

Exploring DNS Using Dig

A Brief Introduction

by Derek Schuurman

Not so long ago, when the when the Internet was comprised of a limited number of hosts, there was a file called **/etc/hosts** contained information about every hosts on the network. This file was maintained and distributed across the Internet. This approach became impractical as the number of hosts grew large . A hierarchical, domain-based, naming scheme called DNS (the Domain Name System) was invented and is defined in RFC 1034 and 1035 . The DNS system uses UDP packets (Request/Reply) and a resolver is used to map names to an IP address by sending a request to a local DNS server. The **/etc/hosts** file is still found on some hosts, but is often used to store local addresses such as the address of the local machine.

The Internet is divided into various top level domains which are approved by ICANN. There are about a dozen root servers in the world which know the addresses of the top-level domain servers. These top-level domains are further subdivided into subdomains, which can be further partitioned, and so on. These domains on the Internet can be represented by a tree with the leaves of the tree representing hosts or groups of hosts.

Every domain has a set of resource records associated with it, and DNS servers use these resource records when replying to queries. When a host has a name query, it passes the query to one of the local name servers. If the name falls under the jurisdiction of the local name server, it returns the authoritative resource records. If the requested name cannot be satisfied locally and is part of a remote domain, then a recursive query can be made to a remote name server.

“Dig” is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers that are returned from the name server(s) that were queried. The “dig” utility is often used to troubleshoot DNS problems. A basic “dig” hostname query is made as follows:

```
dig @server hostname
```

where **server** is the DNS server, and **hostname** is the name to query. The server parameter is optional; if it is not given the default name server will be used. To perform a “reverse look-up” (ie. determine the hostname given an IP address) do the following:

```
dig -x 1 2.3.4
```

where **1.2.3.4** is the IP address of the host to query. Type “man dig” for a detailed description of how it can be used.

Questions:

1. What is the IP address of your name server?
2. Consider the hostname **www.redeemer.ca**:
 - a. What is the IP address of this host when queried inside the CS lab using the DNS server in the CS lab?
 - b. What is the IP address of this host when queried using the external DNS server?
3. Why are these IP addresses different?
4. What is a DNS **MX** record? What is the MX record for **redeemer.ca**?
5. Do DNS requests use the TCP or UDP transport protocol. Why does DNS use the transport protocol that it does?