catchpoint™
(http://www.catchpoint.com)

July 1, 2014 DATA & ANALYSIS (HTTP://BLOG.CATCHPOINT.COM/CATEGORY/DATA-ANALYSIS/)

DNS Lookup: How a Domain Name is Translated to an IP Address

(https://www.facebook.com/sharer.php?u=http%3A%2F%2Fblog.catchpoint.com%2F2014%2F07%2F01%2Fdns-lookup-domain-name-ip-address%2F)

(https://twitter.com/share?url=http%3A%2F%2Fblog.catchpoint.com%2F2014%2F07%2F01%2Fdns-lookup-domain-name-ip-address%2F&text=I just saw%20DNS%20Lookup%3A%20How%20a%20Domain%20Name%20is%20Translated%20to%20an%20IP%20Address%20-%20Click to see also%20☛%20)

(https://www.linkedin.com/shareArticle?mini=true&url=http%3A%2F%2Fblog.catchpoint.com%2F2014%2F07%2F01%2Fdns-lookup-domain-name-ip-address%2F&title=DNS%20Lookup%3A%20How%20a%20Domain%20Name%20is%20Translated%20to%20an%20IP%20Address)

At Catchpoint, we believe that fast DNS (Domain Name System) is just as important as fast content. DNS is what translates your familiar domain name (www.google.com) into an IP address your browser can use (173.194.33.174). This system is fundamental to the performance of your webpage, yet most people don't fully understand how it works. Therefore, in order to help you better understand the availability and performance of your site, we will be publishing a series of blog articles to shed light on the sometimes complex world of DNS, starting with the basics.

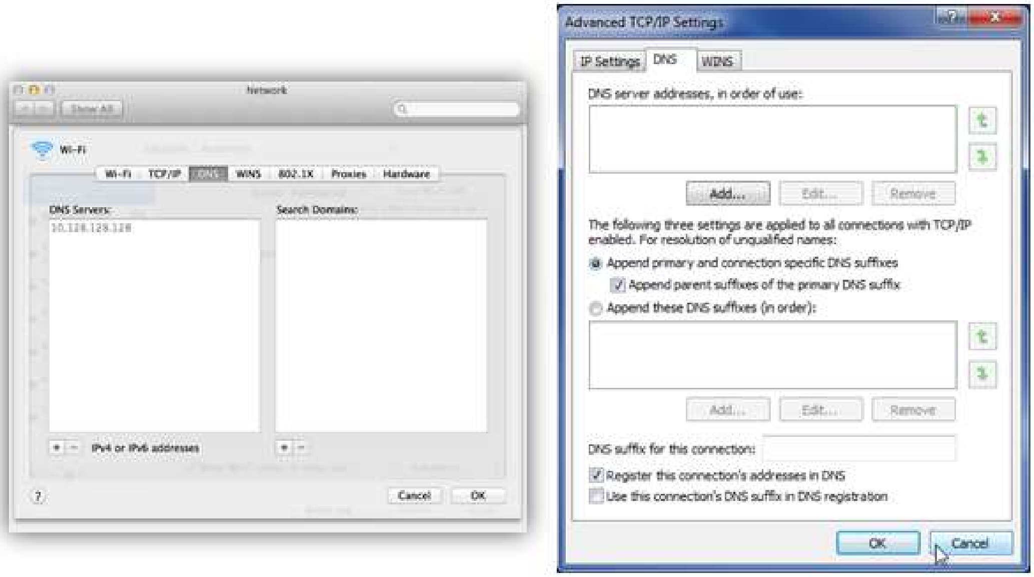
For the sake of simplicity, this article is assuming there was no DNS cached anywhere, hence this is a worst case scenario. We will tackle DNS caching in future articles.

Before the page and any resource on the page is loaded, the DNS must be resolved so the browser can establish a TCP connection to make the HTTP request. In addition, for every external resource referenced by a URL, the DNS resolution must complete the same steps (per unique domain) bef e I Resolution process starts when the user types a URL address on th he er asks the operating system for a specific page, in this case google.com.

Step 1: OS Recursive Query to DNS Resolver

Since the operating system doesn't know where “www.google.com” is, it queries a DNS resolver. The query the OS sends to the DNS Resolver has a special flag that tells it is a “recursive query.” This means that the resolver must complete the recursion and the response must be either an IP address or an error.

