LINGI2141: Analysis a website

Mehdi Dumoulin, Université Catholique de Louvain

Abstract—This paper will provide an analysis of a remote website. We analyse the HTTP requests, DNS and TCP connections of belgianrail.be.

Keywords—Protocol, TCP, HTTP, DNS, Network, Application Layer.

I. INTRODUCTION

T HIS paper will analyse in details a remote website by observing its responses over the Internet. The selected site is belgianrail.be. This is the website of belgium train company. This is a website with a lot of visit so it’s a good choice for analysing. In this paper we focus on HTTP request, DNS request and TCP connections on our website.

II. ANALYSIS HTTP REQUESTS

One component of the world wide web is the HyperText Transfer Protocol (HTTP). HTTP is a text-based protocol where a client sends a request and the server sends a response. In this paper, we will analyse this requests and this responses. In particular we will analyse all the domain names that are accessed when we go on the website, all the different types of resources that are retrieved, the port numbers used and each packet http sent or received deeply.

A. Domain names, Port numbers and Resources

When we go on belgianrail.be several domain names are accessed : belgianrail.be of course for the content, but we have also addthis.com for social bookmarking, google-analytics.com for analysis traffic on the website, youtube.com for videos, maps.google.com for the location and sealserver.trustwave.com for the SSL certificates.

The port numbers used are 80 for domain names in HTTP (e.g. belgianrail.be) and 443 for domain names in HTTPS (e.g. maps.google.com).

When we retrieve a page from this website several types of resources are request by the client : we have CSS for the layout of the website, HTML for representing the website and javascript for the script. For (animated) pictures we have gif, png and jpeg.

B. HTTP Request

In each HTTP request we have for this website at least:

- A method: GET if the website want to get content or POST if the website want to send content for processing by server. Other method exist also like HEAD or PUT but they aren’t used for this website. The method is followed by the part of URI (Uniform Resource Identifier = string used to identify a name of resource) of the requested resource and the version of HTTP used by the client (here always 1.1).
- A Host : contains the domain name of URI being requested.
- User-Agent : contains information about the software client.

Of course, we encounter other field in the HTTP request according to the request:

- Accept, Accept-Language, Accept-Encoding: those fields specifies what the requests expects to receive. The first is for the format of data, or more correctly MIME-types (see resources in previous section), the second for the natural languages accepted (e.g. french) and the last for the encoding (e.g. gzip).
- Referer: specify the address of the resource from wich the URI requested was obtained. This is generally used when the website retrieves content from other domain names e.g. the domain name sealserver.trustwave.com for the picture of SSL certificate, from youtube or google maps and from addthis. This field can violate privacy because the visited site can know what you visited before.
- Connection: this is used, with keep-alive, to indicate that the client expects the TCP connection to be persistent. Used with close it’s to close the connection at the end of the HTTP response. In our website the client sent always a keep-alive to the server.
- Content-Type & Content-Length: it’s used when the HTTP request is a POST method. The first field is for the type of data used and the second for the size of the entity-body in octets. Our site uses application/x-www-form-urlencoded as the content-type because we have several forms for routes, schedule and ticket.

C. HTTP Response

In each HTTP request we have for this website at least:

- HTTP version: with our website we have only HTTP 1.1
- Status code: we have the 200 code when the request is OK, the 302 code when the object required has been moved (e.g. when we access to belgianrail.be the servers told us to look at www.belgianrail.be) or 404 when the resource is not found (e.g. This happened with some pictures).

Other field we encountered :

- Location: when the resource has moved the server respond with the new location so the client requests again the same resource but at a different address.
- Connection: this field is always "close" with the belgianrail.be server. This is maybe because the server
does not support the persistent connection. On the server of addthis or trustwave the server responds with a "keep-alive".

- **Cache-Control and Pragma**: specify the directives that must be obeyed by all caching mechanisms. When max-age = 0 it’s for revalidate its own cache entry. Used with private means must be not cached in a shared cache. Used with a max-age is the number of seconds the resource will be considered fresh.

- **Expires**: this field gives the date/time after which the response is considered stale.

- **Content-Type & Content-Length & Content-Encoding**: same function as http request.

