Project
A Simple DNS System

2023-04-24

From DNS Protocol to DNS Program

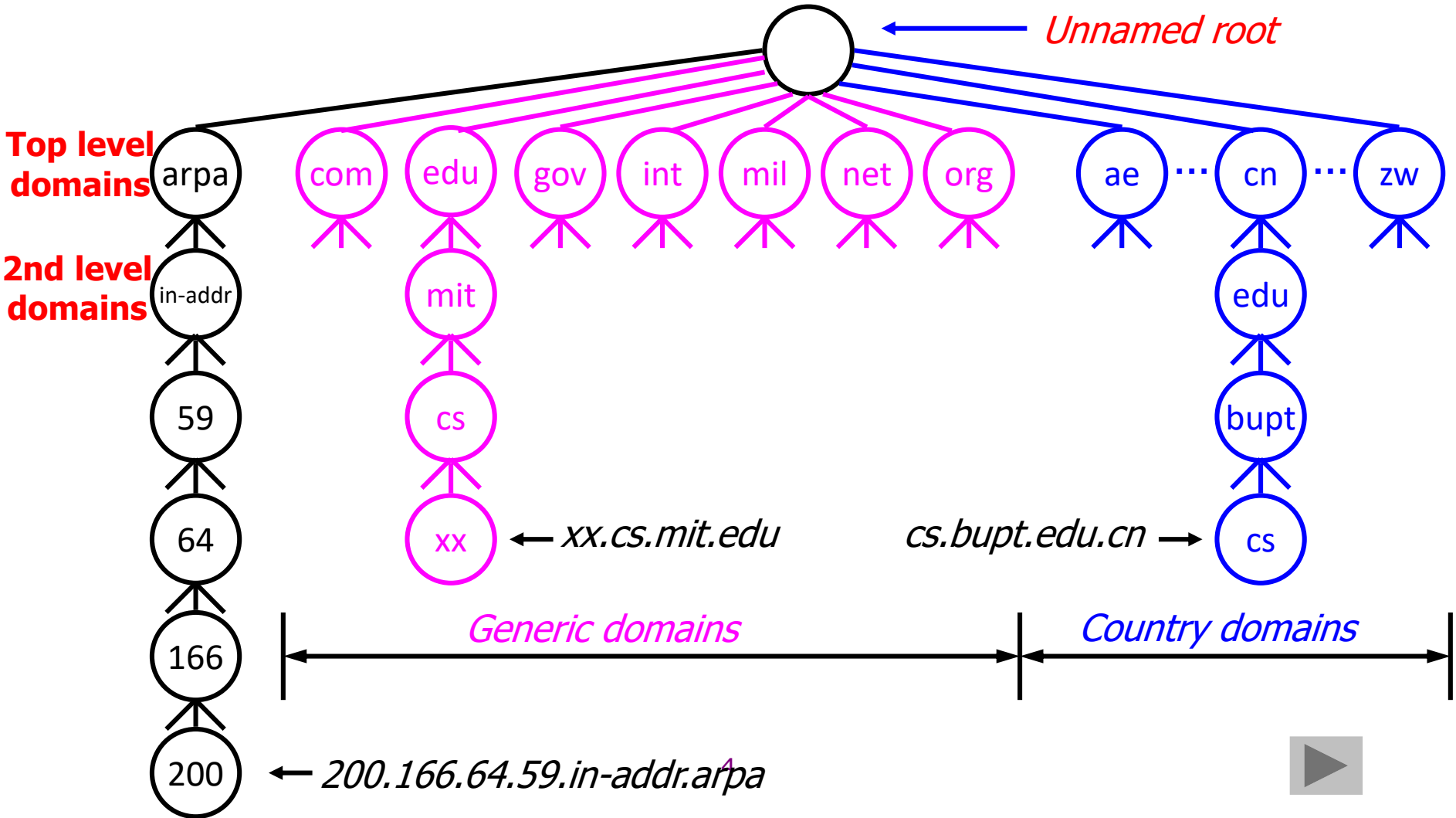
- DNS系统的Hierarchical structure→程序中的实体
- DNS解析过程→实体间的通信过程
- Resource Record定义→server维护的数据结构
- DNS packet format定义→packet的数据结构定义