For most users, their DNS resolver is provided by their Internet Service Provider (ISP), or they are using an open source alternative such as Google DNS (8.8.8.8) or OpenDNS (208.67.222.222). This can be viewed or changed in your network or router settings. At this point, the resolver goes through a process called recursion to convert the domain name into an IP address.


Network
Show All
Wi-Fi
Wi-Fi
TCP/IP
DNS
WINS
802.1X
Proxies
Hardware
DNS Servers:
10.128.128.128
Search Domains:
IPv4 or IPv6 addresses
?
Cancel
OK
Advanced TCP/IP Settings
IP Settings
DNS
WINS
DNS server addresses, in order of use:
Add...
Edit...
Remove
The following three settings are applied to all connections with TCP/IP enabled. For resolution of unqualified names:
● Append primary and connection specific DNS suffixes
☑ Append parent suffixes of the primary DNS suffix
○ Append these DNS suffixes (in order):
Add...
Edit...
Remove
DNS suffix for this connection:
☑ Register this connection's addresses in DNS
□ Use this connection's DNS suffix in DNS registration
OK
Cancel

(http://blog.catchpoint.com/wp-content/uploads/2014/06/Settings.png)

DNS Settings on a Mac (left) and Windows Settings for IPv4 Protocol of the network connection (right).

Step 2: DNS Resolver Iterative Query to the Root Server

The resolver starts by querying one of the root DNS servers (http://www.iana.org/domains/root/servers) for the IP of “www.google.com.” This query does not have the recursive flag and therefore is an “iterative query,” meaning its response must be an address, the location of an authoritative name server, or an error. The root is represented in the hidden trailing “.” at the end of the domain name. Typing this extra “.” is not necessary as your browser automatically adds it.

There are 13 root server clusters named A-M with servers in over 380 locations (http://www.root-servers.org/). They are managed by 12 different organizations that report to the Intern[illegible] s[illegible] Verisign, who controls the A and J clusters. All of the servers are copies of one master server run by IANA.

Step 3: Root Server Response

These root servers hold the locations of all of the top level domains (http://data.iana.org/TLD/tlds-alpha-by-domain.txt) (TLDs) such as .com, .de, .io, and newer generic TLDs such as .camera.

The root doesn't have the IP info for "www.google.com," but it knows that .com might know, so it returns the location of the .com servers. The root responds with a list of the 13 locations of the .com gTLD servers, listed as NS or "name server" records.

Step 4: DNS Resolver Iterative Query to the TLD Server

Next the resolver queries one of the .com name servers for the location of google.com. Like the Root Servers, each of the TLDs have 4-13 clustered name servers existing in many locations. There are two types of TLDs: country codes (ccTLDs) run by government organizations, and generic (gTLDs). Every gTLD has a different commercial entity responsible for running these servers. In this case, we will be using the gTLD servers controlled by Verisign, who run the .com, .net, .edu, and .gov among gTLDs.

Step 5: TLD Server Response

Each TLD server holds a list of all of the authoritative name servers for each domain in the TLD. For example, each of the 13 .com gTLD servers has a list with all of the name servers for every single .com domain. The .com gTLD server does not have the IP addresses for google.com, but it knows the location of google.com's name servers. The .com gTLD server responds with a list of all of google.com's NS records. In this case Google has four name servers, "ns1.google.com" to "ns4.google.com."

