What is traceroute?

Traceroute is a command used to know the path of data packets sent by a local network user as they travel all over the internet, until they reach a destination. Think about the internet as a big set of networks that are interconnected with the use of multiple routers that track the packet traffic between computers and servers all over the world. With this in mind, it’s pretty clear that a packet does not travel from a point to another directly, passing through different networks and, implicitly, different routers.

Traceroute command is more complex than the ping one – while *ping* only pings the final destination, traceroute not only pings the final destination, but it also pings each router on its way to the final destination; the computer sends three data packets to all routers they pass, and they send back these packets, allowing to obtain DNS domains and measure the round trip time that these took to and from each router. Note that traceroute is based on TTL mechanism: data packets are sent with a related value, and this value is decremented when they reach a router, reaching zero when they discover a new “checkpoint” (a new router).

With all of this said, traceroute can be a great tool to help discover network issues when it is not possible to establish a communication with a given IP address or domain.

Traceroute command executions

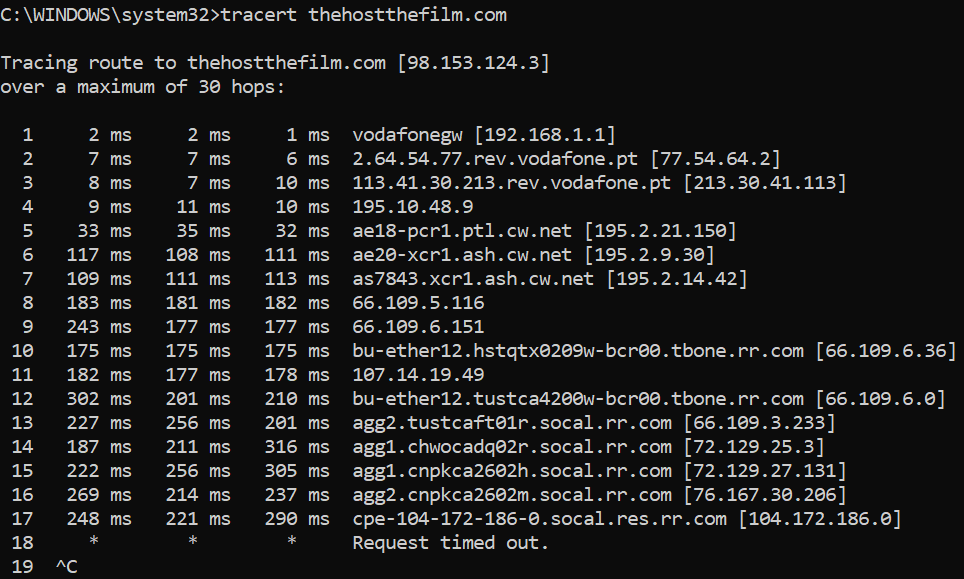


Fig.1 – Example of usage of the tracert command within Anthony’s network

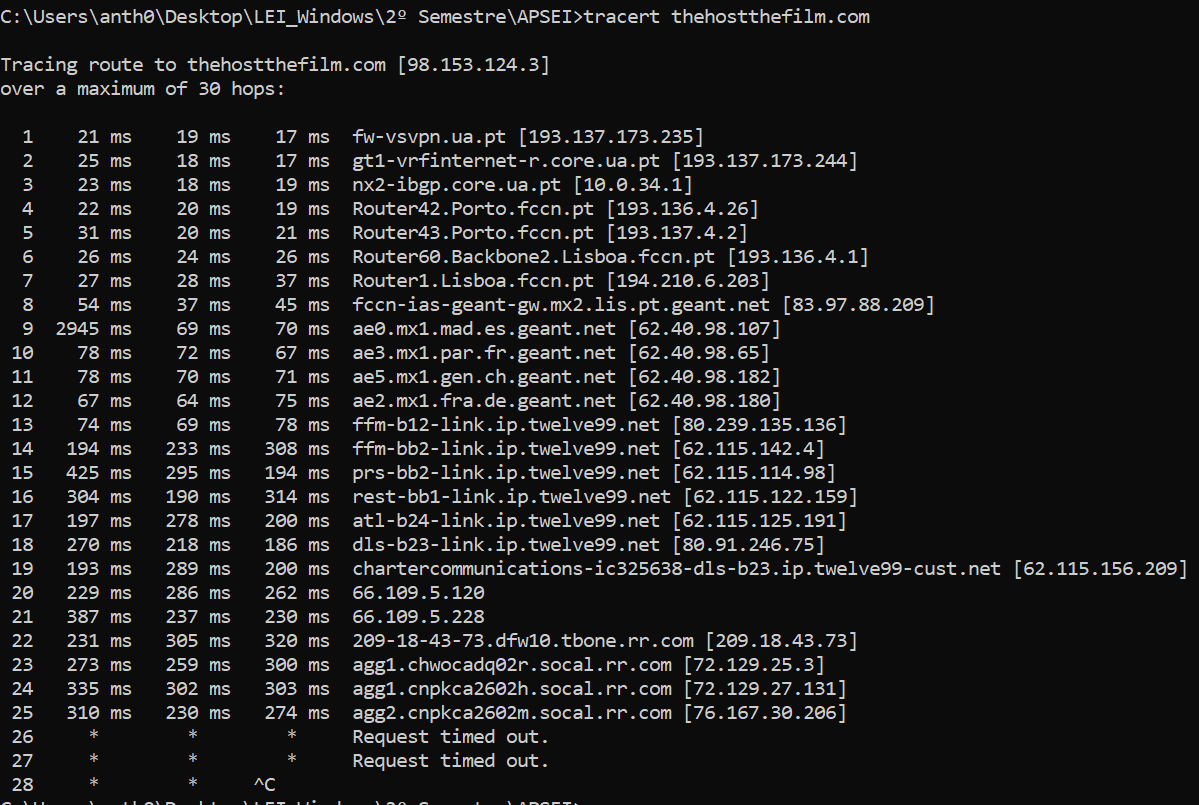


Fig. 2 – Example of usage of the tracert command within the UA network

Traceroute result interpretations

As shown by the screenshots above, the location of an user interferes with the traceroute results, for instance, by changing the initial IPs and domains. This happens because the user was in different networks, so the path that each data packet chose to reach its destination has changed.

Both inside and outside the network of the unniversity, it was never possible to complete the traceroute research - the last hops always have the “request timed out” message and there is no hop with the IP that’s used as parameter for the traceroute command. Most of the time, the last shown IP is 76.167.30.206 (check the screenshot above), but sometimes there is a new one – 104.172.186.0. Pay attention because we’ll come back to this later.

