Dynamic DNS

Subjects: Telecommunications
Contributor: HandWiki Huang

Dynamic DNS (DDNS) is a method of automatically updating a name server in the Domain Name System (DNS), often in real time, with the active DDNS configuration of its configured hostnames, addresses or other information. The term is used to describe two different concepts. The first is "dynamic DNS updating" which refers to systems that are used to update traditional DNS records without manual editing. These mechanisms are explained in RFC 2136, and use the TSIG mechanism to provide security. The second kind of dynamic DNS permits lightweight and immediate updates often using an update client, which do not use the RFC2136 standard for updating DNS records. These clients provide a persistent addressing method for devices that change their location, configuration or IP address frequently.

Keywords: ddns ; dns ; tsig

1. Background

In the initial stages of the Internet (ARPANET), addressing of hosts on the network was achieved by static translation tables that mapped hostnames to IP addresses. The tables were maintained manually in form of the host file. The Domain Name System brought a method of distributing the same address information automatically online through recursive queries to remote databases configured for each network, or domain. Even this DNS facility still used static lookup tables at each participating node. IP addresses, once assigned to a particular host, rarely changed and the mechanism was initially sufficient. However, the rapid growth of the Internet and the proliferation of personal computers in the workplace and in homes created the substantial burden for administrators of keeping track of assigned IP addresses and managing their address space. The Dynamic Host Configuration Protocol (DHCP) allowed enterprises and Internet service providers (ISPs) to assign addresses to computers automatically as they powered up. In addition, this helped conserve the address space available, since not all devices might be actively used at all times and addresses could be assigned as needed. This feature required that DNS servers be kept current automatically as well. The first implementations of *dynamic DNS* fulfilled this purpose: Host computers gained the feature to notify their respective DNS server of the address they had received from a DHCP server or through self-configuration. This protocol-based DNS update method was documented and standardized in IETF publication RFC 2136 in 1997 and has become a standard part of the DNS protocol (see also nsupdate program).

The explosive growth and proliferation of the Internet into homes brought a growing shortage of available IP addresses. DHCP became an important tool for ISPs as well to manage their address spaces for connecting home and small-business end-users with a single IP address each by implementing network address translation (NAT) at the customer-premises router. The private network behind these routers uses address space set aside for these purposes (RFC 1918), masqueraded by the NAT device. This, however, broke the end-to-end principle of Internet architecture and methods were required to allow private networks, with frequently changing external IP addresses, to discover their public address and insert it into the Domain Name System in order to participate in Internet communications properly. Today, numerous providers, called *Dynamic DNS* service providers, offer such technology and services on the Internet.

1.1. Domain Name System

DNS is based on a distributed database that takes some time to update globally. When DNS was first introduced, the database was small and could be easily maintained by hand. As the system grew this task became difficult for any one site to handle, and a new management structure was introduced to spread out the updates among many domain name registrars. Registrars today offer end-user updating to their account information, typically using a web-based form, and the registrar then pushes out update information to other DNS servers.

Due to the distributed nature of the domain name systems and its registrars, updates to the global DNS may take hours to distribute. Thus DNS is only suitable for services that do not change their IP address very often, as is the case for most large services like Wikipedia. Smaller services, however, are generally much more likely to move from host to host over

shorter periods of time. Servers being run on certain types of Internet service provider, cable modems in particular, are likely to change their IP address over very short periods of time, on the order of days or hours. Dynamic DNS is a system that addresses the problem of rapid updates.

2. Types

The term DDNS is used in two ways, which, while technically similar, have very different purposes and user populations. The first is *standards-based DDNS*, which uses an extension of the DNS protocol to ask for an update; this is often used for company laptops to register their address. The second is *proprietary DDNS*, usually a web-based protocol, normally a single HTTP fetch with username and password which then updates some DNS records (by some unspecified method); this is commonly used for a domestic computer to register itself by a publicly known name in order to be found by a wider group, for example as a games server or webcam.