Step 6: DNS Resolver Iterative Query to the Google.com NS

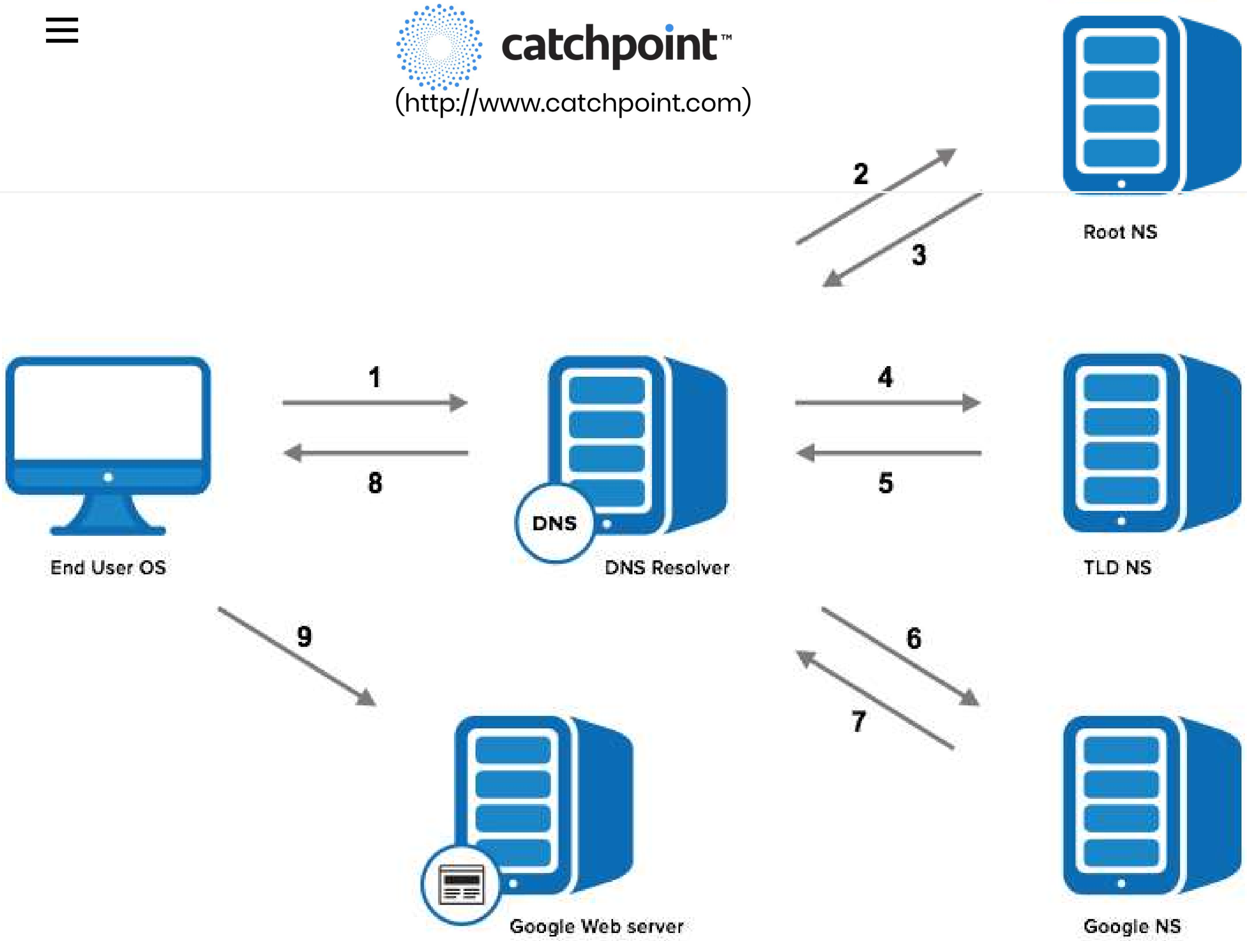
Finally, the DNS resolver queries one of Google's name server for the IP of "www.google.com."

Step 7: Google.com NS Response

This time the queried Name Server knows the IPs and responds with an A or AAAA address record (depending on the query type) for IPv4 and IPv6, respectively.

Step 8: DNS Resolver Response to OS

At this point the resolver has finished the recursion process and is able to respond to the end user's operating system with an IP address.



(http://blog.catchpoint.com/wp-content/uploads/2014/06/dns101.jpg)