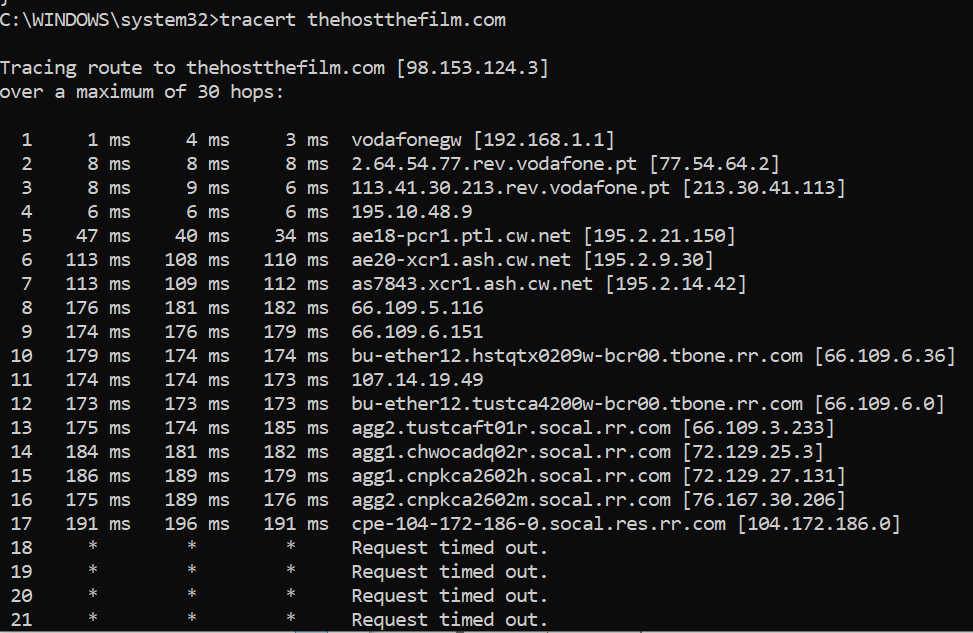


Fig.3 – Traceroute with a new last shown IP.

To help the process of interpeting the results obtained while testing the traceroute command, an IP geolocation API was used – ipinfo.io. Using the curl command together with ipconfig and the IP to be analyzed, it was possible to retrieve a json with additional information related to that IP:

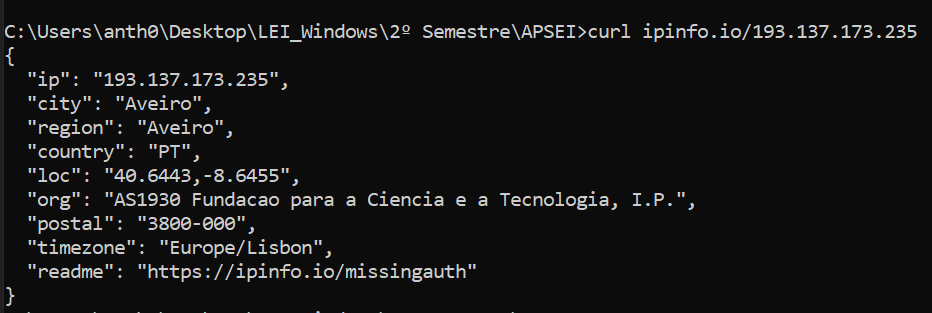


Fig.2 – Example of usage of the IP geolocation API ipinfo.io

With this in mind, a python program was made to help obtain all the jsons for each of the IPs obtained when executing the tracert command – this program accepts as an argument the name of a file where the output of a traceroute command is saved, and it filters the document to obtain all the IPs and perform the curl with each one, obtaining the multiple json outputs in a results file:

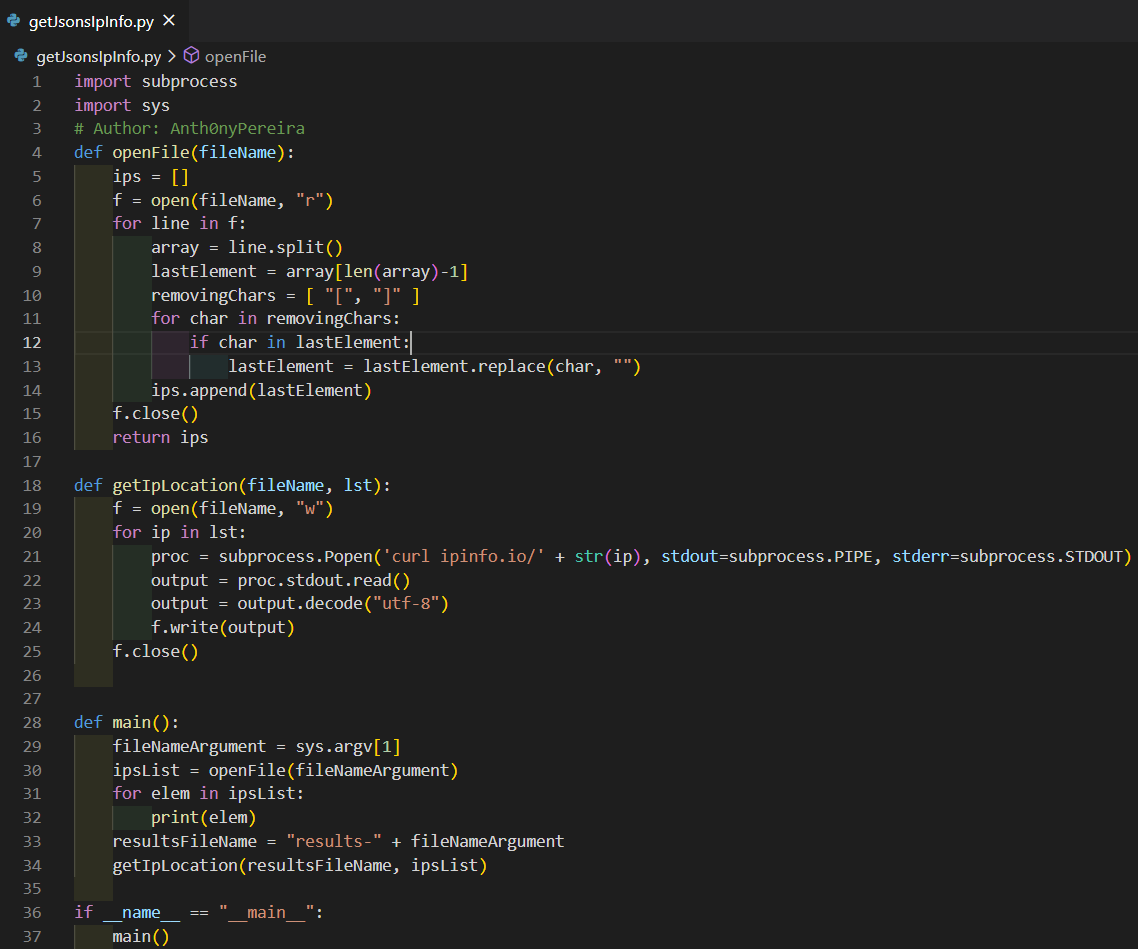
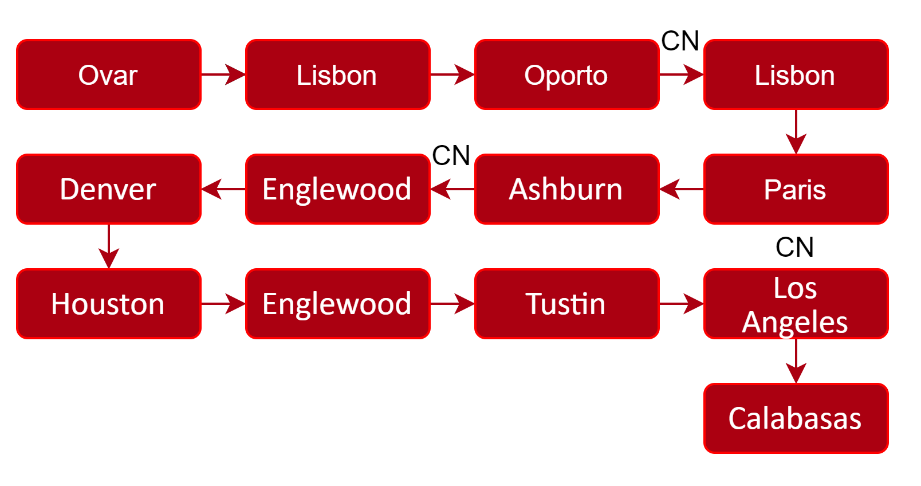
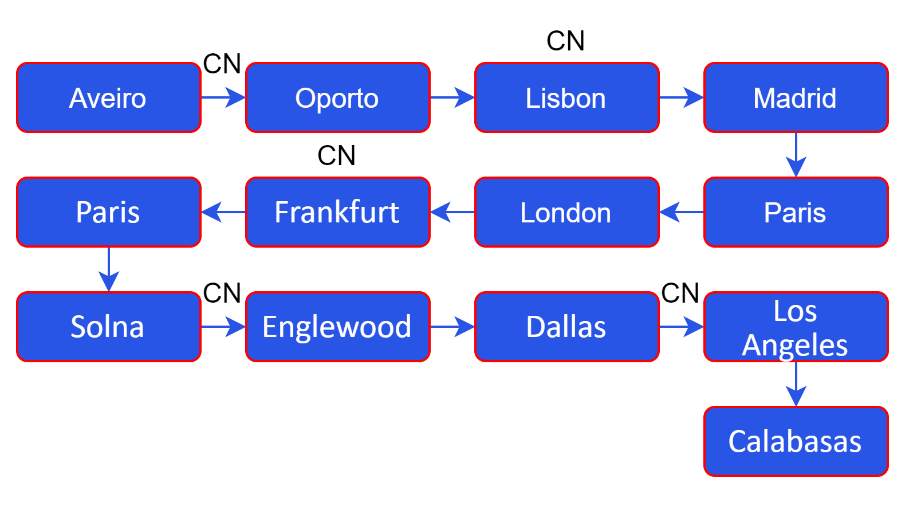