End users of Internet access receive an allocation of IP addresses, often only a single address, by their Internet service provider. The assigned addresses may either be fixed (i.e. static), or may change from time to time, a situation called *dynamic*. Dynamic addresses are generally given only to residential customers and small businesses, as most enterprises specifically require static addresses.

Dynamic IP addresses present a problem if the customer wants to provide a service to other users on the Internet, such as a web service. As the IP address may change frequently, corresponding domain names must be quickly re-mapped in the DNS, to maintain accessibility using a well-known URL.

Many providers offer commercial or free Dynamic DNS service for this scenario. The automatic reconfiguration is generally implemented in the user's router or computer, which runs software to update the DDNS service. The communication between the user's equipment and the provider is not standardized, although a few standard web-based methods of updating have emerged over time.

2.1. Standards-Based DDNS

The standardized method of dynamically updating domain name server records is prescribed by RFC 2136, commonly known as dynamic DNS update. The method described by RFC 2136 is a network protocol for use with managed DNS servers, and it includes a security mechanism. RFC 2136 supports all DNS record types, but often it is used only as an extension of the DHCP system, and in which the authorized DHCP servers register the client records in the DNS. This form of support for RFC 2136 is provided by a plethora of client and server software, including those that are components of most current operating systems. Support for RFC 2136 is also an integral part of many directory services, including LDAP and Windows' Active Directory domains.

2.2. Applications

In Microsoft Windows networks, dynamic DNS is an integral part of Active Directory, because domain controllers register their network service types in DNS so that other computers in the domain (or forest) can access them.

Increasing efforts to secure Internet communications today involve encryption of all dynamic updates via the public Internet, as these public dynamic DNS services have been abused increasingly to design security breaches. Standards-based methods within the DNSSEC protocol suite, such as TSIG, have been developed to secure DNS updates, but are not widely in use. Microsoft developed alternative technology (GSS-TSIG) based on Kerberos authentication.

Some free DNS server software systems, such as dnsmasq, support a dynamic update procedure that directly involves a built-in DHCP server. This server automatically updates or adds the DNS records as it assigns addresses, relieving the administrator of the task of specifically configuring dynamic updates.

DDNS for Internet access devices

Dynamic DNS providers offer a software client program that automates the discovery and registration of the client system's public IP addresses. The client program is executed on a computer or device in the private network. It connects to the DDNS provider's systems with a unique login name; the provider uses the name to link the discovered public IP address of the home network with a hostname in the domain name system. Depending on the provider, the hostname is registered within a domain owned by the provider, or within the customer's own domain name. These services can function by a number of mechanisms. Often they use an HTTP service request since even restrictive environments usually allow HTTP service. Most providers have an API similar to a first provider DynDNS (Dyn.com) so it's often called DynDNS2.

Many home networking modem/routers include client applications in their firmware, compatible with a variety of DDNS providers.

DDNS for security appliance manufacturers

Dynamic DNS is an expected feature or even requirement for IP-based security appliances like DVRs and IP cameras. Many options are available for today's manufacturer, and these include the use of existing DDNS services or the use of custom services hosted by the manufacturers themselves.

In almost all cases, a simple HTTP based update API is used as it allows for easy integration of a DDNS client into a device's firmware. There are several pre-made tools that can help ease the burden of server and client development, like MintDNS, cURL and Inadyn. Most web-based DDNS services use a standard user name and password security schema. This requires that a user first create an account at the DDNS server website and then configure the device to send updates to the DDNS server whenever an IP address change is detected.

Some device manufacturers go a step further by only allowing their DDNS Service to be used by the devices they manufacture, and also eliminate the need for user names and passwords altogether. Generally this is accomplished by encrypting the device's MAC address using an cryptographic algorithm kept secret on both the DDNS server and within the device's firmware. The resulting decryption or decryption failure is used to secure or deny updates. Resources for the development of custom DDNS services are generally limited and involve a full software development cycle to design and field a secure and robust DDNS server.
