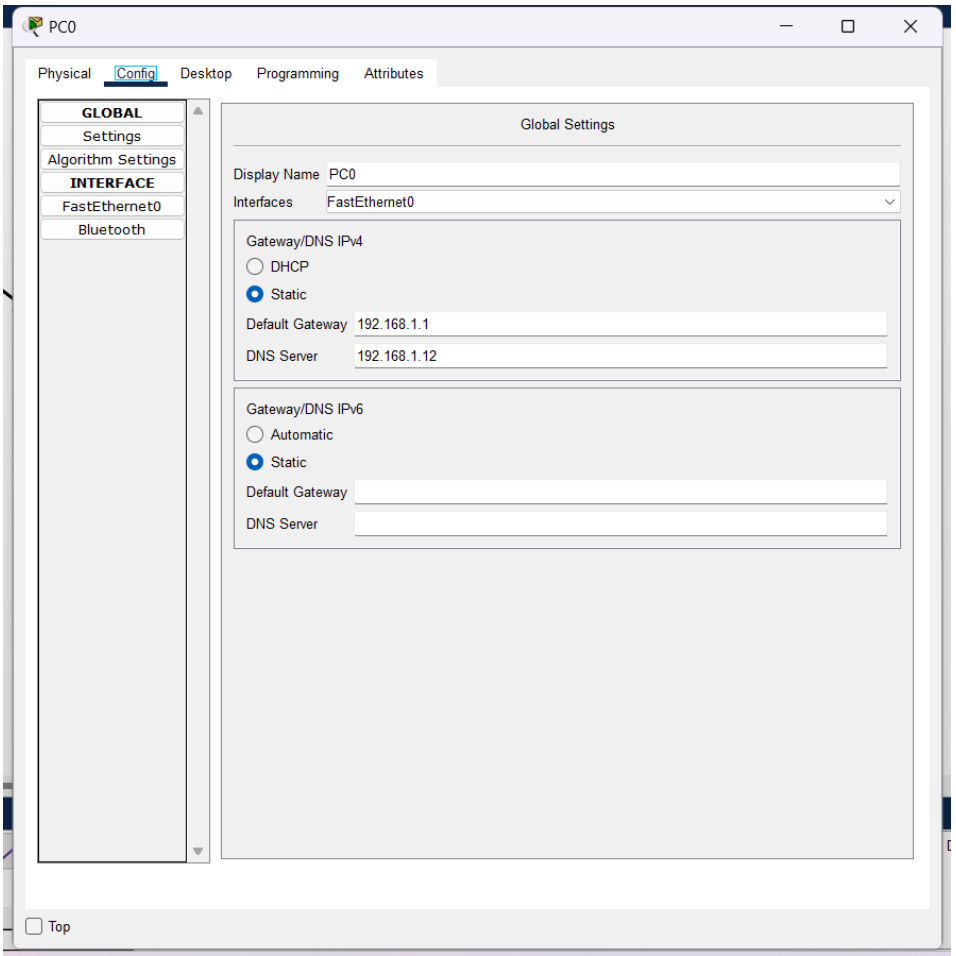


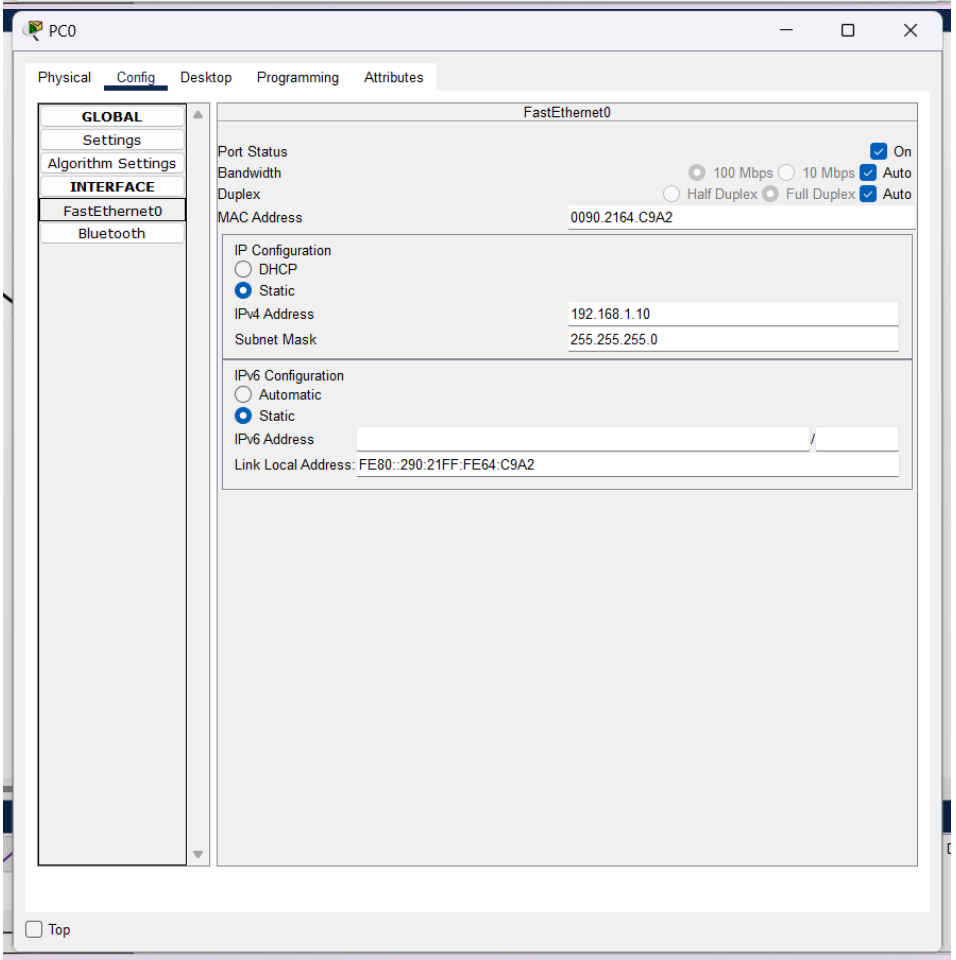


- 在DNS SERVER1添加name和ip的映射（以左侧一组为例，右侧同理）
  - name：`www.baidu.com` ip：`192.168.1.13`
  - name：`www.tongji.edu.cn` ip：`192.168.2.13`

