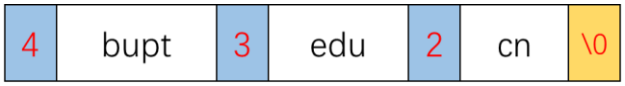
DNS Project 常见问题

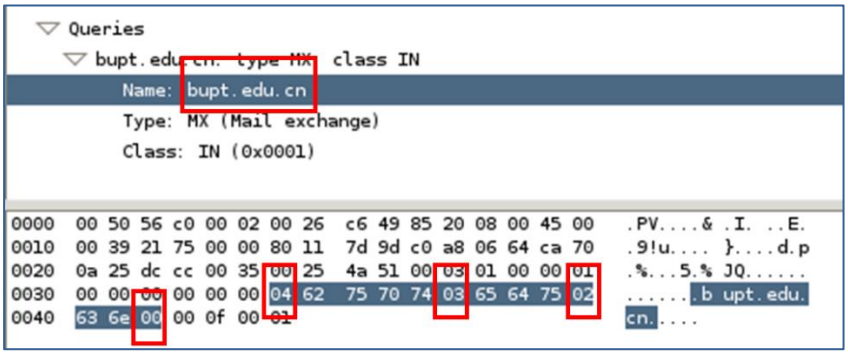
难点提示

1. **DNS 报文构建：**仔细理解 DNS 报文各字段含义、所占字节数。根据 DNS 报文大致可分为三部分：DNS 报头部分、DNS 查询字段部分、DNS 响应字段部分。每个部分可根据字节长度、数据类型给出相应的结构体，用于构建整个 DNS 报文。
2. **查询域名字符串和 DNS 报文中对应字符串的转换：**仔细理解查询域名字符串和 DNS 报文中对应字符串的格式类型。查询域名字符串之间用 ***“.”*** 来分割关键词。DNS 报文中对应字符串由一个或者多个标识符序列组成，每个标识符以字节数的计数值来说明该标示符长度，每个名字以‘/0’结束。

   例如：bupt.edu.cn

   对应 Wireshark 中 Packet Bytes Pane 部分，可以看到：

3. **DNS 本地字节序与网络字节序之间的转换：**仔细理解 ntohs()、htons()，在使用中体会字节的存储方式。
4. **可以用文件来维护资源记录数据库：**在进行文件读取时，因为每一行格式相同，可以考虑按行读取，利用 fgets()函数。

常见问题

Q1：数据库记录（RR）设计的合理性

数据记录的内容需符合 DNS 的设计逻辑，例如，下面的记录即为不合理记录：

www.bupt.edu.cn，86400，IN，A，www.beijingyd.edu.cn

bupt.edu.cn，86400，IN，MX，211.68.71.9

Q2：任务书要求“对于 MX 类型的查询，要求在 Additional Section 中携带对应 IP 地址”。

附件中的“MX query.pcapng”是个例子，可参见其中的 No.=132 和 133 的消息交互。

Q3：如果实现缓存功能，server 维护的数据记录和缓存不应该放在同一个文件里；

Q4：验收过程中，要求能够用 Wireshark 同步抓到通信流程，并实现 DNS packet 的解析；

实现正常解析，需要满足以下两个条件：

（1）每一段通信的 Server 端，必须使用 53 端口，Wireshark 才能识别其为 DNS packet；

（2）DNS packet 的构造必须严格符合 DNS 标准的定义。

Q5：请在编程过程中把 Wireshark 当作重要的工具。

可用 Wireshark 抓正常 DNS 的 packet。如果自己程序的 packet 不能被 Wireshark 正常解析，需要与正常的 DNS Packet 做比较，以发现问题。

（1）可用 Wireshark 配合 nslookup 抓不同类型的查询消息。例如，下面的消息可以查询 MX 类型的记录：

nslookup –query=MX bupt.edu.cn

（2）一些需要做特殊处理的地方

- 基于 TCP 的通信，需要在整个 DNS 报文之前加上 2 个字节，表示整个报文的长度。这样才能用 wireshark 解析出来。
- MX 类型的 RR，多了两个字节，具体参加附件的”MX query.pcapng”文件，No.=133 的包，里面可以看到多出来的 Preference field（占 2 字节）
- strlen()函数只读到’\0’但不包含’\0’，所以为了把结束符也复制进去，长度要+1。
- 虚拟机 ubuntu 系统中，bind()函数绑定小于 1024 号的端口需要 root 权限，所以需要以 sudo 运行程序，否则会 bind failed。
- C 语言中结构体大小计算，这个问题会涉及到对结构体存储空间的分配和操作，具体参见：
  http://blog.sina.com.cn/s/blog_623a81b40100giob.html