- **X-Powered-By**: it’s for specified the technology supporting web application. Here we have PHP/5.1.5 for trustwave and ASP.Net for other domain names.

- **Date**: represents date and time at which the message was originated.

- **Etag & Last-Modified**: it’s for caching purpose. It’s used for validation.

- **Content-Disposition**: Used with inline and the file name the browser will try to open the content within browser. Here it’s used with several pictures.

- **Via**: The Via general-header must be used by gateways and proxies to indicate the intermediate protocols and recipients. Here it’s used with Varnish: an HTTP accelerator capable of delivery cached content fast. This is used for the domain names addthis.com

- **Server**: when our website requests content from maps.google.com the server responds with the field server for specified the software used by the origin sever (e.g. : mafe, Apache).

- **X-XSS-Protection & X-Frame-Options**: Those fields are used with google server for safety. The first used to be sure the Cross-site scripting (security breach allowing to inject content on website) is enable on the browser. The second is to provide Clickjacking protection.

- **Alternate-Protocol**: 80:quic,p=0.002: this is used by Google for its experimental protocol QUIC.

- **x-content-type-options: nosniff**: This is used with a google server, this is for prevent content sniffing (method use to attempt to determine the "real" content type of a response by looking at the content itself and no the header content-type value).

### D. Cookies & Cache

A cookie is like a small text file stored on the terminal client. It allows to store user information or permit some features. For our website two tests are made: with and without cookies and cache on the browser (from previous navigation on this website).

Without cookies and cache the client sends much more request because the browser has to load from server all the content of website. First we have to choose the language used on the website (because there is no cookie for tell to server what language the user prefers). At this time we have four cookies: SC_ANALYTICS_SESSION_COOKIE, SC_ANALYTICS_GLOBAL_COOKIE, SC_ANALYTICS_SESSION_COOKIE, ASP.NET_SessionId (to identify the user on the server) and a default language (English). Once the language choose the client sends with a POST method the language to the server with a field **Cookie**. This field has several informations :

- **SC_ANALYTICS_GLOBAL_COOKIE**: it’s a persistent session cookie for identify the repeat visit of a single user.
- **SC_ANALYTICS_SESSION_COOKIE**: To identify a sequence of HTTP requests from a single user.

This field is provided by Sitecore Analytics that provides web content management and multichannel marketing.

We are also the content of form (ie choice of language) and the content-length. The server responds with a field **Set-Cookie** that saves the language selected by user. Then the client sends a lot of request to load all the content of the website (e.g. pictures, forms). The server responds with the different content and specify the lifetime of the resource. Some cookie have a session lifetime while some cookie have a time expiration.

When we reload the website later, with the same browser, we have 17 request (279Kb) instead of 77 request (2,1Mb) (test made on Firefox). The first requests to the server is the same, the browser sends a request to the domain name www.belgianrail.be. The difference is the cache. Now we have already the language choice (thanks to cookies) we go directly on the home page and we requests only few content. The rest is already in the cache. Cache is a temporary storage on client’s machine for quicker loading pages and contains pictures, movies, ... whereas cookie stored users preferences. For example, in Firefox, with cache, 34 answer on 73 are from cache. The client sends a request with a field **If-None-Math** and **If-Modified-Since** to identify the resource with the first field and for verify if the resource was modify since we save it in cache. The server responds with a 304 status if the resource was not modified otherwise the new resource is returned by the server.

### E. In practice

Let’s test HTTP requests with a concrete example: A route. After having completed the forms, the client sends a http request with the POST method with a cookie, the forms and other usual field. The server responds with a 302 status : FOUND and a new address for redirection with requested schedule. The cookie is useful in particular for the selected language. Of course the browser continues to obtain the rest of content the pages with pictures, CSS, HTML... with request to the server with GET method.

### F. Problem Encountered

Sunday 14 December 2015 morning the site encountered a problem: on a blank browser (either Firefox or Chrome) when you try to accede to belgianrail.be three request are sent:

- **GET belgianrail.be for the HTML file. The server responds with a 302 code status and provide the new address (see the next section for explication about this code).**
• **GET** www.belgianrail.be for the same HTML file but at the new location. The server responds again with a 302 status code and a new address.