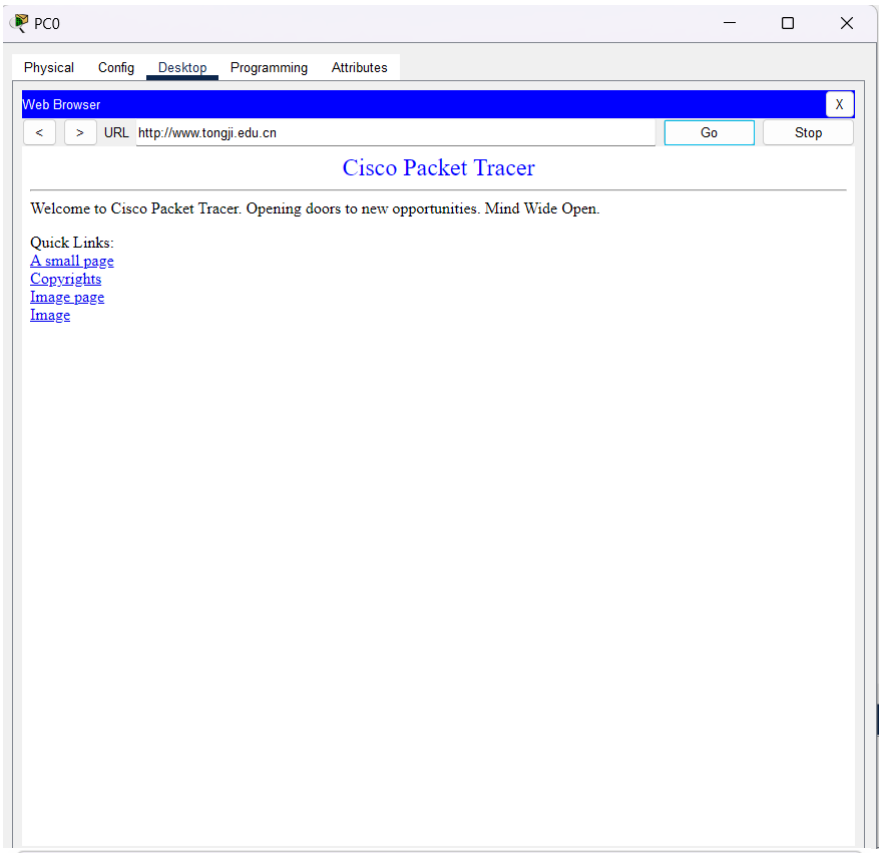


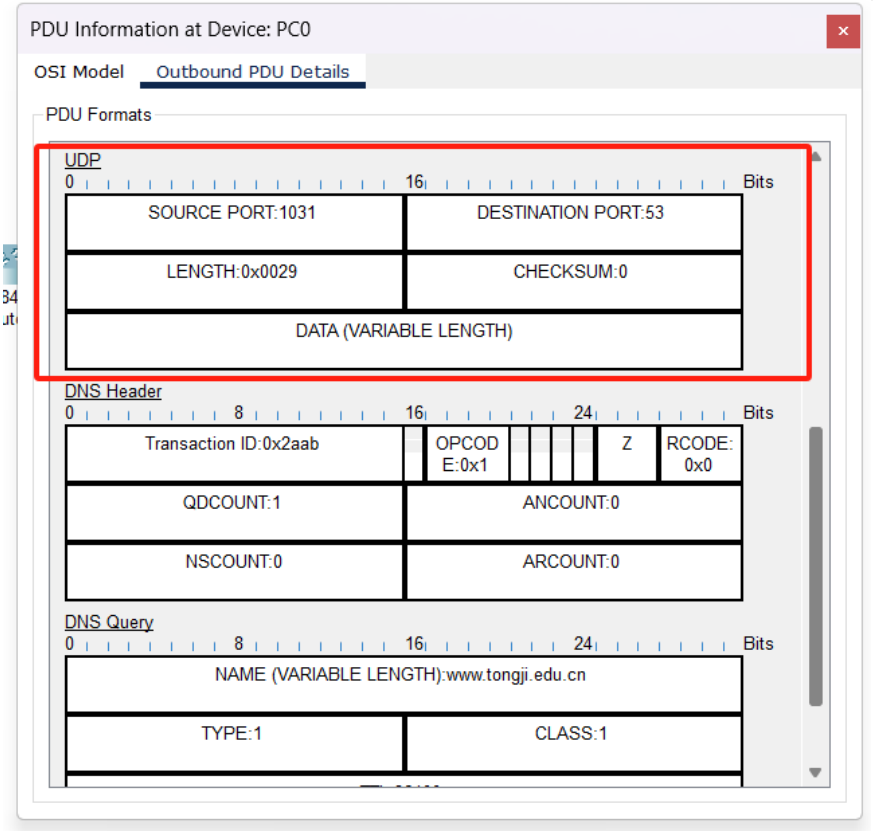
- 配置各个PC（以PC0为例）



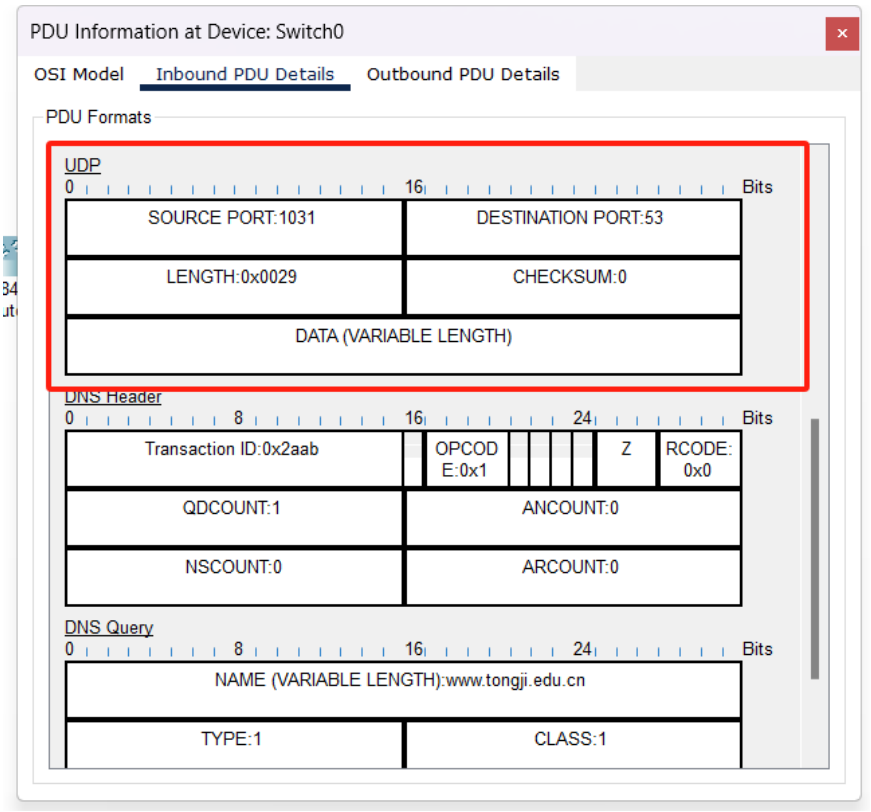


