DNS Explained

(Extractd from a tutorial created by Anthony Eden from DNSimple)

1. Introduction

Every time you visit a web site or send an email DNS is used, even without you knowing it. DNS is used to convert human-friendly domain names like dnsimple.cominto an address that software on your computer can use to communicate with another computer hooked up to the Internet. To convert a domain name into an Internet address your computer will first make a connection to a **Resolving Name Server** (or simply a resolver). A resolver is a computer that is close to you on the network which means you can get to it quickly.

When a resolver receives a DNS question (for example, what is the address for the name dnsimple.com) it will first look to see if it already has the answer in its **cache**. A cache is a place where the resolver stores previously found answers so it can give you back a response more quickly. If the answer is in the cache and it's not too old (more on this later) then you'll get the answer back really fast. If the answer is *not* in the cache then the resolver will try to go find the answer.

To find the answer for a domain not in its cache, a resolver starts by breaking down the domain name you provided into its parts. As an example, dnsimple.com will be broken down into "dnsimple", "com", and the root.

Wait, what is this root thing? The root is the top of the domain name hierarchy. Sometimes you might see a domain written like "dnsimple.com." - notice the period at the end, that represents the root. There are thirteen root name servers, spread around the world.

Once a domain has been broken down into its parts, the resolver start at the root name servers and ask one of them where to find "com". The root name server that was queried will respond with a bunch of name servers and the resolver will pick one and go ask that "com" name server where to find "dnsimple.com". That name server will then respond with multiple name servers that know about "dnsimple.com". Finally, the resolver will ask one of those name servers the original question it needed to ask. If the name server is able to answer the original question then it is called the **Authoritative Name Server**.

Once the authoritative name server provides the resolver with the answer to its original question, the resolver will then add that to its cache and send it to you. In most cases, all of this happens within less than a second, which is pretty neat. Future requests to the resolver for the same record will be answered directly from the resolvers cache, as long as the record isn't stale, keeping things nice and fast.

2. Respect Authority

Authoritative name servers are responsible for providing the actual answers to DNS questions. Whether you're planning on running your own name servers or using a DNS provider, you'll need to know how name server updates will behave when you make changes to your DNS records. Knowing how a DNS update works will save you lots of time and frustration and will help reduce the likelihood of downtime.

When you update a DNS record, you may have to wait for those changes to become visible to everyone in the world. There are several reasons that the record takes time to **propagate**.

The first reason is that the record has to get out into each of your authoritative name servers. If you are running your own name servers then you may be using zone transfers to get data out. If you're using a provider like DNSimple, then records will likely be deployed through proprietary mechanisms designed to reduce the time it takes to get records out to all of your authoritative name servers. In either case, you should keep this delay in mind when you make changes.

In addition to the time it takes for records to get pushed out to authoritative name servers, you also need to be aware that **resolvers** (which I mentioned in yesterday's email) may be caching DNS answers. The amount of time that the resolver may cache can vary depending on the record's **time-to-live (TTL)** value (more on this in a future lesson) as well as when the record was last put in the cache.

The best way to avoid having an outage is to plan for DNS changes accordingly. If you're going to move from one server to another, which requires a change to your A records (which I'll explain tomorrow), then you should make sure **both** systems are running until you are confident the DNS changes have propagated. Sure, it might take more effort to plan a change, but when you're making a significant change to yourDNS it helps to be prepared.

There's one other value that you might need to be aware of when adding new DNSrecords. That is the negative-TTL value, which is the last number in your SOA record. I'll write more about SOA records later, but for now just know that if you ask aDNS resolver a question and the authoritative server cannot answer, the resolver may cache that negative result for up to the amount of time defined by the negative-TTL value. Therefore, if the negative-TTL value is set to 600 (seconds) then it might take up to 10 minutes for the negative result to disappear. It's common that you will look up a name, find it's not in your DNS, add it and then wonder why it isn't working. The negative TTL may be the reason you're not seeing the new result, so keep that in mind.

3. Record Types

A DNS record defines how a DNS query for a particular name and type should be answered. For example, if a name server receives a question asking it about A records for the name dnsimple.com, then the server would need to have at least one A record pointing the name dnsimple.com to an IP address (since that's what A records do).

There are lots of different DNS record types, but for most people there are a few

which are most important:

- A records
- CNAME records
- MX records
- TXT records
- SPF records
- NS records
- SOA records

Let's briefly take a look at each of these.

As I mentioned above, **A records** are address records. They tell how to convert a host name to an IP address. All computers that are hooked to the Internet will have an IP address. In order for people to get to your web site, they'll need to be able to know the IP address where your web server is running. A records tell software what the IP address is for a given hostname.

CNAME records are used to point one hostname to another, canonical, name. You'll often see CNAME records used for your www host name (i.e. www.example.com is a CNAME for example.com). There's an important gotcha for CNAME records that you should keep in mind: they cannot appear along side any other record on the same host name. This includes NS and SOA records, which means you cannot use a CNAME on your **apex**. Fortunately some providers have created a new proprietary record type, called an ALIAS record, which helps solve this problem, but you'll need to use a DNS provider that supports it.

