Website Analysis: Cambridge Dictionaries Online

Ballarini Robin

I. Introduction

This report will describe the results of a chosen website’s analysis. In this case, the website is Cambridge Dictionaries Online and is accessible at http://dictionary.cambridge.org/.

First, this report will describe the website’s Domain Name System (DNS). This will cover the website’s main server as well as the servers which provide extra resources or advertisements.

Next, we will go over the HTTP requests and responses that are exchanged between client and server when accessing this website. More specifically, this report will analyse the type of resources that are retrieved, the HTTP header lines and the use of cookies.

The last part will explain how the website uses TCP. This includes which TCP options are used, how the site closes TCP connections as well as the use of parallel connections. The section will end with a few statistics related to performance.

II. Domain Name System

In this section, the DNS queries on the site’s web servers will be detailed. Note that since the site accessed more than 60 different domain names, only a few of them will be explained, along with their particularities.

The domain names that will be explained below were chosen because they use different CDNs1 or because they are important for the website. They are displayed below along with their CDN.

- dictionary.cambridge.org: no CDN
- pagead2.googlesyndication.com: Google
- stats.g.doubleclick.net: Google
- mc.yandex.ru: no CDN
- edge.quantserv.com: Akamai
- cambridgewords.files.wordpress.com: no CDN

For each of these domain names, we will go over their IPv4 (A) and IPv6 (AAAA) DNS record, their use of aliases (CNAME) and the TTL2 of the records. We will also see that some records change depending on which DNS resolver is used.

Since the TTL provided by Dig is the remaining time to live. The TTLs given in the following sections are the maximum TTLs observed and, as such, are an approximation of each record’s total TTL.

A. dictionary.cambridge.org

This is the website’s primary domain name. It makes use of a single alias: prod.cdo.pitchleads.com.

When using Dig on this alias, a single IPv4 address is returned. On the other hand, there is no AAAA record for this domain.

When the DNS resolver changes, the alias remains the same. However, the IPv4 address changes. Across all the DNS requests made, two different IPv4 addresses were retrieved:

- 54.72.24.212
- 54.83.42.186

In addition, we can note that the maximum TTL seen is 600 seconds for the CNAME record and 30 seconds for the IPv4 record.

B. pagead2.googlesyndication.com

This a domain used by the website. As previously indicated, it has Google as CDN. As such it’s expected that the results of a DNS request will vary depending on the resolver.

The domain uses an alias: pagead46.1.doubleclick.net and provides both IPv4 and IPv6 records. More specifically, it actually provides four A records and a single AAAA record for a given DNS resolver. It can also be noted that the A and AAAA records depend on which resolver is used while the CNAME doesn’t.

The maximum TTL is 300 seconds for all three types of record (CNAME, A, AAAA).

C. stats.g.doubleclick.net

Just like the previous domain, this one also has Google as CDN.

The domain uses an alias: stats1.doubleclick.net and provides both IPv4 and IPv6 records. More specifically, it actually provides four A records and a single AAAA record for a given DNS resolver. Also, the A and AAAA records depend on the resolver while the CNAME doesn’t.

The maximum TTL observed is 300 seconds for the A and AAAA record. It’s around 86000 seconds for the CNAME record.
D. mc.yandex.ru
This domain has no visible CDN, unlike the previous two.

According to Dig, it provides four different A records and a single AAAA one, which all remain the same regardless of the DNS resolver used. Also, it doesn’t make use of a alias.

The maximum TTL observed is around 600 seconds for the AAAA record and 300 seconds for the A records.

E. edge.quantserve.com
This domain has a CDN other than Google: Akamai.

This domain uses a hierarchy of CNAMEs, which varies depending on the resolver. Here are the two hierarchies that were found:

- map-js.quantserve.com.akadns.net
  - anycast-americas.quantserve.com.akadns.net
- map-js.quantserve.com.akadns.net
  - edge.quantserve.com.edgesuite.net
  - a1811.g.akamai.net

We can note that the number of CNAME records varies but that the first CNAME remains the same. Furthermore, it seems the number of A records and their TTL varies depending on the hierarchy of CNAMEs. The smaller hierarchy provides two A records with a TTL of 20 seconds. The bigger one provides as many as eight, with a TTL of over 7000 seconds. There are no IPV6 records however.