- 打开PC0浏览器，输入配置Web服务器的Web地址，产生UDP数据报文

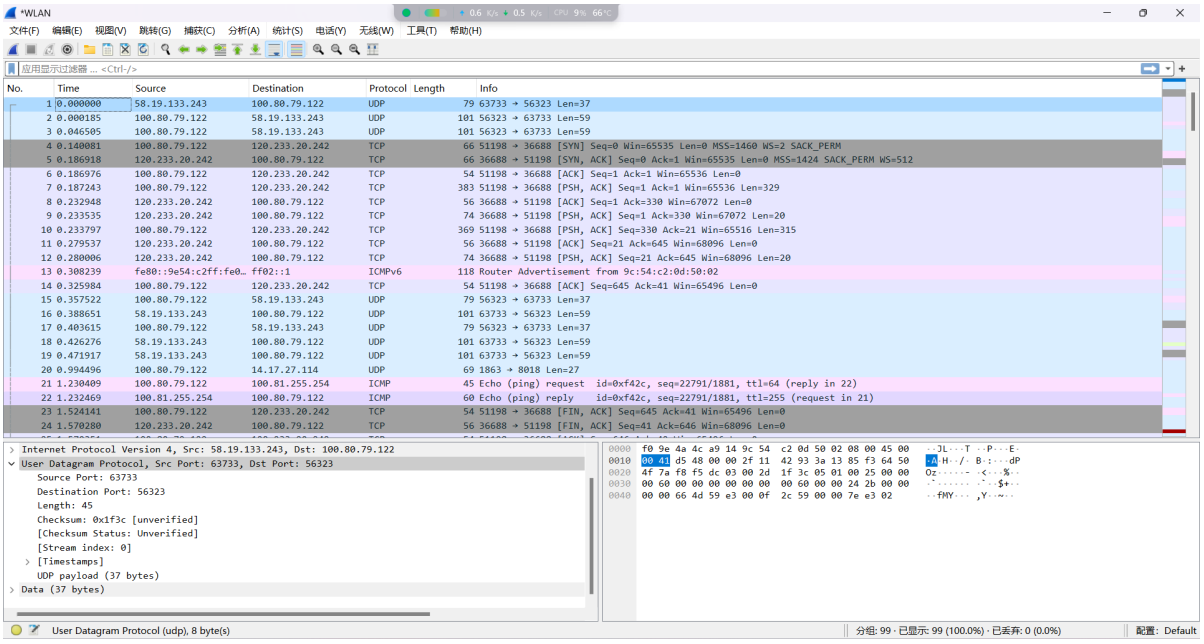




- 观察并分析UDP数据报文

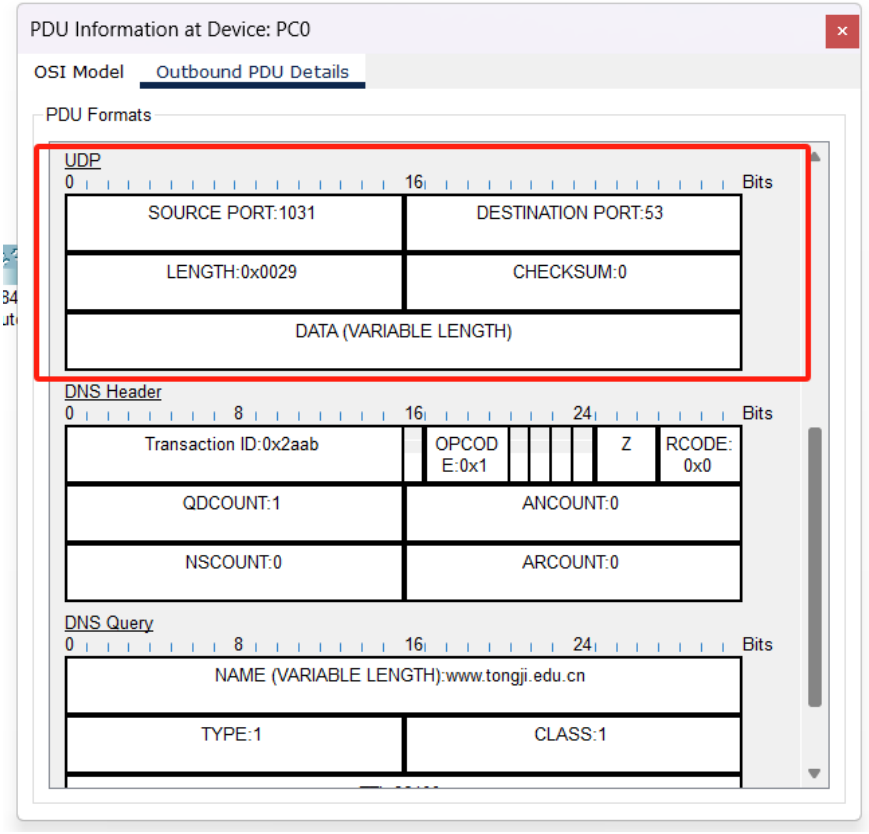