DNS系统的Hierarchical structure

→

程序中的实体

Domain Namespace – the hierarchical structure

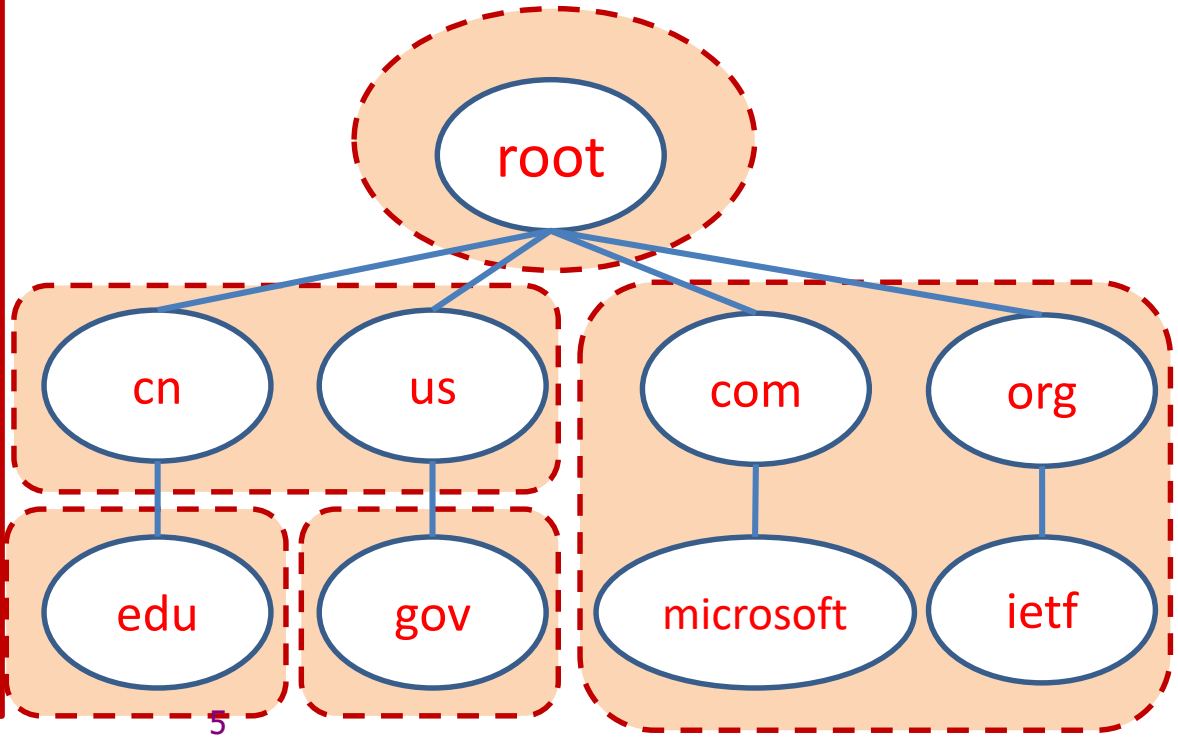


Name Servers

- Name servers are the repositories of information that make up the domain database.
- The database is divided up into sections called zones, which are distributed among the name servers. A zone may be one or more domains or even a sub-domain
- Each name server handles one or more zones. And the essential task of a name server is to answer queries using data in its zones.

- 实验要求
  - 至少支持4个顶级域，至少实现三级域名的解析。
  - 4个顶级域名：

    .cn .org .com .us
  - 二-三级域名：自定义（例如：edu.cn，bupt.edu.cn）
  - DNS server的部署架构可参考右图的示例