The TTLs of the CNAMEs vary between 300 and 21000 seconds and the A records vary depending on the DNS resolver.

F. cambridgewords.files.wordpress.com
The last domain described in this report has no visible CDN.

DNS requests on this domain show the presence of two A records that do not seem to change depending on the DNS resolver. Furthermore, there is no CNAME, nor AAAA record.

The maximum TTL observed is around 285 seconds for the A records.

III. HTTP REQUESTS AND RESPONSES
In this second section of the report, the HTTP requests and responses exchanged with the website will be analysed.

When accessing http://dictionary.cambridge.org/, resources are retrieved. The types of those resources include html, css, javascript, png, jpg, gif and mp3. Other basic information is that the website as a whole uses HTTP 1.1 exclusively. Furthermore, according to Wireshark, the port number used is 80.

A. Typical use: searching for a word
Since the studied website is an online dictionary, a typical scenario is searching for a word.

The first step is the auto-completion feature, which has its own GET request. In particular, the request’s header has the following lines:
- X-Requested-With: XMLHttpRequest
- Referer with contains the current page’s url.

The response indicates a “text/javascript” content-type.

At first sight, it looks like a new request is sent for a new url each time a letter is typed. While this is true to some extent, there will not be a new request if the same letters have been auto-completed recently. There are no explicit references to using caches in the request and response headers however.

The second step consists of request to an intermediate url such as http://dictionary.cambridge.org/search/british/direct/?q=cat. The response has a header line labelled "location" which points to another url and a "content-length" line of 0 bytes.

The third and final step is a request to the url from the "location" header line in the previous response. The new response has an html file attached to it. This html file corresponds to the web page for the word the user was searching for. It’s interesting to note that the html page is encoded in gzip format according to the "Content-Encoding" header line and that, of course, other resources are retrieved with other requests when loading the page.

B. Header lines
The http requests and responses with the web servers sometimes use non-standard lines in their headers. This section contains a non-exhaustive list of such non-standard lines and attempts to document them. The provided urls are meant to be references to requests that use them.

A list of non-standard header lines found in requests:
- DNT: corresponds to the browser’s "Do not track" option.
- X-requested-with: used to identify Ajax requests. The site’s auto-completion feature uses this.

A list of non-standard header lines found in responses:
- X-XSS protection: Cross-site scripting filter. (http://pagead2.googlesyndication.com/)
- X-Content-Type-Options: prevents a type sniffing problem when using internet explorer.(http://pagead2.googlesyndication.com/)
- X-Cache: Indicates if the requested element was found in the website’s cache. (http://dictionary.cambridge.org/)
- X-Powered-By: technology that supports the web application. (http://sync.tidaltv.com)
- X-Frame-Options: allows or disallows a browser to render a page in a <frame>, <iframe> or <object> tag. Sites can use this to avoid
Some experimental header lines used by Google were excluded from the two previous lists. Two examples of such lines are "Alternate-Protocol" and "X-Firefox-Spdy".

C. Interesting uses of header lines

Some uses of HTTP header lines are a bit more interesting than their standard use. These uses will be explained in this section of the report.

The first use that’s worth mentioning is actually a small mistake. The "Content-Type" of an HTTP response is supposed to contain the MIME type for whatever content is attached to it. As of RFC 4329, the MIME type for javascript is supposed to be "application/javascript". When using the site however, we can find responses that contain the deprecated "text/javascript" alongside responses that use the new MIME type.

The second interesting point the use of "Cache-Control" with the argument "max-age" alongside "Expires". Both those header lines do similar things: they indicate an amount of time after which the resource should be fetched from the remote server again, instead of fetched from the local cache. For HTTP 1.1, "Cache-Control" with "max-age" overrides "Expires". On the other hand, it’s "Expires" that’s used with HTTP 1.0. Another thing to note is that request often contain the "Cache-Control" header line with "max-age=0" as argument. This impedes the use of the local cache.

The "Content-Language" line in HTTP responses seems to always contain "en" for the English language. This happens even if the website is displayed in French or any other available language.