- WireShark抓取TCP报文并分析

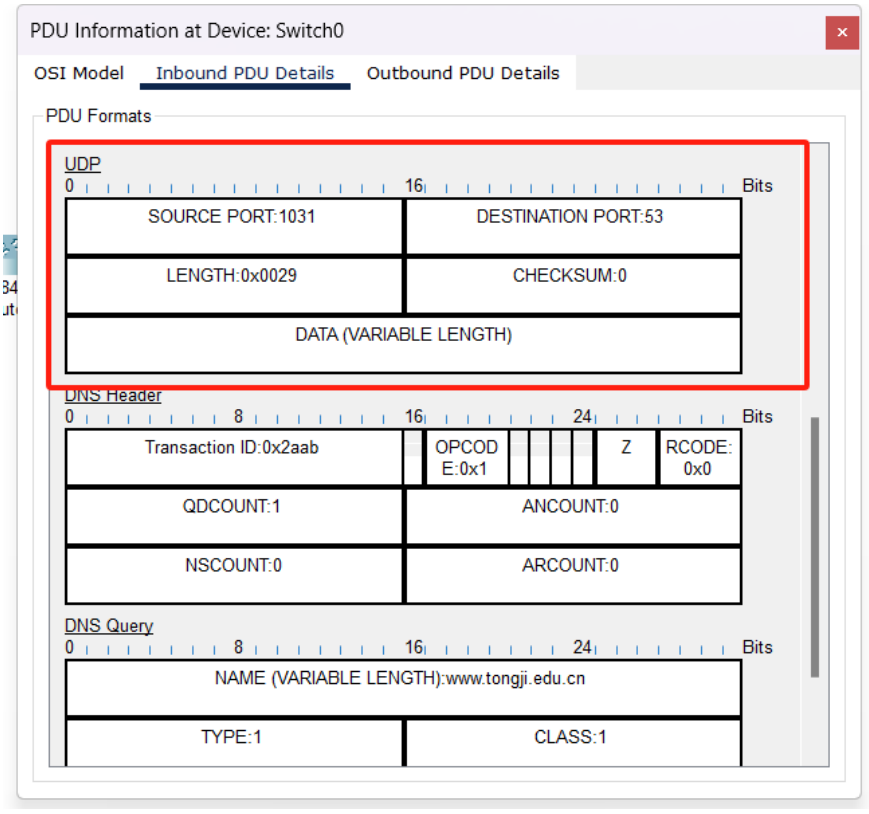


五、实验现象

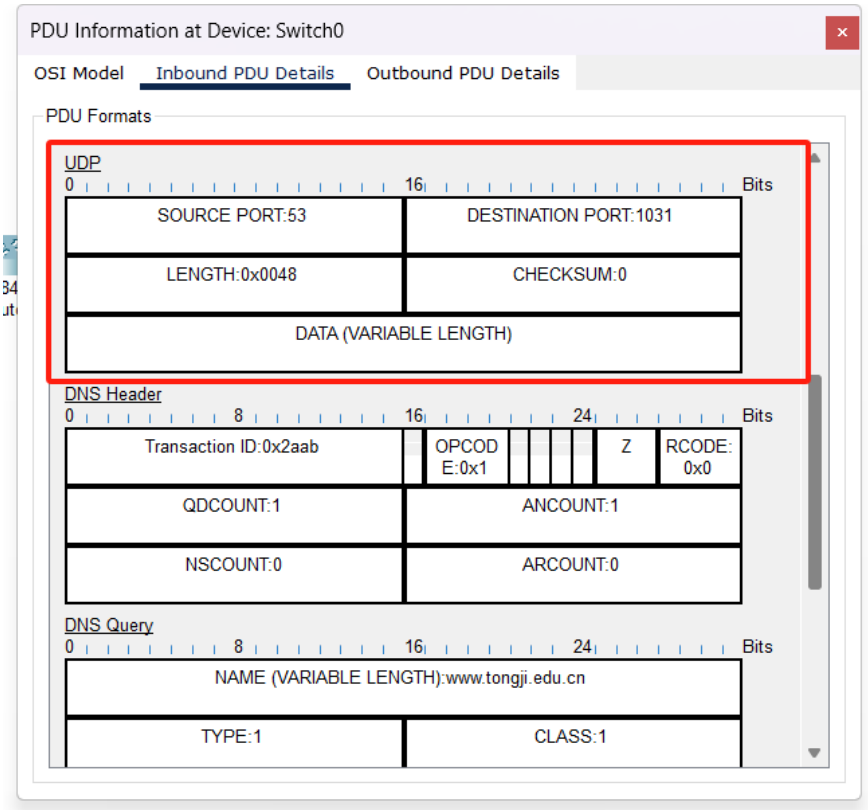
- 分析在Packet tracer中UDP报文情况
  - UDP数据报分为首部和用户数据部分。在上图中我们可以看到：
    - 源端口号为1031
    - 目的端口号为53
    - UDP数据报长度为0x0029即41（包括了首部和数据）
    - 校验和为0标识源主机不想计算校验和