Fig.3 – Python program to obtain a sequence of JSONs using an IP geolocation API

With the results that were achieved, two diagrams were made for helping to visualize the way that these data packets perform during the traceroute command (CN means “changed network”):



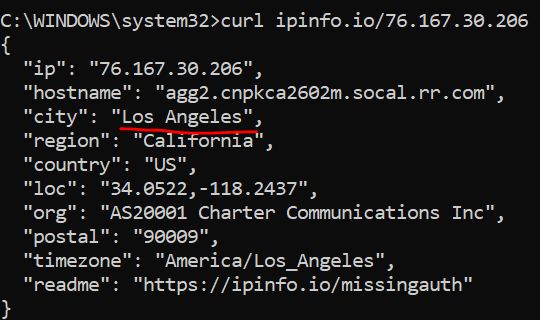
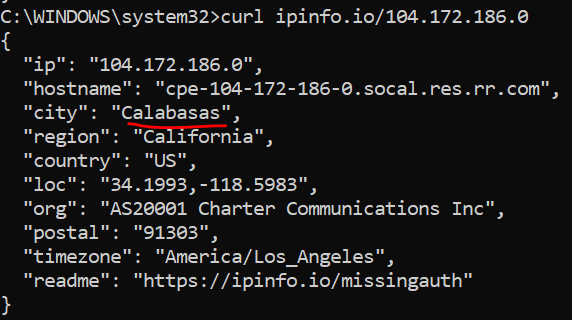
Diag.1 – Path traced with VPN switched off



Diag.2 – Path traced with VPN switched on

Some cities are out of place or have changed (check Tustin and Dallas, for example), so, it’s reliable to say (again) that the path changed in these two different situations. Checking the two tables (see the appendix), it can be observed that even the networks (organizations) have changed too, even after the UA network. So, we can assume that the tracert path is not always the same and that it’s dependent on the location where the user is, leading to a possible mid-different path/networks – not only the user’s network can (obviously) change, everything can change.

Now that the location of the IPs were recorded, let’s go back a little bit and check the location of both IPs 76.167.30.206 and 104.172.186.0:

Figs. 4 & 5 – Results of the consultation of the geolocation API for two specific IPs

The target IP 98.153.124.3 was also inspected:

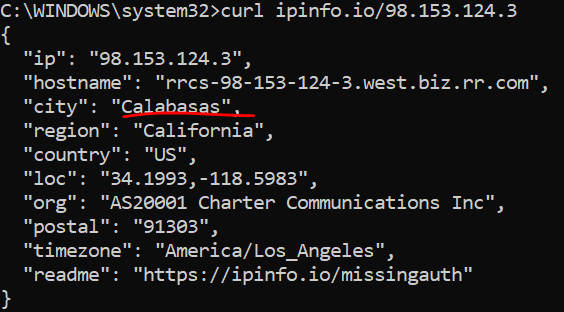


Fig.6 – Result of the consultation of the geolocation API for the target IP

All three IPs are from the same organization and region – California. 104.172.186.0 and 98.153.124.3 (the IP that sometimes appears in the trace and the initial IP, respectively) are from the same city – Calabasas!

