Insider Tips: Dynamic DNS services

Matchmaking

There's only one thing wrong with some low-budget ADSL Internet connections. Providers tend to give you a temporary IP address and enforce a daily address change. That makes it difficult to provide services. Dynamic DNS services can solve this problem by using permanent name assignments.

BY MARC ANDRÉ SELIG

Every computer needs an IP address to communicate via the Internet Protocol. However, providers tend to be mean and give their customers dynamic addresses rather than static ones. When you open up an Internet connection, you are assigned an arbitrary address from your provider's address pool. This makes life difficult for server admins as they would need to inform their clients each time the address changes.

Providers have good reason to be mean, however. The current IPv4 uses four byte IP addresses. Theoretically, this would give you 256 to the power of four addresses, that is about four billion. Unfortunately, only a small fraction of these addresses are really available. This is partly due to specific protocol features (such as multicast addresses), but mainly to over-generous address reservations back in the early days of the Internet. Address space dimensioning is very tight for a global network, and that makes free address blocks hard to obtain nowadays.

Listing 1: Validity of DNS responses

```
;; QUERY SECTION:
;;      www.abcxyz.com, type = A, class = IN

;; ANSWER SECTION:
www.abcxyz.com. 8h10m41s IN CNAME abcxyzsu7.abcxyz.com.
abcxyzsu7.abcxyz.com. 2h39m44s IN A   136.199.85.18
```

Most providers resort to address pools with a lot less entries than they have customers. When a customer dials in, the provider assigns that customer a temporary address from its pool. The assignment is valid for exactly the duration of the customer's current dial-in session. When the customer quits the session, the address immediately becomes available for the next customer. In other words, the provider needs enough IP addresses to cover the maximum number of current connections.

Server Problems

This trick avoids running out of addresses, and has its advantages for the provider. Unfortunately, it has its downside from the customer's point of view. A dynamic address is fine, if you simply need to send HTTP queries to Web servers, but running a server is a different story. Browsers would need a method of discovering the server's current dynamic address.

This is not typically an issue that affects larger enterprises, who have leased lines and static IP assignments. Small businesses and home users tend to avoid Web hosting on their own machines and outsource this job to specialized providers.

In small to mid-size networks, there are a few tasks where a server with an ADSL connection is a useful option. An enterprise might decide to allow a consultant remote access to its computers to reduce management costs. Home users can use their cellphones to dial up the WAP server in their home offices to provide mobile email services. In an age where ADSL flat-rates or traffic volume based invoicing have become a matter of course, it is cheap, and it makes sense to leave machines online 24x7. This said, both the consultant and the cellphone user need to know the server's current IP address.

Dynamic DNS

The Domain Name Service (DNS) translates numerical IP addresses into symbolic hostnames and vice-versa. To use the Internet Protocol to access a computer called *www.abcxyz.com*, you need its IP address, *136.199.85.18*. A user requiring data from the server will typically make a note of, or memorize, the symbolic name, *www.abcxyz.com*. The user's computer queries DNS to discover the matching IP address. If the IP

Listing 2: Short-lived dynamic DNS data

```
;; QUERY SECTION:
;;      selig.dyndns.org, type = A, class = IN

;; ANSWER SECTION:
selig.dyndns.org.       51S IN A        80.128.96.209
```

changes, the service provider changes the DNS entry correspondingly, and when another browser looks for the name later, the new address is returned.

This system is also useful for documenting the current status of dynamic IP addresses. You can modify the DNS server entry for your own address each time it changes, allowing clients to query the latest address when they need access. There are a few things you need to be aware of, however.

Besides publishing name/address mappings, DNS servers also add validity data. This allows subordinate DNS servers to save bandwidth. If multiple clients request the same address within the validity period, the DNS will answer immediately with a cached response, without needing to query the authoritative name server.

Listing 1 shows an example of a manual DNS query using *dig www.abcxyz.com*. The hostname *www.abcxyz.com* is an alias for *abcxyzsu7.abcxyz.com*. The assignment will be valid for at least eight hours, ten minutes and 41 seconds. The IP address assigned to *abcxyzsu7.abcxyz.com* will not change for about three hours.

Unexpected Changes

Access problems can occur in case of unexpected changes. For example, if *abcxyzsu7.abcxyz.com* has an address from a dynamic pool, and the address changes before the 2 hours, 39 minutes and 44 seconds have elapsed, the client would continue using the cached response, just like any other machine that accesses the same name server. In contrast to this, name servers without a cached entry for *www.abcxyz.com* would immediately return the new entry.