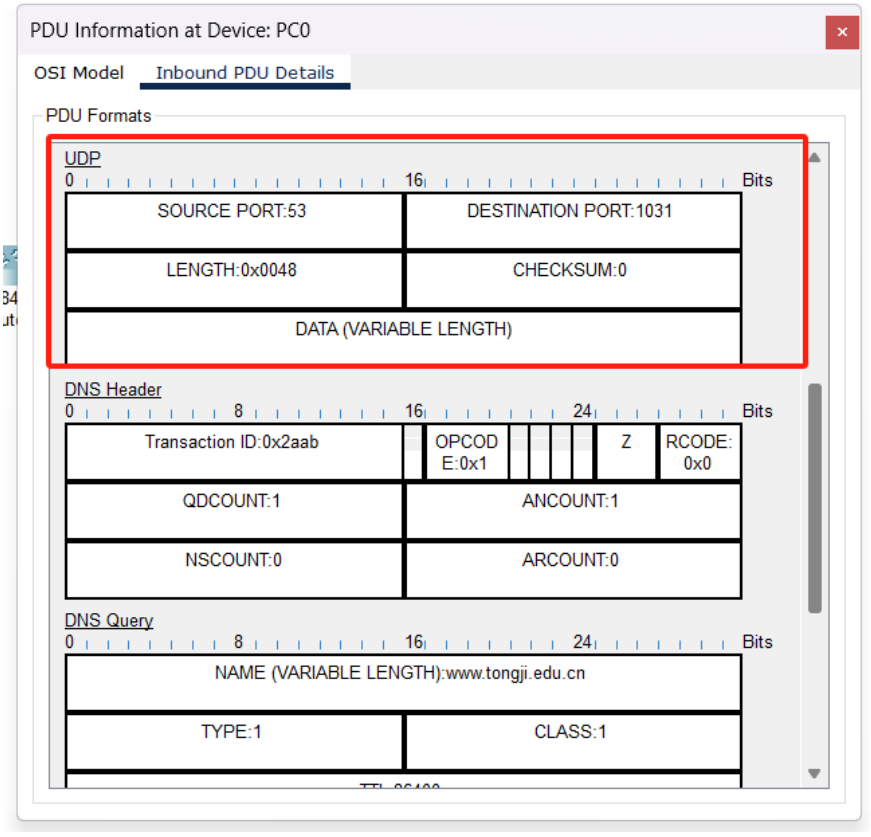
- 这是DNS服务器接收的PC0发来的UDP报文，可以看到其内容和PC0发送时完全一致



