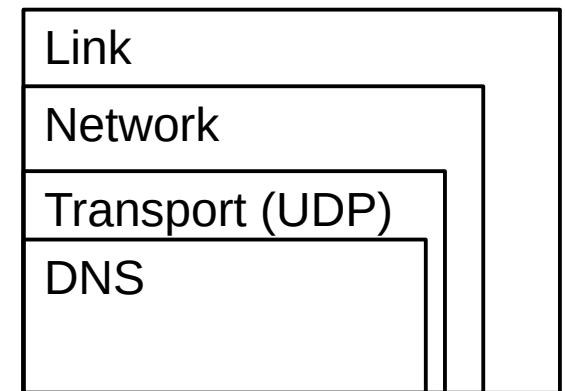


DNS

- Distributed, hierarchical database that maps hostnames to IP addresses
- Application level protocol
- Berkeley Internet Name Domain (BIND)
- UDP
- What does `zeus.cs.pacificu.edu` *mean*?
- Top level domain



Centralized Database Issues

- Single point of failure
- Traffic Volume
- Distant centralized database
- Maintenance

Originally

- How does it work?
 - originally, just ONE file, hosts.txt, that was copied around to all the machines on the Internet (ARPANET) every night
 - /etc/hosts file still exists in UNIX
 - look here first, then queries the DNS server
 - on zeus: `cat /etc/hosts | more`
- Zones:
 - non-overlapping areas in the DNS
 - each zone as its own Name Server (plus a back up or two)
 - the Name Server contains the authoritative records for all hosts in the zone
 - not cached, always correct

Hierarchical Database

- Root server
- Top Level Domain (TLD) server
- Authoritative server
- Local DNS server