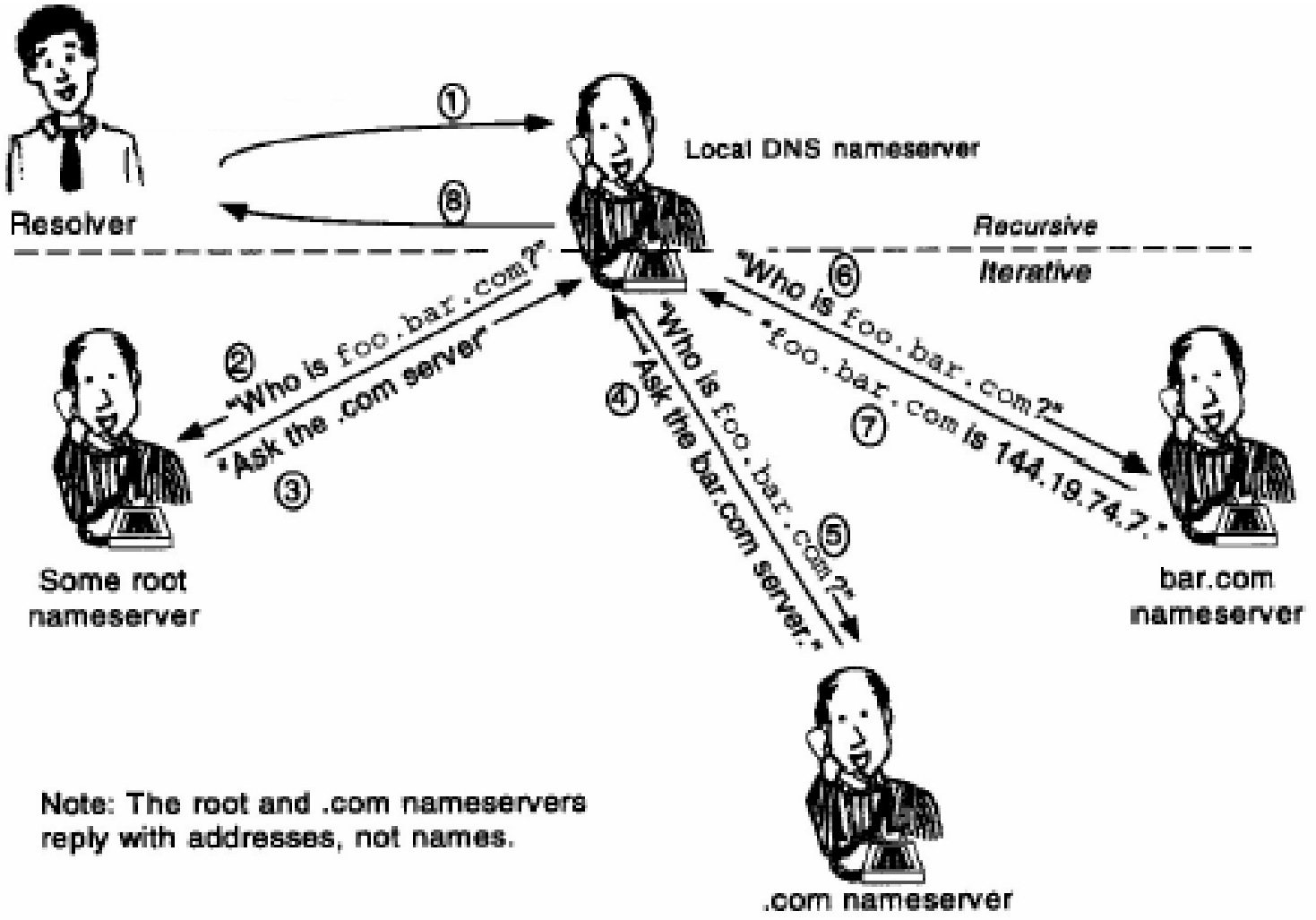


DNS解析过程

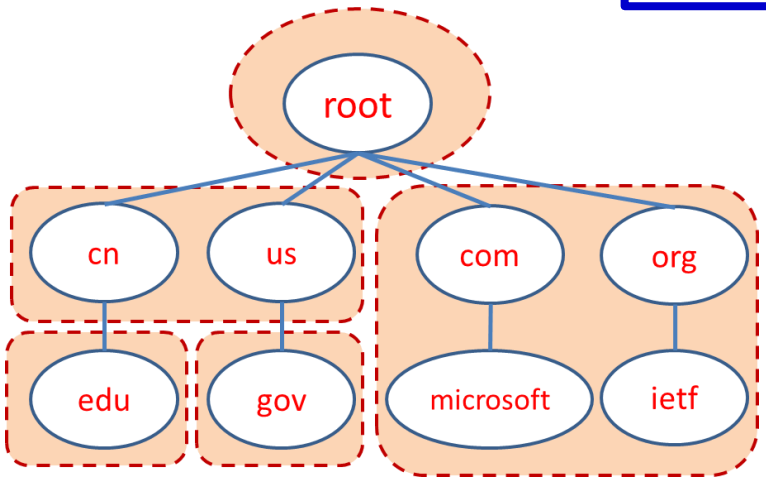
→

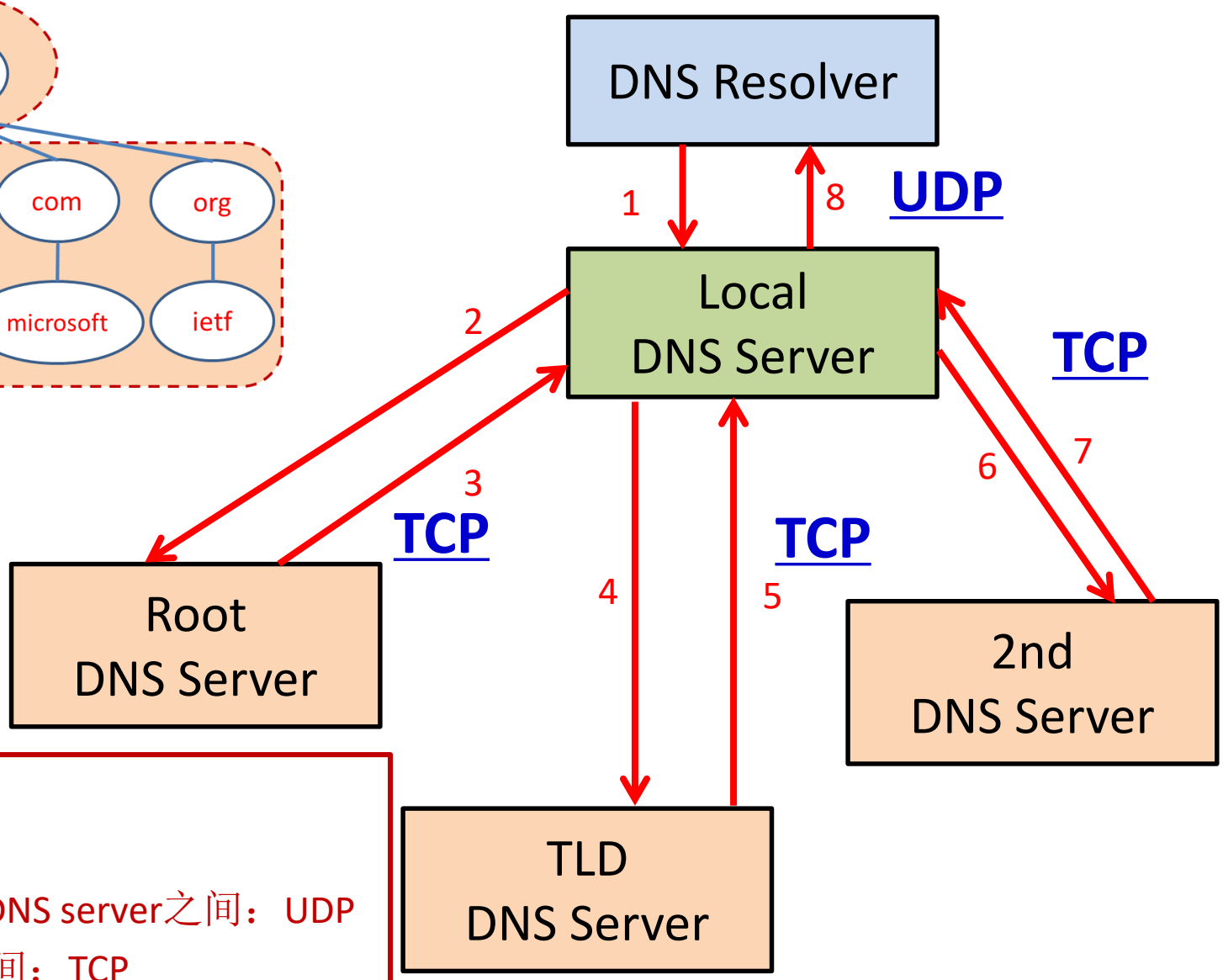
实体间的通信过程

Mapping Domain Names to Addresses – example of iterative resolution



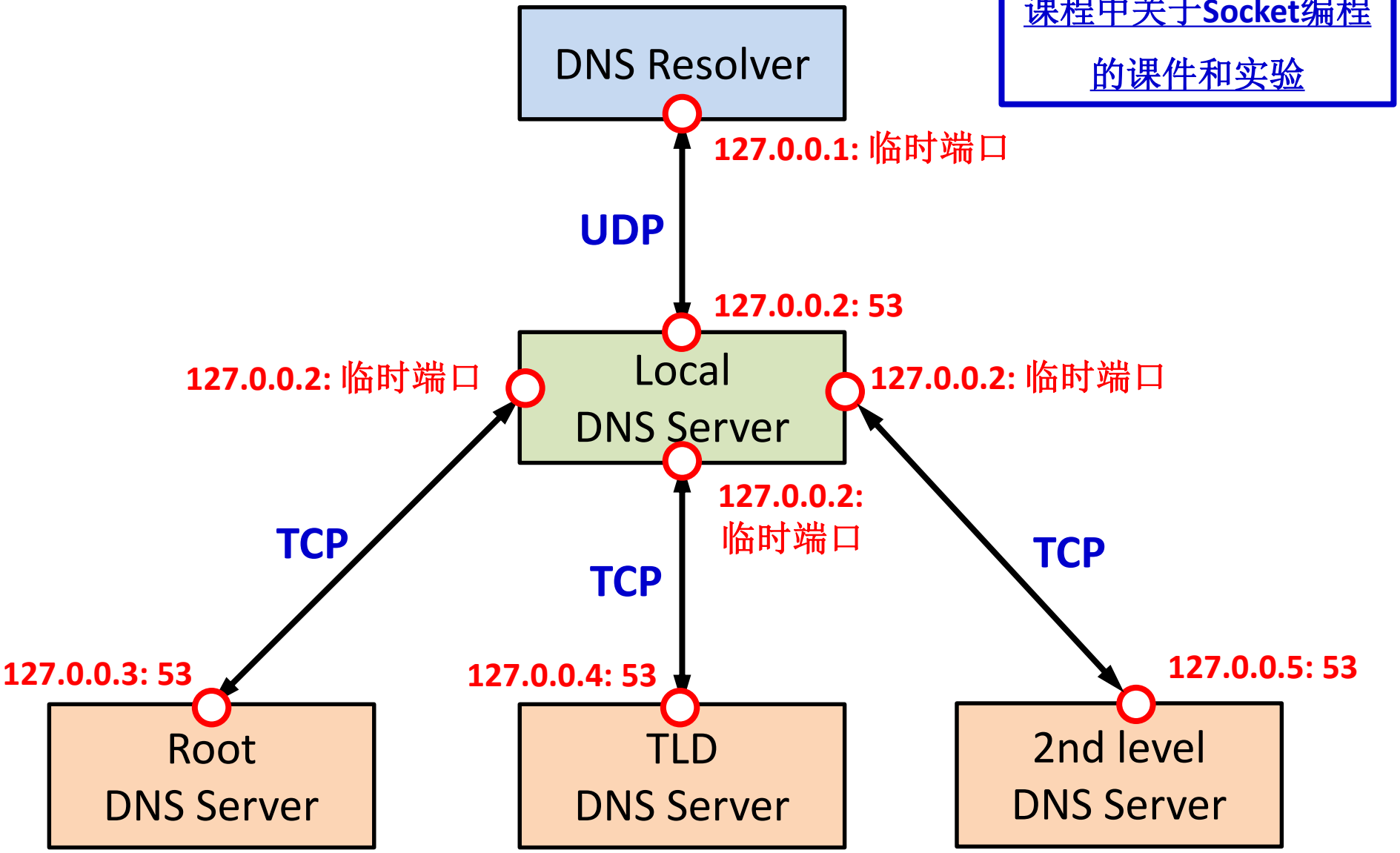
这是不存在cache的情况，每一步都可能由于cache了查询结果而得到优化



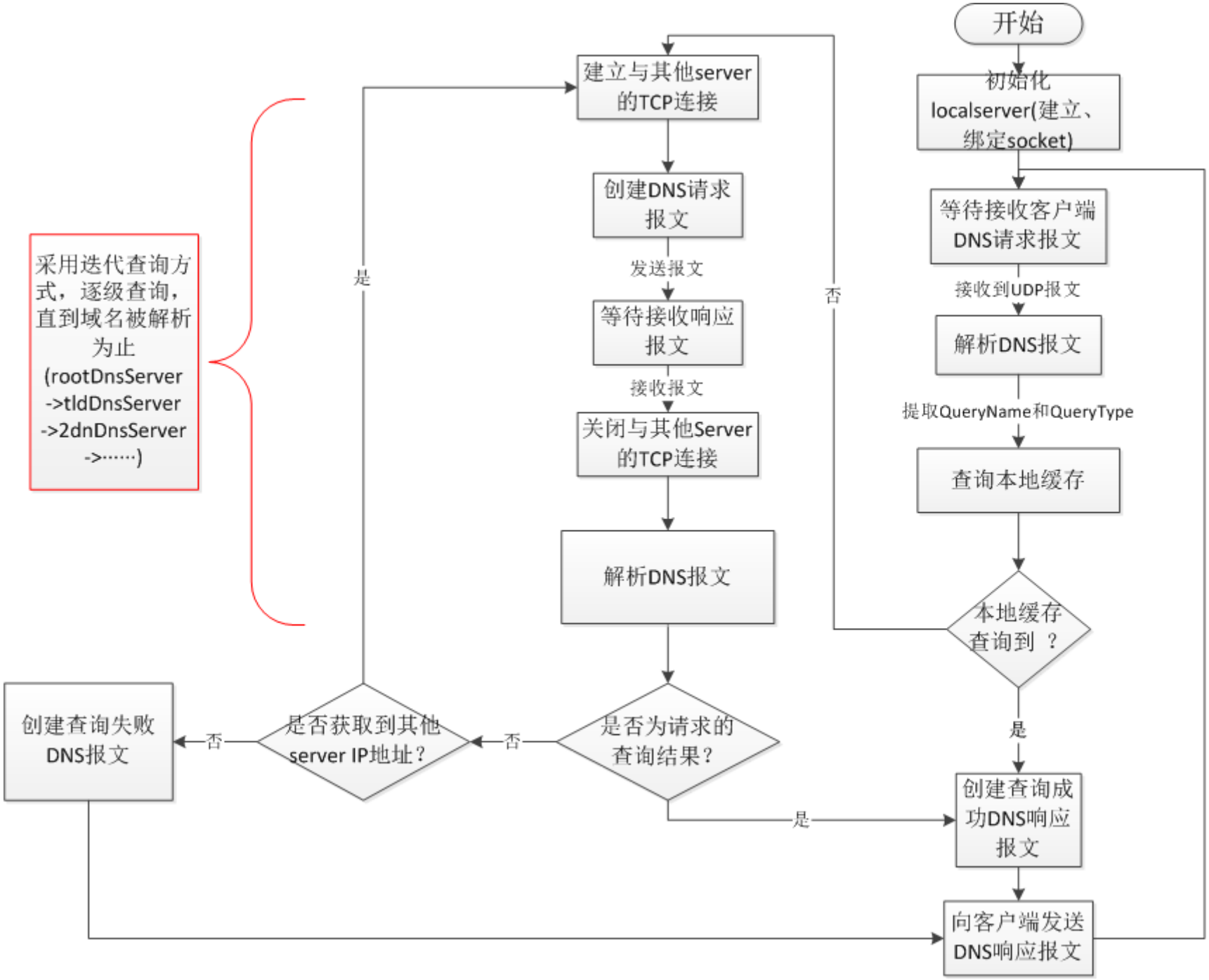


- 实验要求
  - 传输层协议：
  - client与local DNS server之间：UDP