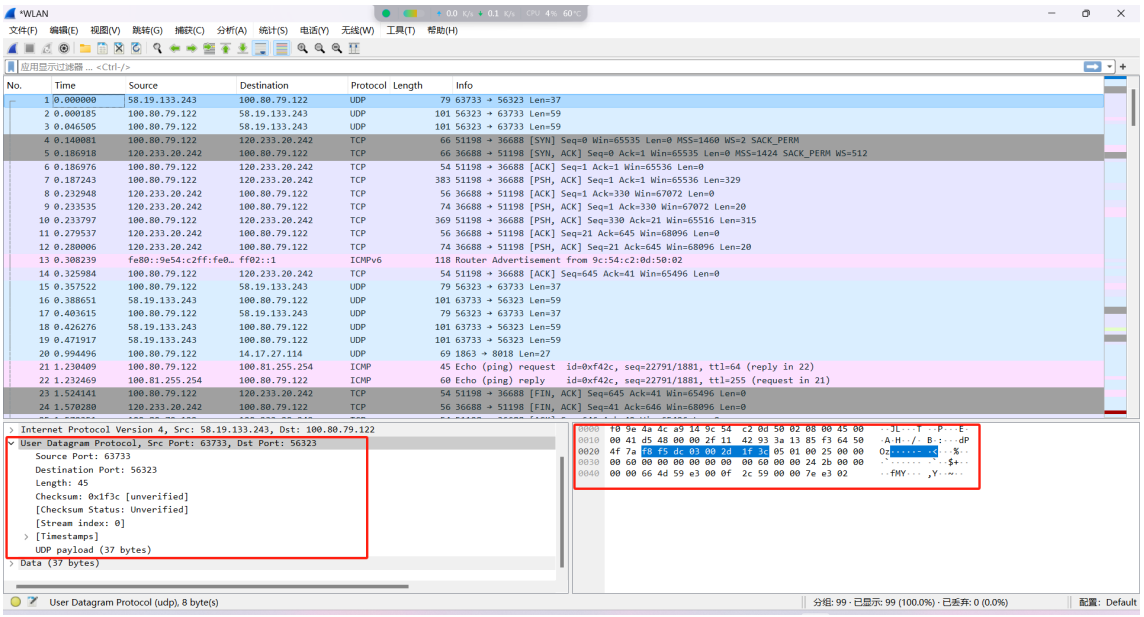
- 这是DNS服务器向PC0发送的UDP报文，可以看出：
  - 源端口号为DNS服务器的53端口
  - 目的端口号为PC0的1031端口
  - 数据报长度为0x48即72
  - 校验和为0.



- 这是PC0接受的DNS服务器发来的UDP报文，其内容和服务器发送时所封装的内容一致



- 用WireShark抓取UDP数据包，查看并分析



- 可以看到，在这个UDP数据报中：
  - 源端口号为63733
  - 目的端口号为56323
  - 数据报长度为0x2d即45
  - 校验和为0x1f3c

六、实验结论

- 通过本次UDP数据包分析实验，得出以下结论：
  - **UDP协议特点**：
    - UDP（用户数据报协议）是一种无连接的、不可靠的数据传输协议。它不保证数据的顺序和完整性，适合对传输时延敏感且可以容忍部分数据丢失的应用场景，如实时视频、在线游戏和语音通话等。
  - **报文结构分析**：
    - UDP报文由首部和数据部分构成。首部包括源端口、目的端口、长度和校验和字段，其中校验和用于检测数据传输中的错误，但不强制保证数据的正确接收。
    - 通过分析抓取的UDP报文，可以看到首部的各个字段在数据传输中的具体作用。
  - **无连接性和不可靠性**：
    - 实验证明，UDP协议在发送数据时无需建立连接，也不会对数据进行重传和确认。这种机制降低了协议的复杂性和开销，适合需要快速传输的小数据量应用。
    - 由于不维护连接状态，UDP的数据传输效率高，但可靠性较低，所有的错误处理和数据校验工作需要在应用层完成。
  - **实时应用优势**：
    - UDP没有拥塞控制，允许应用层更好地控制数据发送的时间和频率，因此在实时应用中具有明显优势。实时视频、直播等应用可以接受一定程度的数据丢失，但不能容忍较大的时延，因此UDP协议在这些场景中得到了广泛应用。
- 本次实验通过理论与实践结合，深入理解了UDP协议的基本概念、报文结构及其在实际网络中的应用。实验结果表明，UDP协议在对传输时延敏感的应用场景中具有独特优势，虽然它不保证数据的可靠传输，但其低开销和高效率使其在特定应用中不可或缺。实验还提高了我们对网络协议分析工具的使用能力，为今后的网络研究和应用打下了坚实基础。