MX records explain what servers to use to deliver email that is sent to email addresses at a domain. For example, if I send email to john@example.com then a mail delivery agent, which is the bit of software that tries to deliver the email, will askexample.com for its MX records in order to figure out who it should talk to in order to complete delivery. You can, and probably should have multiple MX records for your domain in case one mail server is offline, but you'll need to make sure your email provider can provide you with multiple mail servers.

TXT records just hold text. They're used for things where no other record type exists and often will be proprietary. TXT records have their own little quirks, but one important one has to do with semi-colons. A semi-colon in a TXT record needs to be escaped (with a backslash in front of it), but often providers will automatically escape semi-colons. The point is this: make sure you know what your provider does: you don't want to escape the semi-colons if your provider does it for you because that will cause you trouble.

SPF records are used by mail servers to help reduce spam. The Sender Policy Framework (which is what SPF stands for) lets you tell your mail service who should be allowed to send mail for your domain. SPF records can get quite complicated so you'll want to work with your DNS and mail providers to make sure your SPF records are correct for your email setup. There are also some gotchas with SPF records, specifically that you may only have one SPF record on any host name (well, one SPF and one TXT that holds SPF data, since SPF records can be sent either with the SPF

type or the TXT type).

NS records have two purposes: first they are used to **delegate** a domain to authoritative name servers. More on this on day 5. The other purpose is to be used in the authority section of responses from your authoritative name servers. In most cases you won't need to worry too much about NS records, you just need to make sure that the name servers set by your domain registrar agree with the name servers returned from your authoritative name servers.

The final type I want to tell you about today is the **SOA record** type. SOA stands for Start of Authority and is the record that tells resolvers that your authoritative name server is, in fact, authoritative for your domain. It's content defines the master name server, the responsible person for the DNS, and then several numeric values that are used for zone synchronization when your name servers run in a master/slave setup. The best way to talk about SOA records is to show you one:

ns.example.com hostmaster.example.com (2003080800 172800 900 1209600 3600)

In this example, the primary name server is ns.example.com and the responsible person is hostmaster@example.com. Note that in the SOA record the @ symbol in the responsible person's email address is replaced with a period. As previously mentioned, the first four numbers inside the parenthesis are only used when you have a master/slave DNS setup. The final number is used to set a time-to-live value for *negative* responses. That is, when someone sends a question to a name server for a domain for which it is authoritative, but the name server does not have an answer to the question, it will return a response indicating that the record is not found. Resolvers may then cache this response for up to the number of seconds specified in the SOA negative cache value.

4. Time to Live

The time-to-live (TTL) value on a DNS record indicates how many seconds that record should stay in a resolving name server's cache. When the record is first requested from the resolver the resolver will start counting down from the TTL number. When the countdown reaches zero then the record will be removed from the resolver's cache and the next request for the same record will require a trip through the DNS resolution process.

Finding the right balance between too short and too long on your **time-to-live (TTL)**value for your DNS records can be a challenge. The right value often depends on how often a particular record changes. For example, your NS records (which will often be set automatically by your DNS provider) may have a TTL of a day or more. This is because the actual NS records don't usually change. The same applies for Start of Authority records (remember them from yesterday?).

On the other hand, you might want to set the TTL on your A record low if you are using a hosting company where the IP addresses might change automatically. For example, if you're using Heroku then keeping your TTL fairly low (anywhere from 10 minutes to an hour, depending on your needs) means that if Heroku removes an IP address, or makes a change, then you can change your record as well (or just use

the DNSimple ALIAS record and we'll handle that for you).

One other point about TTLs is that you don't have to think of them as permanent. You can keep a record's TTL up reasonably high (say 12 hours) if you don't usually change it, and only decrease it before you need to make the change. Just remember that you need to give caches enough time to clear, so if you change your TTL you'll need to wait at least the amount of time in the old TTL before completing your DNSchange.

5. Registration and Delegation

Before you can get started setting up your domain you need to register one and delegate it. Domain registration is the process of telling a **domain registry** that you would like to use an available domain name. Note that the domain must be available for you to register it. If someone else gets to the registry before you do and registers a domain, then it is theirs to use until it expires.

To register a domain, you must typically go through a **domain registrar**. A domain registrar is a company that is authorized to add domains to the domain registry on behalf of clients. For example, when you want to register a .com domain, you must go to an **ICANN-accredited registrar** or to a **domain reseller**. An accredited registrar is one that has gone through the accreditation process with ICANN. They must follow certain rules for how to register domain names for their clients. A domain reseller, on the other hand, goes through an accredited registrar to register domains. The same restrictions for apply to accredited registrars apply to resellers as well, however resellers do not have to go through the accreditation process.

When you register your domain you will need to provide various pieces of information about who will be the **registrant** for the domain (it's a person or an organization) as well as who is responsible for administration, billing and technical management of the domain. Some providers require that you enter this information every time you register a domain, while others make it easier and let you choose from information you've previously entered.

Once an available domain is registered, it is then **delegated**. Delegation is the process where name servers are placed in the registry's name servers to help tell resolvers where to look to find the DNS records for the domain. Without these delegation records, resolvers would never be able to get past the registry's name servers to find your DNS records.

Most registries require at least two name servers to be specified for each domain in the registry. Often, your registrar will automatically specify name servers when they register your domain with the registry, but you should always check that your domain is delegating properly and that its DNS is working using tools provided by your registrar or something like http://intodns.com/.