Query Process

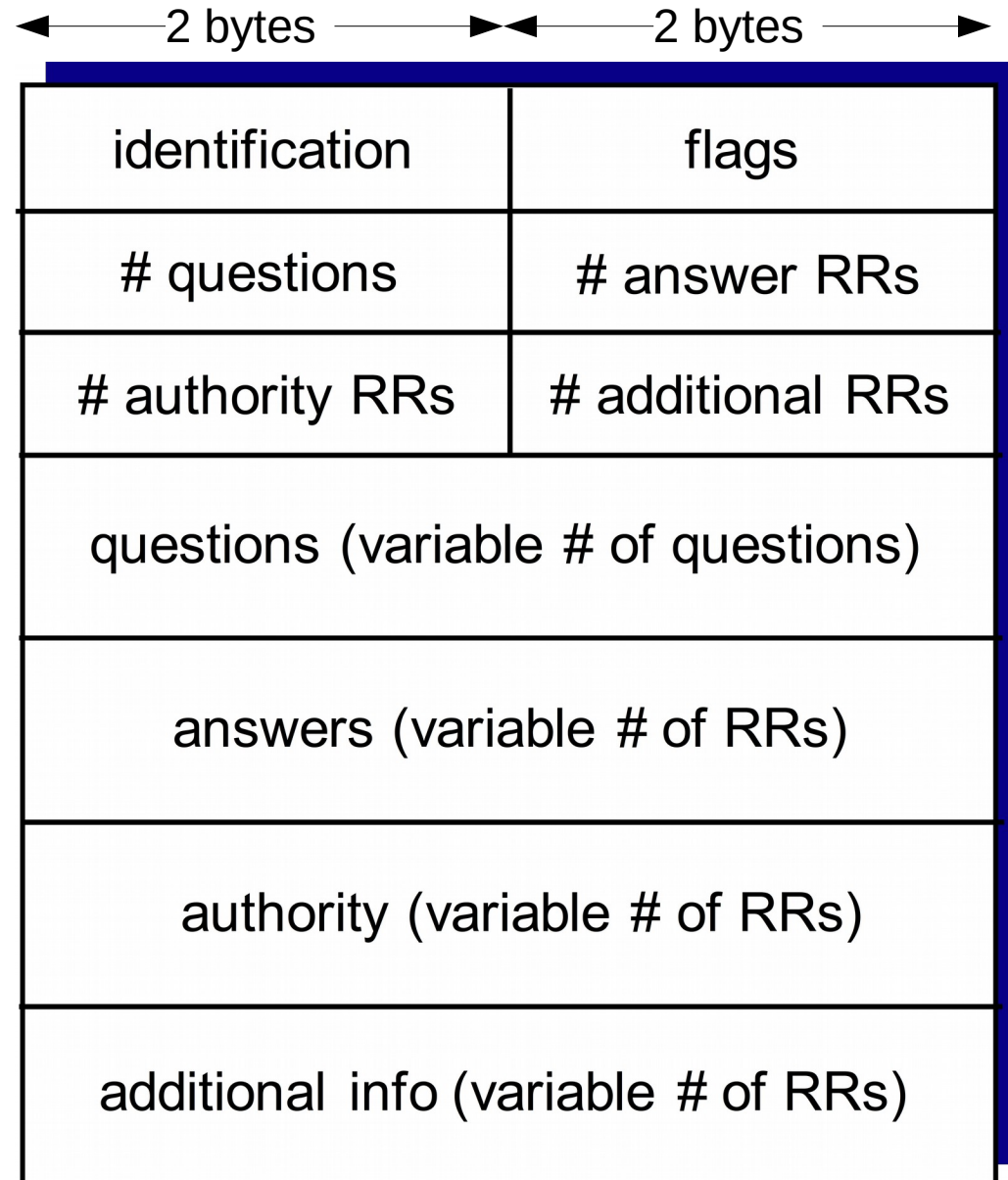
Query 2

- `resolve(zeus.cs.pacificu.edu)`

DNS Records

- Resource Record
 - Domain Name: pacificu.edu (string)
 - TimeToLive: How stable is this record (int, seconds)
 - Class: In – Internet (string)
 - Type: A – Address, SOA – Authority Info, NS – Name Server, etc (string)
 - Value: Data (IP address)

DNS Messages

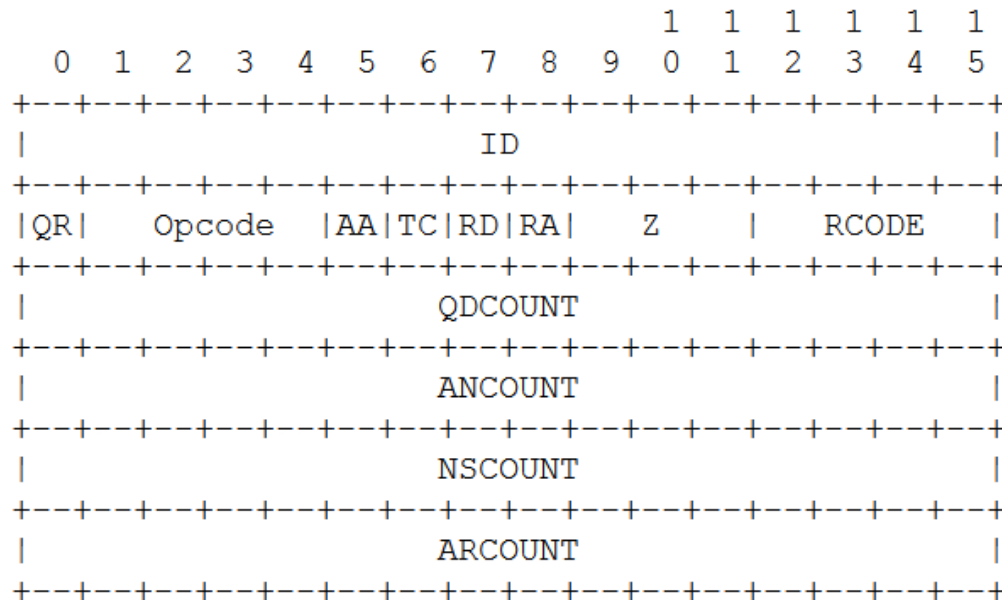


Header

(RFC 1035, section 4)

4.1.1. Header section format

The header contains the following fields:



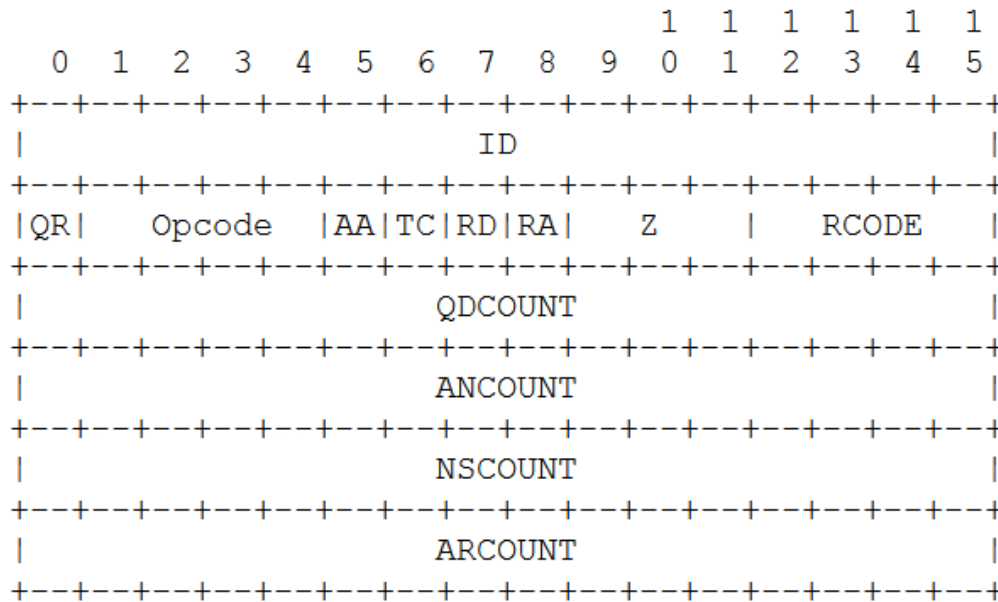
```
#define PACKED __attribute__((__packed__))

typedef struct PACKED
{
    uint16_t id;
    uint16_t flags;
    uint16_t numQ;
    uint16_t numAnswerRR;
    uint16_t numAuthRR;
    uint16_t numAddRR;
    char data;
} DNS;
```

C Code

4.1.1. Header section format

The header contains the following fields:



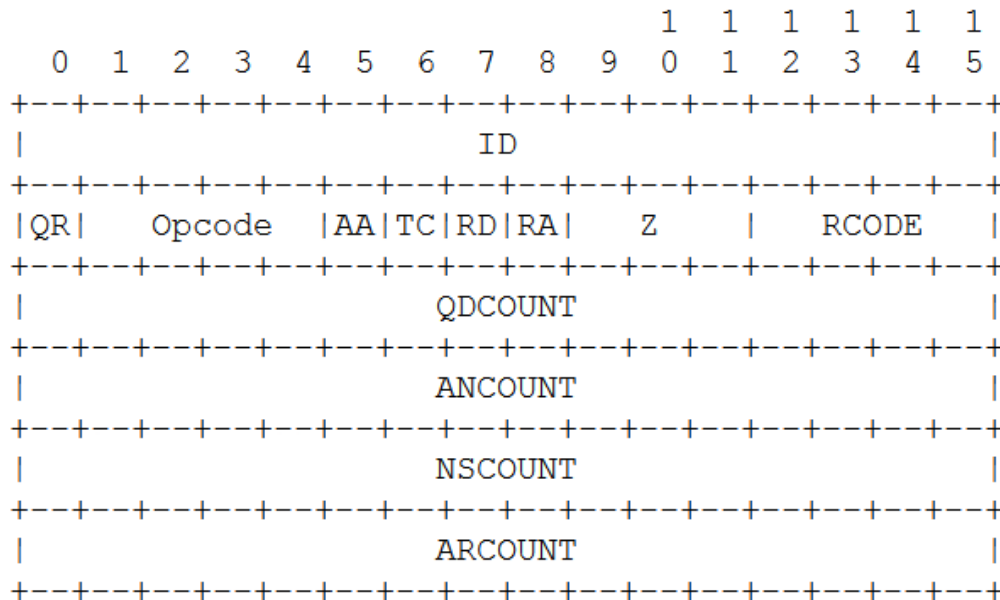
```
#define SET_RD 0x0100
#define SET_RA 0x0080
#define SET_QR 0x8000
#define SET_AA 0x0400
#define SET_OPCODE_QUERY 0x0000
#define SET_OPCODE_IQUERY 0x0800
#define SET_OPCODE_STATUS 0x0100
#define SET_RCODE_FORMAT_ERROR 0x0001
#define SET_RCODE_SERVER_ERROR 0x0002
#define SET_RCODE_NAME_ERROR 0x0003
#define SET_RCODE_NOT_IMPL_ERROR 0x0004
#define SET_RCODE_REFUSED_ERROR 0x0005
```

