7.2.2. Resource Records

Every domain, whether it is a single host or a top-level domain, can have a set of **resource records** associated with it. For a single host, the most common resource record is just its IP address, but many other kinds of resource records also exist. When a resolver gives a domain name to DNS, what it gets back are the resource records associated with that name. Thus the real function of DNS is to map domain names onto resource records.

A resource record is a five-tuple. Although they are encoded in binary for efficiency, in most expositions resource records are presented as ASCII text, one line per resource record. The format we will use is as follows:

```
Domain_name  Time_to_live  Class  Type  Value
```

The *Domain_name* tells the domain to which this record applies. Normally, many records exist for each domain and each copy of the database holds information

about multiple domains. This field is thus the primary search key used to satisfy queries. The order of the records in the database is not significant. When a query is made about a domain, all the matching records of the class requested are returned.

The *Time_to_live* field gives an indication of how stable the record is. Information that is highly stable is assigned a large value, such as 86400 (the number of seconds in 1 day). Information that is highly volatile is assigned a small value, such as 60 (1 minute). We will come back to this point later when we have discussed caching.

The third field of every resource record is the *Class*. For Internet information, it is always *IN*. For nonInternet information, other codes can be used, but in practice, these are rarely seen.

The *Type* field tells what kind of record this is. The most important types are listed in Fig. 7-26.

Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

Fig. 7-26. The principal DNS resource record types.

An *SOA* record provides the name of the primary source of information about the name server's zone (described below), the email address of its administrator, a unique serial number, and various flags and timeouts.

The most important record type is the *A* (Address) record. It holds a 32-bit IP address for some host. Every Internet host must have at least one IP address, so other machines can communicate with it. Some hosts have two or more network connections, in which case they will have one type *A* resource record per network connection (and thus per IP address).

The next most important record type is the *MX* record. It specifies the name of the domain prepared to accept email for the specified domain. A common use of this record is to allow a machine that is not on the Internet to receive email from Internet sites. Delivery is accomplished by having the non-Internet site

make an arrangement with some Internet site to accept email for it and forward it using whatever protocol the two of them agree on.

For example, suppose that Cathy is a computer science graduate student at UCLA. After she gets her degree in AI, she sets up a company, Electrobrain Corporation, to commercialize her ideas. She cannot afford an Internet connection yet, so she makes an arrangement with UCLA to allow her to have her email sent there. A few times a day she will call up and collect it.

Next, she registers her company with the *com* domain and is assigned the domain *electrobrain.com*. She might then ask the administrator of the *com* domain to add an *MX* record to the *com* database as follows:

```
electrobrain.com  86400  IN  MX  1  mailserver.cs.ucla.edu
```

In this way, mail will be forwarded to UCLA where she can pick it up by logging in. Alternatively, UCLA could call her and transfer the email by any protocol they mutually agree on.

The *NS* records specify name servers. For example, every DNS database normally has an *NS* record for each of the top-level domains, so email can be sent to distant parts of the naming tree. We will come back to this point later.

CNAME records allow aliases to be created. For example, a person familiar with Internet naming in general wanting to send a message to someone whose login name is *paul* in the computer science department at M.I.T. might guess that *paul@cs.mit.edu* will work. Actually this address will not work, because the domain for M.I.T.'s computer science department is *lcs.mit.edu*. However, as a service to people who do not know this, M.I.T. could create a *CNAME* entry to point people and programs in the right direction. An entry like this one might do the job:

```
cs.mit.edu  86400  IN  CNAME  lcs.mit.edu
```

Like *CNAME*, *PTR* points to another name. However, unlike *CNAME*, which is really just a macro definition, *PTR* is a regular DNS datatype whose interpretation depends on the context in which it is found. In practice, it is nearly always used to associate a name with an IP address to allow lookups of the IP address and return the name of the corresponding machine.

HINFO records allow people to find out what kind of machine and operating system a domain corresponds to. Finally, *TXT* records allow domains to identify themselves in arbitrary ways. Both of these record types are for user convenience. Neither is required, so programs cannot count on getting them (and probably cannot deal with them if they do get them).

Finally, we have the *Value* field. This field can be a number, a domain name, or an ASCII string. The semantics depend on the record type. A short description of the *Value* fields for each of the principal records types is given in Fig. 7-26.

As an example of the kind of information one might find in the DNS database of a domain, see Fig. 7-27. This figure depicts part of a (semihypothetical)

database for the *cs.vu.nl* domain shown in Fig. 7-25. The database contains seven types of resource records.

```
; Authoritative data for cs.vu.nl
cs.vu.nl.          86400  IN  SOA     star boss (952771,7200,7200,2419200,86400)
cs.vu.nl.          86400  IN  TXT     "Faculteit Wiskunde en Informatica."
cs.vu.nl.          86400  IN  TXT     "Vrije Universiteit Amsterdam."
cs.vu.nl.          86400  IN  MX      1 zephyr.cs.vu.nl.
cs.vu.nl.          86400  IN  MX      2 top.cs.vu.nl.

flits.cs.vu.nl.    86400  IN  HINFO   Sun Unix
flits.cs.vu.nl.    86400  IN  A       130.37.16.112
flits.cs.vu.nl.    86400  IN  A       192.31.231.165
flits.cs.vu.nl.    86400  IN  MX      1 flits.cs.vu.nl.
flits.cs.vu.nl.    86400  IN  MX      2 zephyr.cs.vu.nl.
flits.cs.vu.nl.    86400  IN  MX      3 top.cs.vu.nl.
www.cs.vu.nl.      86400  IN  CNAME   star.cs.vu.nl
ftp.cs.vu.nl.      86400  IN  CNAME   zephyr.cs.vu.nl

rowboat                   IN  A       130.37.56.201
                          IN  MX      1 rowboat
                          IN  MX      2 zephyr
                          IN  HINFO   Sun Unix

little-sister             IN  A       130.37.62.23
                          IN  HINFO   Mac MacOS

laserjet                  IN  A       192.31.231.216
                          IN  HINFO   "HP Laserjet IIISi" Proprietary
```

Fig. 7-27. A portion of a possible DNS database for *cs.vu.nl*

The first noncomment line of Fig. 7-27 gives some basic information about the domain, which will not concern us further. The next two lines give textual information about where the domain is located. Then come two entries giving the first and second places to try to deliver email sent to *person@cs.vu.nl*. The *zephyr* (a specific machine) should be tried first. If that fails, the *top* should be tried next.

After the blank line, added for readability, come lines telling that the *flits* is a Sun workstation running UNIX and giving both of its IP addresses. Then three choices are given for handling email sent to *flits.cs.vu.nl*. First choice is naturally the *flits* itself, but if it is down, the *zephyr* and *top* are the second and third choices. Next comes an alias, *www.cs.vu.nl*, so that this address can be used without designating a specific machine. Creating this alias allows *cs.vu.nl* to change its World Wide Web server without invalidating the address people use to get to it. A similar argument holds for *ftp.cs.vu.nl*.

The next four lines contain a typical entry for a workstation, in this case, *rowboat.cs.vu.nl*. The information provided contains the IP address, the primary and secondary mail drops, and information about the machine. Then comes an entry for a non-UNIX system that is not capable of receiving mail itself, followed by an entry for a laser printer.

What is not shown (and is not in this file), are the IP addresses to use to look up the top level domains. These are needed to look up distant hosts, but since they are not part of the *cs.vu.nl* domain, they are not in this file. They are supplied by the root servers, whose IP addresses are present in a system configuration file and loaded into the DNS cache when the DNS server is booted. They have very long timeouts, so once loaded, they are never purged from the cache.