  - DNS server之间：TCP

每一段的通信，请参考课程中关于Socket编程的课件和实验



Local DNS Server的流程示例


开始
初始化localserver(建立、绑定socket)
等待接收客户端DNS请求报文
接收到UDP报文
解析DNS报文
提取QueryName和QueryType
查询本地缓存
本地缓存查询到 ？
否
是
创建查询成功DNS响应报文
是
是否为请求的查询结果？
否
是否获取到其他server IP地址？
否
是
建立与其他server的TCP连接
创建DNS请求报文
发送报文
等待接收响应报文
接收报文
关闭与其他Server的TCP连接
解析DNS报文
采用迭代查询方式，逐级查询，直到域名被解析为止(rootDnsServer->tldDnsServer->2dnDnsServer->……)
创建查询失败DNS报文
向客户端发送DNS响应报文

Resource Record定义

→

server维护的数据结构

Resource Record

- Each domain in the DNS has one or more Resource Records (RRs), which are fields that contain information about that domain
- Each RR has the following information
  - Owner: the domain name where the RR is found
  - Type: specifies the type of the resource in this RR
    - A – Host Address ; MX – Mail Exchanger; **…**
  - Class: specifies the protocol family to use
    - IN – the Internet system
  - TTL: specifies the Time To Live (in unit of second) of the cached RRs
  - RDATA: the resource data

- 实现英文域名的解析
- 例如： www.bupt.edu.cn ；
- 数据库记录示例：
  - www.bupt.edu.cn 86400 IN A 192.168.1.25
  - bupt.edu.cn 86400 IN MX mail.bupt.edu.cn
  - mail.bupt.edu.cn 86400 IN A 192.168.1.37
- 支持的Resource Record类型：A、MX、CNAME；对于MX类型的查询，要求在Additional Section中携带对应IP地址；