Danger! Bit Fields

Header

4.1.1. Header section format

The header contains the following fields:



```
typedef struct PACKED
{
    uint16_t id;

    union
    {
        uint16_t flagData;
        struct PACKED
        {
            unsigned char RCODE:4;
            unsigned char Z:3;
            unsigned char RA:1;

            unsigned char RD:1;
            unsigned char TC:1;
            unsigned char AA:1;
            unsigned char Opcode:4;
            unsigned char QR:1;

        } Bits;
    } BitFlags;

    uint16_t numQ;
    uint16_t numAnswerRR;
    uint16_t numAuthRR;
    uint16_t numAddRR;
    char data;
} DNS;
```


Questions (RFC 1035, section 4)

4.1.2. Question section format

The question section is used to carry the "question" in most queries, i.e., the parameters that define what is being asked. The section contains QDCOUNT (usually 1) entries, each of the following format:

```

          1 1 1 1 1 1
    0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
|           QNAME
|
+---+---+---+---+---+---+---+---+---+---+
|           QTYPE
|
+---+---+---+---+---+---+---+---+---+---+
|           QCLASS
|
+---+---+---+---+---+---+---+---+---+---+
```

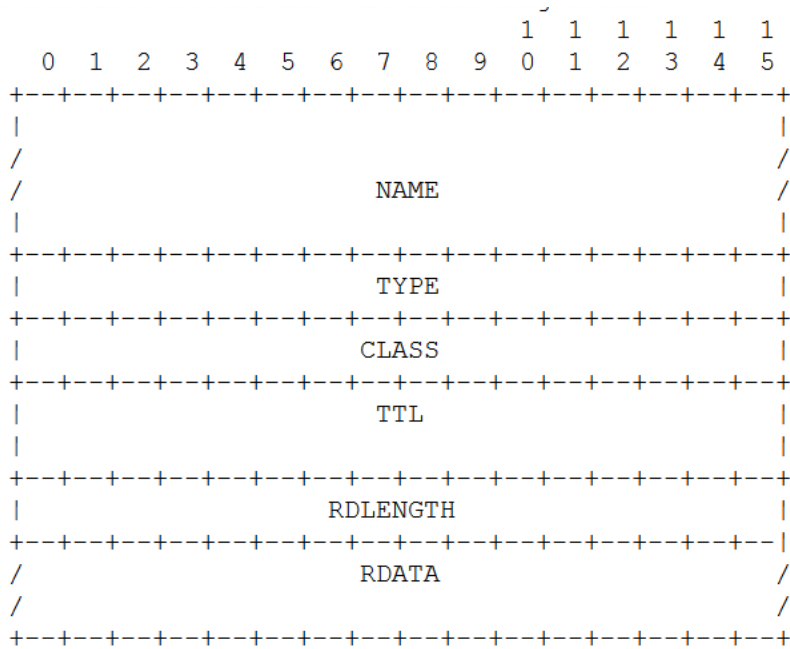
where:

QNAME a domain name represented as a sequence of labels, where each label consists of a length octet followed by that number of octets. The domain name terminates with the zero length octet for the null label of the root. Note that this field may be an odd number of octets; no padding is used.

QTYPE a two octet code which specifies the type of the query. The values for this field include all codes valid for a TYPE field, together with some more general codes which can match more than one type of RR.

QCLASS a two octet code that specifies the class of the query. For example, the QCLASS field is IN for the Internet.

Answers



where:

- NAME a domain name to which this resource record pertains.
- TYPE two octets containing one of the RR type codes. This field specifies the meaning of the data in the RDATA field.
- CLASS two octets which specify the class of the data in the RDATA field.
- TTL a 32 bit unsigned integer that specifies the time interval (in seconds) that the resource record may be cached before it should be discarded. Zero values are interpreted to mean that the RR can only be used for the transaction in progress, and should not be cached.
- RDLENGTH an unsigned 16 bit integer that specifies the length in octets of the RDATA field.
- RDATA a variable length string of octets that describes the resource. The format of this information varies according to the TYPE and CLASS of the resource record. For example, the if the TYPE is A and the CLASS is IN, the RDATA field is a 4 octet ARPA Internet address.

Distributed, Hierarchical database

How to register a domain name

- Registrar
- Dynamic DNS