• **GET** http://www.belgianrail.be/jp/sncb-horaires/query.exe/en and here the request is aborted so the site doesn’t work.

This is likely because at this date the new transport plan is implemented and the webmaster probably wanted that a user goes directly on the page for prepared his itinerary instead of home page. This probably saturate the server. In the afternoon this problem has been solved and the first page become the home page. Maybe it was a redirection problem.

III. **Analysis DNS Requests**

DNS is a service that allowing to translate a particular domain name to a corresponding IP address for retrieving the website. This facilitates the navigating because users has not to remember the changing IP address of all website. In this section we will analyse what’s happening from the DNS viewpoint.

The domain name belgianrail.be uses four name servers:

- ns.belnet.be
- dns.b-rail.be
- secdns.b-rail.be
- ns1.dns.interoute.net

They aren’t accessible over IPV6, only over IPV4. The domain names contacted have only A records for IPV4. They don’t use CNAMEs. CNAME (Canonical Name record) is used to specify that a domain name is an alias for another. The TTL for our domain names is 36155 seconds. The TTL is the lifetime of resource record in seconds. This is indicates how long a client or a resolver can store the resource record in its cache.

All DNS resolver contacted return the same A records. There is no load balancing to prevent an overload of server. All DNS resolvers provide the same answer. Thanks to RIPE Atlas measurement platform a lot of request have done outside Belgium and to each request the same answer is provided.

The domain name addthis.com uses four name servers:

- ns3.p27.dynect.net
- ns1.p27.dynect.net
- ns2.p27.dynect.net
- ns4.p27.dynect.net

They are accessible only over IPV4 and they don’t use CNAMEs. The TTL is 60797 seconds. The DNS resolvers contacted provide the same A records but the AAAA records may be different. This domain name supports CDN. We see that with the results provide by RIPE Atlas.

The domain name youtube.com uses also four name servers:

- ns2.google.com
- ns3.google.com
- ns1.google.com
- ns4.google.com

Because this domain names belongs to Google the same things that google-analytics.com applies here. Same for other domain like maps.gstatic.com, maps.googleapis.com and maps.google.com.

IV. **Analysis TCP**

TCP is a protocol that provides a reliable byte stream connection-oriented transport service on top of the unreliable connectionless network service provided by IP. This is used by a large number of application whose HTTP.

A. **Parallel TCP Connections**

The number of parallel TCP connections is six for the domain name belgianrail.be. This is correspond to the limit of Firefox for the number of parallel TCP connections. We don’t talk here about the other domain name because only few content are provided by them so only few connection TCP are established. Each TCP connection uses different port (even if the port was close before).

B. **Establishment TCP Connections**

We will analyse here establishment of TCP connections for belgianrail.be. This is used three-way handshake.

1) First the client send a packet with the **SYN** flag. His sequence number is 0 (relative number) and his window size value is 65535. This sent to the port 80 to the server (80 because is HTTP).

2) The server responds with a **SYN,ACK** flag to the client. With an akc number set to 1 (relative number) and a sequence numer set to 0. Its window size value is to 8760.

3) Finally the client responds with a **ACK** flag to the server. With a sequence number sets to 1 (relative number) and the ack number sets to 1 also (this confirms the establishment of TCP connection).

C. **TCP Options**

For the first connection to the server the client set some options:

- Maximum segment size: 1460 bytes. This is indicate the maximum segment size that the client is able to receive.
- Window scale: 5 (multiply by 32). It’s an option to increase the receive window size allowed in TCP.
- Timestamps: TSval is 723829757 and TSecr is 0 because no segment has already been received by the client.
• Sack permitted. Sack (selective acknowledgement) allow a TCP receiver to indicate that a blocks of data out of sequence has received correctly. The TCP connections with the web server terminate always with a FIN segment.

V. CONCLUSION

In conclusion, we analysed in this paper the domain name belgianrail.be from viewpoint the DNS, HTTP request and TCP connections. Thanks to this we have a better view of the functioning a website and what’s happen when you enter in your browser the url belgianrail.be. Different tools were very useful like Wireshark, Firebug (extension for Firefox) and Ripe Atlas measurement platform.

REFERENCES