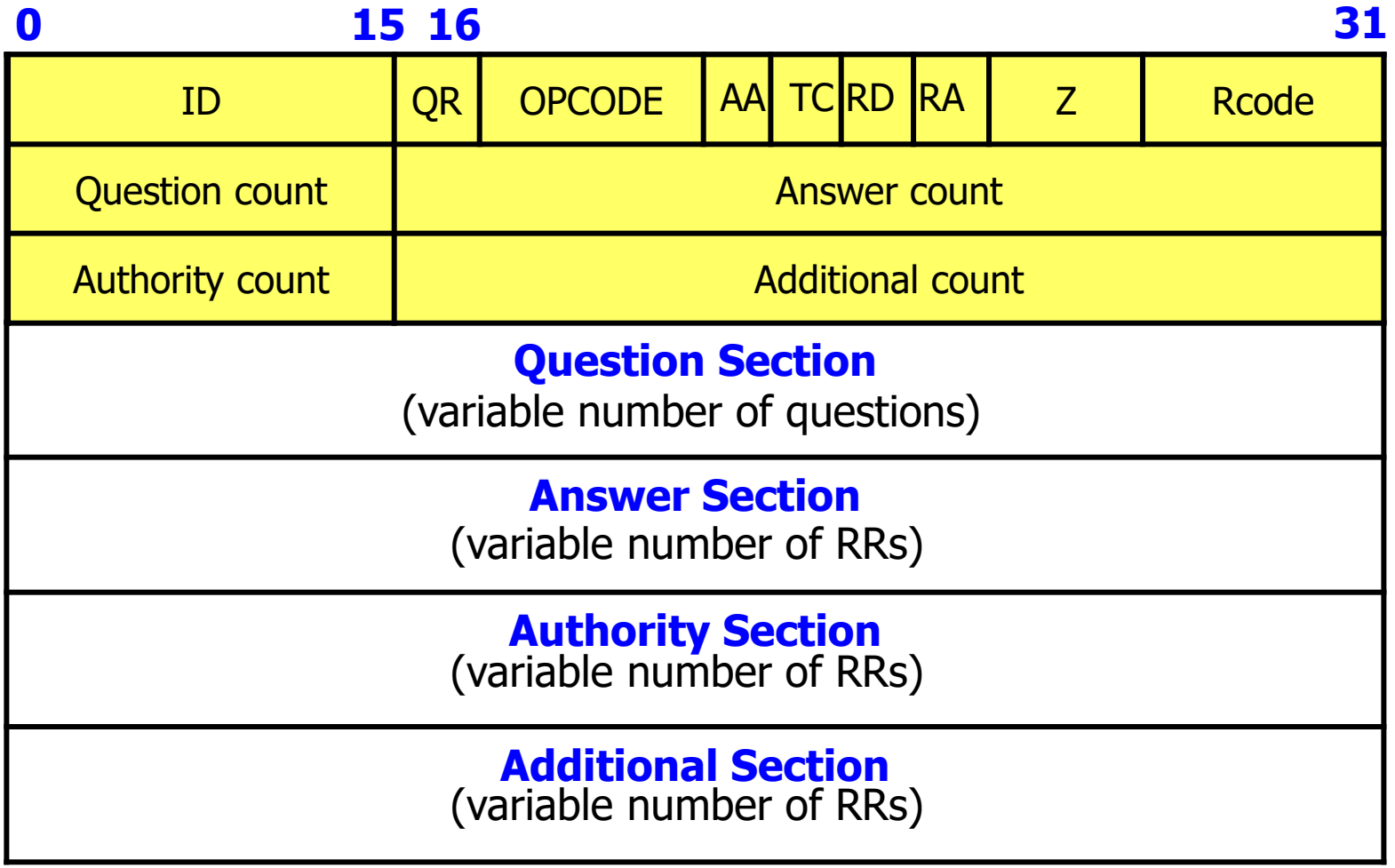
DNS packet format定义

→

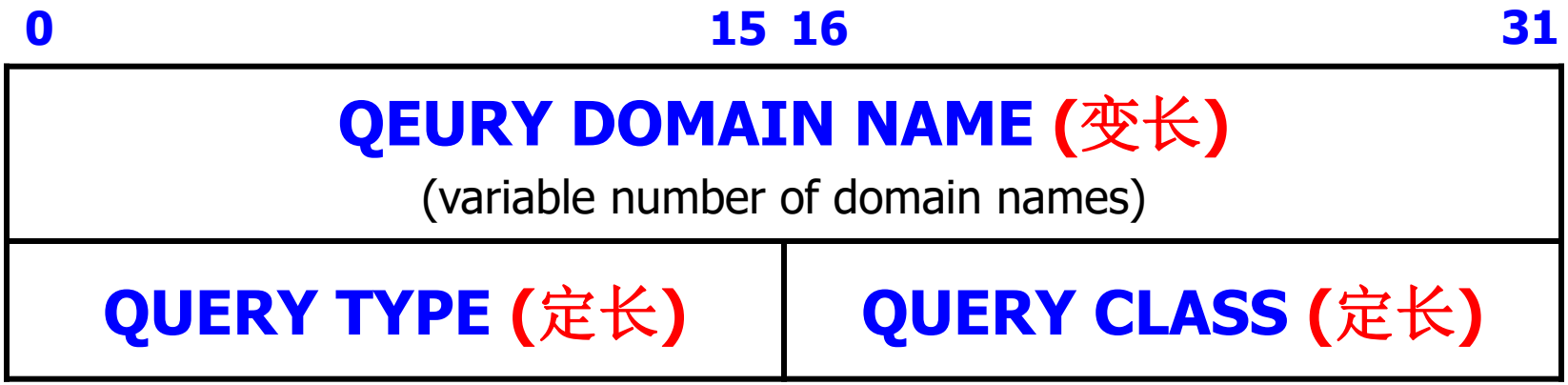
packet的数据结构定义

DNS Message Format (from RFC 1035)

- *Query* and *Response* messages, both with *same message format*



Question Section Format



- QUERY TYPE: 16-bit field used to specify the type of the query
  - A – Host address
  - MX – Mail exchanger for the domain
  - ...
- QUERY CLASS: 16-bit field used to specify the class of the query
  - IN – Internet system
  - ...