Step 9: Browser Starts TCP Handshake

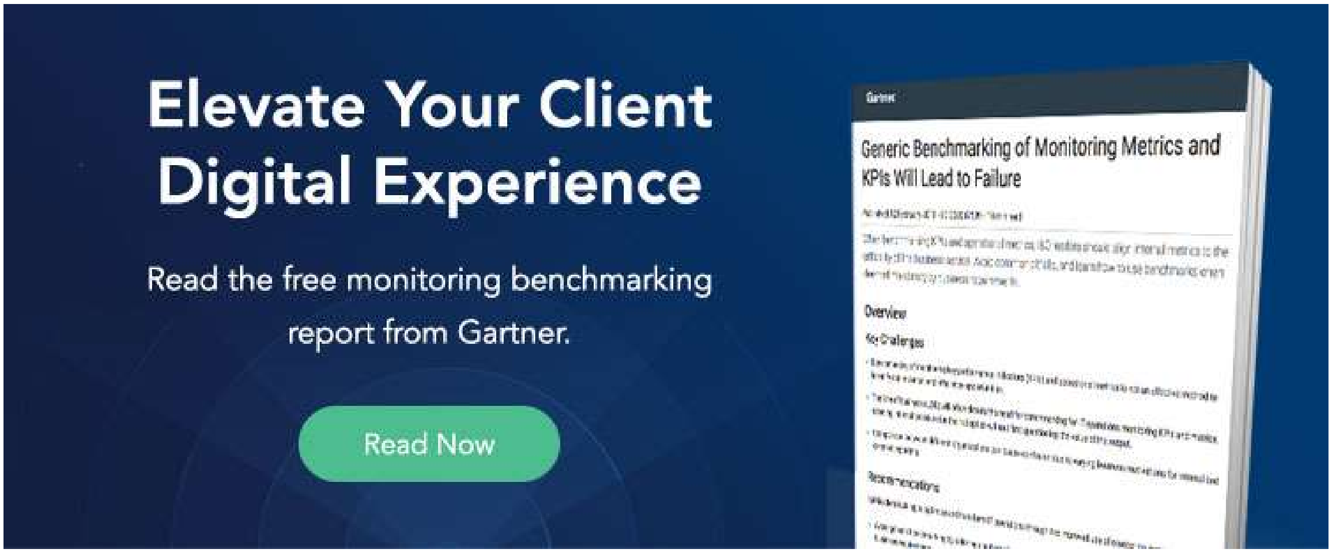
At this point the operating system, now in possession of www.google.com's IP address, provides the IP to the Application (browser), which initiates the TCP connection to start loading the page. For more information of this process, we wrote a blog post on the anatomy of HTTP (http://blog.catchpoint.com/2010/09/17/anatomyhttp/).

As mentioned earlier, this is worst case scenario in terms of the length of time to complete the resolution. In most cases, if the user has recently accessed URLs of the same domain, or other users relying on the same DNS resolver have done such requests, there will be no DNS resolution required, or it will be limited to the query on the local DNS resolver. We will cover this in later articles.

In this DNS non-cached case, four sets of DNS servers were involved, hence a lot could have gone wrong. The end user has no idea what is happening behind the scenes; they are simply are waiting for the page to load and all of these DNS queries have to happen before the browser can request

This is why we stress the importance of fast DNS. You can have a fast and well-built site, but if your DNS is slow, your webpage will still have poor response time.

Learn more about elevating your client digital experience by reading the report, *Generic Benchmarking of Monitoring Metrics and KPIs Will Lead to Failure* (http://pages.catchpoint.com/Gartner-Benchmarking-KPIs-Report.html).


Elevate Your Client Digital Experience

Read the free monitoring benchmarking report from Gartner.

Read Now

Generic Benchmarking of Monitoring Metrics and KPIs Will Lead to Failure

(http://pages.catchpoint.com/Gartner-Benchmarking-KPIs-Report.html)

DNS MONITORING (HTTP://BLOG.CATCHPOINT.COM/TAG/DNS-MONITORING/)

(http://blog.catchpoint.com/author/yng/)

AUTHOR

Yu Ng (http://blog.catchpoint.com/author/yng/)

⟵ PREVIOUS POST (HTTP://BLOG.CATCHPOINT.COM/2014/06/25/UNDERSTANDING-ANATOMY-WIRELESS-CONNECTION/)

NEXT POST ⟶ (HTTP://BLOG.CATCHPOINT.COM/2014/07/09/DISSECTING-DNS-COMMUNICATIONS/)

You Might Also Like

(http://blog.catchpoint.com/2018/09/25/best-practices-for-a-fast-performing-banking-website/)

Best practices for a fast performing banking website (http://blog.catchpoint.com/2018/09/25/best-practices-for-a-fast-performing-banking-website/)

(http://blog.catchpoint.com/2018/06/14/analyzing-impact-public-dns-resolver-outage/)

Analyzing the Impact of a Public DNS Resolver Outage (http://blog.catchpoint.com/2018/06/14/analyzing-impact-public-dns-resolver-outage/)

(http://blog.catchpoint.com/2018/06/11/practicality-of-observability/)