Now, let’s check the distance between Los Angeles and Calabasas:

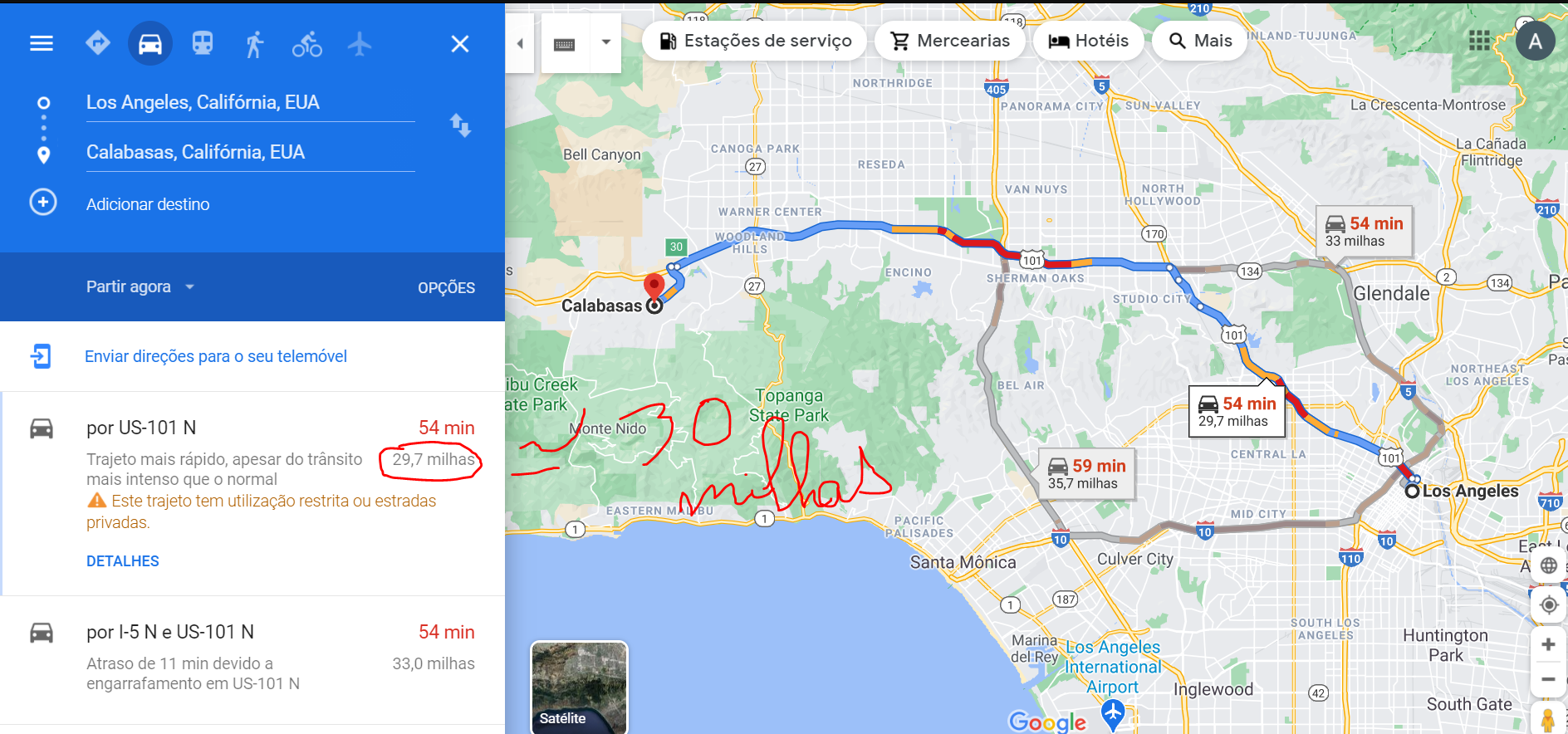


Fig. 7 – Distance between Los Angeles and Calabasas, using Google Maps

Time to make a conversion:

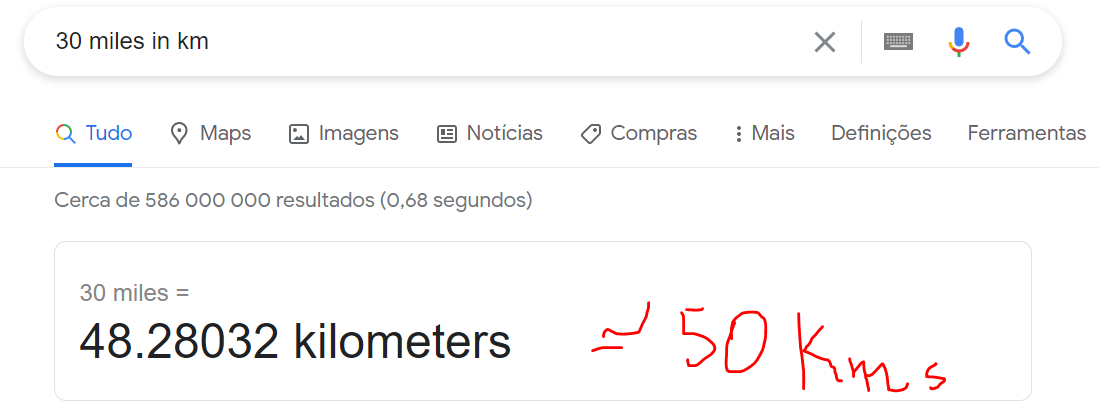


Fig.8 – Conversion between miles and kilometers, using Google

With all of this shown, it can be deduced that the server where the website that is being targeted by traceroute is located is probably denying any interaction with the user, that is, it is not responding with the traceroute packages that were sent to it. Both possible last IPs are close to the real location of the server - either they are from the same city or are 50 km apart from each other - so it’s safe to discard the possibility of 98.153.124.3 being too far away from 76.167.30.206 that they are not able to connect to each other. Another thing is that all three IPs are from the same network, so, we can surely say that the problem is not that it’s not possible to move from a previous network to the last one.

A quick search was performed and a curious result was gotten:

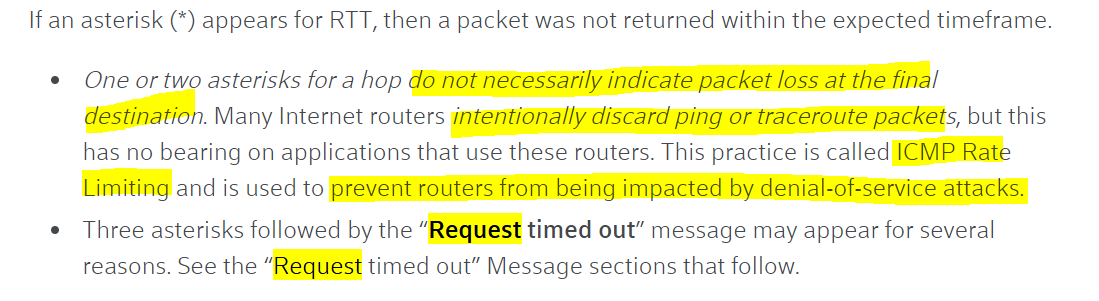


Fig.9 – Paragraph from the website Xfinity.com

It seems that some routers can “intentionally discard traceroute packets” because of security reasons, such as preventing DoS attacks. This could be what’s happening in this case.

Another reason that sustains this assumption is the fact that the website seems forgotten by everyone:

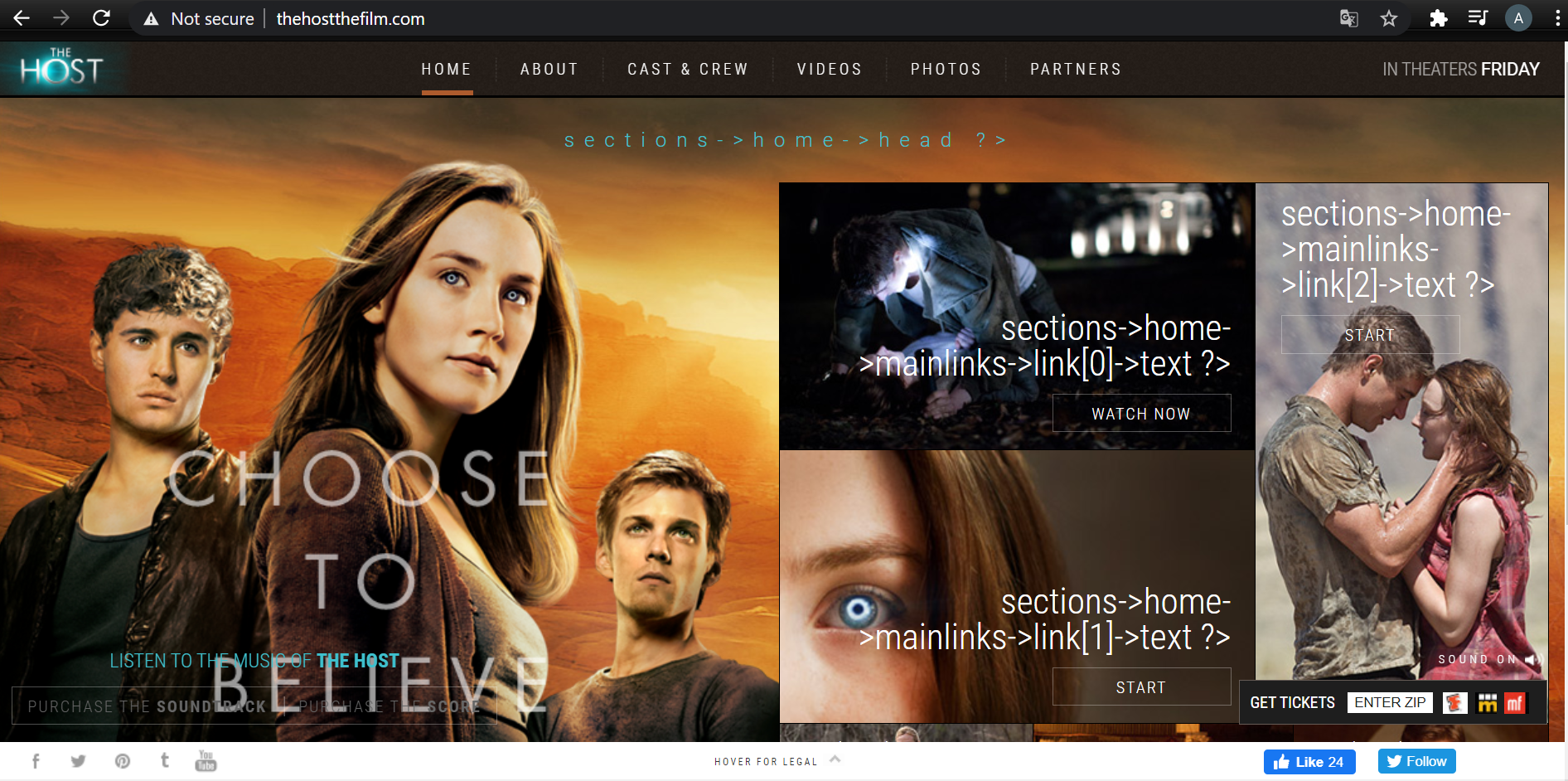


Fig.10 – Homepage of the website The Host, a movie from the year 2013

As we can clearly see from the screenshot taken, some things seem misconfigured (some raw code appears) and even if we check some features that it offers (or that it once offered), we can see that there are some bugs or some features that no longer exist. Thus, it’s possible to assume that this website is no longer maintained – which can lead to security problems. So, the owners could probably block some ways to communicate with the internet – to protect what’s left.

<https://www.netnod.se/ix/what-is-an-ixp-and-what-is-peering>

<https://www.xfinity.com/support/articles/run-traceroute-command>

Social and economic implications of a web connection

Social implications

- acesso à internet diferenciado

-

Table