The HTTP responses from Google have a P3P header that summarizes the privacy policy used.

D. Cookies

The studied website uses cookies. Most of the information found inside them is unreadable but the small amount that’s readable offers interesting insight.

The first piece of readable information is very simple: it contains the user’s localization. For example, the "localisation" line of the cookies could contain "BE".

The second piece of readable information contains a list of the user’s favourite dictionaries. Interestingly enough, the information in the cookies seems to match which dictionaries are displayed on a short-cut toolbar. More specifically, each time a dictionary is used, it’s placed on top of the list and removed from its old position in the list. The first three dictionaries listed in the cookie are displayed in the short-cut bar. See figure 1 to see what’s referred to as the short-cut bar.

IV. TRANSMISSION CONTROL PROTOCOL

In this section, the report will describe several characteristics of the website’s TCP. Note that this analysis only takes the main domain into account (advertisement/secondary servers are ignored). In order, we will go over how the TCP connection is opened and with which TCP options. This will be followed by a description on how connections are closed.

Then, this section will end with an explanation on parallel TCP connections as well as some performance statistics for TCP connections.

A. TCP options

The TCP connections are opened with a three way handshake:

- The client sends a SYN segment.
- The server replies with a SYN+ACK segment.
- The client sends an ACK to the SYN+ACK segment.

It can be noted that there are several option values exchanged between the two hosts in the SYN and SYN+ACK segments. Using Wireshark, we can retrieve those values:

- Selective acknowledgements are permitted. (SACK_PERM=1)
- The maximum segment size agreed upon is 1460 bytes. (MSS=1460)
- The connections use timestamps to identify each segment by using TSval and TSecr.
- The explicit congestion notification flag is set to 0 (Not-ECT)
- The window scaling factor for the website’s server is set to 32.

B. Closing a connection

Connections are closed using a graceful release. In other words,

- The server sends a FIN+ACK segment.
- The client relies with another FIN+ACK segment.
- The server ends the connection release with an ACK segment.

In some cases, the connections use a RST segment to close. This seems to happen when connection speed is poor.

C. Parallel connections

The website makes use of multiple parallel TCP connections. In the TCP dumps obtained with Wireshark, the number of parallel connections seems to vary between 1 and 4. This number seems to depend on the connection speed/quality.
<table>
<thead>
<tr>
<th>Statistic</th>
<th>To website</th>
<th>From website</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max RTT (in ms)</td>
<td>29.9</td>
<td>11.36</td>
</tr>
<tr>
<td>Unacknowledged data (in bytes)</td>
<td>141</td>
<td>304</td>
</tr>
<tr>
<td>Max window size (in bytes)</td>
<td>53833</td>
<td>21184</td>
</tr>
</tbody>
</table>

Table I. PERFORMANCE STATISTICS

D. Performance

Using tcptrace, it was possible to get some statistics on the TCP’s performance. They are visible in Table I. It’s important to know that these statistics were taken from a typical use of the website and not by clicking randomly on every visible button.

In addition, it should be noted that, while retransmissions do happen, they occur very rarely. Furthermore, using the Initcwnd checker tool properly was impossible as the analysed website doesn’t not have any file bigger than 70 KB. Nevertheless, for the tested url, it indicated 10 packets (13381 bytes) in the first roundtrip.

V. Tools

Several tools were used to obtain the information written in this report:

- **Developer extensions** in the web browser
- **Wireshark**: https://www.wireshark.org/
- **Dig**
- **tcptrace**: [http://www.tcptrace.org/](http://www.tcptrace.org/)

VI. Conclusion

In conclusion, this analysis shows that even small and seemingly simple websites have a lot hidden behind them. Indeed, while the TCP analysis provided expected results for the most part, the DNS and HTTP analyses showed that the website depends on a lot of other elements.

The website itself uses CNAMEs and has multiple IPV4 addresses. Furthermore, it depends on a lot of Google services, which also have interesting DNS records.

The analysis also uncovered peculiar uses of typical HTTP header lines as well as a few header lines which were just plain strange in and of themselves. For instance, there are experimental header lines used by Google such as “X-Firefox-Spdy” and ”Alternate-Protocol”.