The answer to this dilemma is to assign an extremely short validity period dynamic DNS entries. But this can cause a drastic increase in traffic to the DNS server. On the upside it means less potential for errors following an IP address change. Listing 2 shows a typical example of a dynamic entry. The output from *dig selig.dyndns.org* tells us that the address assignment is valid for only 51 seconds. In reality, the name server will tend to delete this entry from its cache even sooner.

Listing 3: */etc/ddclient.conf* example

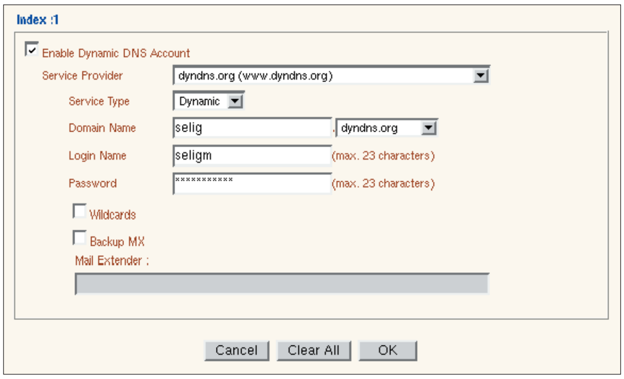
```
01 daemon=60        # Check IP
   address every 60 seconds
02 syslog=yes
03 mail=root
04 use=if, if=eth0
05 login=seligm
06 password=topsecret
07 server=members.dyndns.org,
   protocol=dyndns2
   selig.dyndns.org
```

Automatic Updates

A DNS server that publishes a dynamic addresses is not allowed to have a dynamic IP address itself. This would be a chicken or egg scenario. Fortunately, there are a number of large DNS service providers that map static addresses to dynamic IPs. I use DynDNS [1] and NO-IP [2], but there are many other similar services.

The next thing you need to take care of is getting your own machine to update its DNS data. The configuration files that admins normally feed to their name servers are quite complex. It would be too much trouble to update the files manually each time your IP changes. Special client software that automatically transmits the new IP address to the name server whenever the machine opens up an Internet connection makes more sense. There are a number of protocols that can handle this.

Many routers designed for use in small business or home offices have a dynamic DNS service client. Figure 1 shows an example. Any Linux system can automatically update its entries with equal ease. It makes little difference whether you access the Internet via a router, or use Linux's own routing capabilities with a modem, ISDN, ADSL or any other technology.



Index :1

☑ Enable Dynamic DNS Account

Service Provider: dyndns.org (www.dyndns.org)

Service Type: Dynamic

Domain Name: selig . dyndns.org

Login Name: seligm (max. 23 characters)

Password: *********** (max. 23 characters)

□ Wildcards

□ Backup MX

Mail Extender :

Figure 1: Many router appliances (Draytek shown here) have a client for DynDNS service. Simply use the Web interface to supply the required data to allow DNS-based access to your own server.

The choice of client software depends on the DNS provider. The *ddclient* [3] tool is a popular program used by various providers. Many Linux distributions include the client, Suse and Debian, for example. The client parses the */etc/ddclient.conf* configuration file (see Listing 3). In the simplest case, the program runs as a daemon and checks the network interface for IP address changes at regular intervals (line 4). In our example, the interval between checks is 60 seconds (line 1). More sophistication is required for hardware routers with private IP space in their LANs, but *ddclient* provides graceful support for these cases as well.

Whenever the address changes, *ddclient* updates the DNS server entries. It uses the protocol and the hostname configured in line 7 to do so. This causes the name server to respond with the current IP address of the machine, whenever it receives a query for *selig.dyndns.org*. This in turn allows the machine to provide services round the clock. ■

THE AUTHOR

Marc André Selig spends half of his time working as a scientific assistant at the University of Trier and as an ongoing medical doctor in a hospital in Augsburg, Germany. His current preoccupation is programming Web based databases on various Unix platforms.

INFO

[1] One of the oldest DynDNS services: *http://www.dyndns.org*

[2] Another DynDNS service: *http://www.no-ip.com*

[3] Perl client for DynDNS.org: *http://www.dyndns.org/services/custom/clients.html##3*