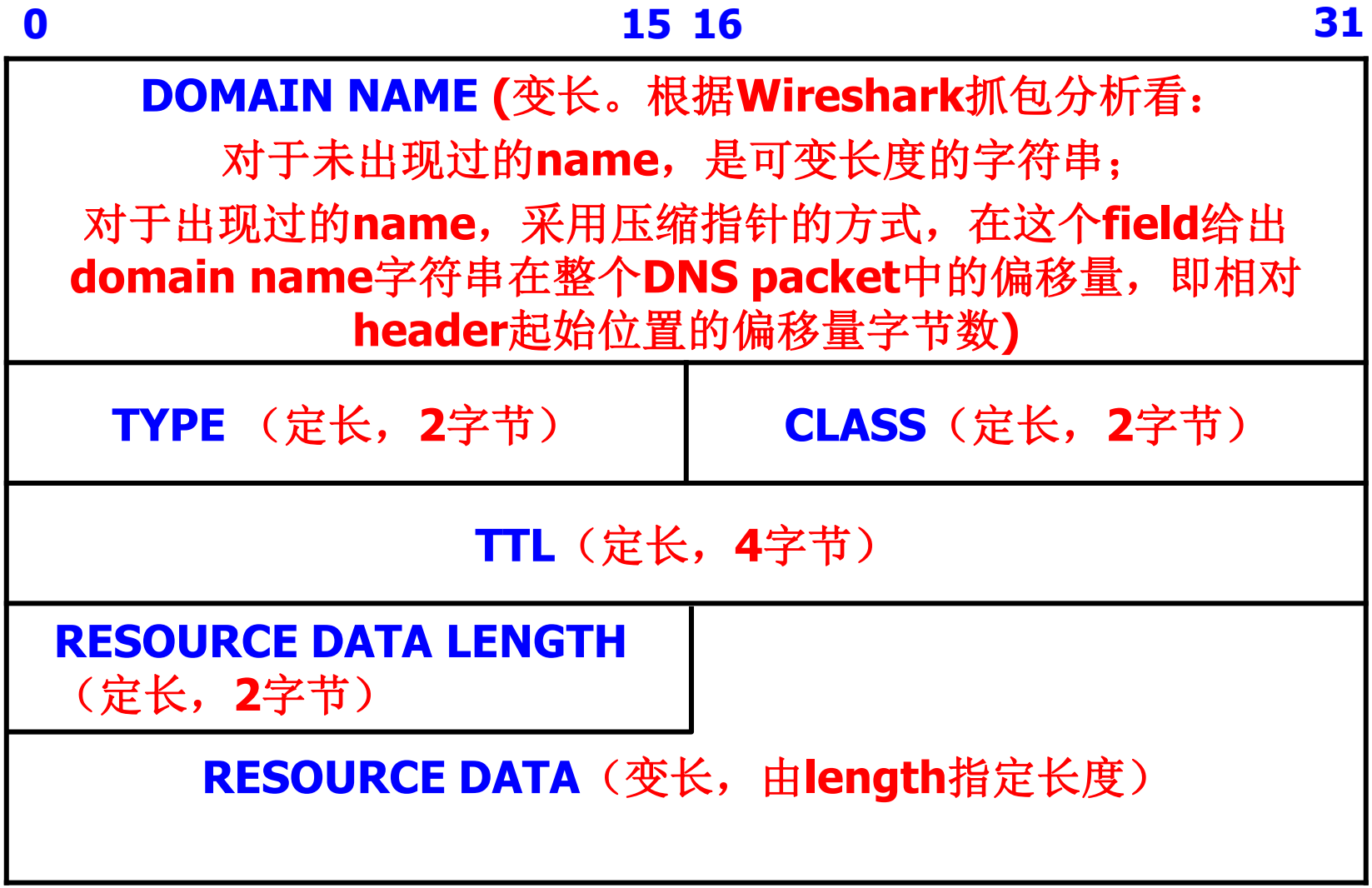
DNS packet的数据结构定义（仅供参考）

```
struct DNS_Header {
        unsigned short id;
        unsigned short tag; (包含QR到Rcode的定义)
        unsigned short queryNum;
        unsigned short answerNum;
        unsigned short authorNum;
        unsigned short addNum;
};

struct DNS_Query{
        unsigned char *name;
        unsigned short qtype;
        unsigned short qclass;
};
```

具体定义可根据编程技术做优化

Resource Record Format



DNS Packet中RR的结构体定义（仅供参考）

处理方式一：结构体定义中，只定义定长的field，变长的field，在构造DNS packet的时候，直接填写进buffer；

```
struct DNS_RR{
    unsigned  short  type;
    unsigned  short  class;
    unsigned  int  ttl;
    unsigned  short  length;
};
```

DNS Packet中RR的结构体定义（仅供参考）

处理方式二：结构体定义中，变长的field，定义成指针，其指向的空间临时分配，填写内容后copy进buffer；

```
struct DNS_RR {
        unsigned char *name;
        unsigned short type;
        unsigned short _class;
        unsigned int ttl;
        unsigned short data_len;
        unsigned char *rdata;
};
```