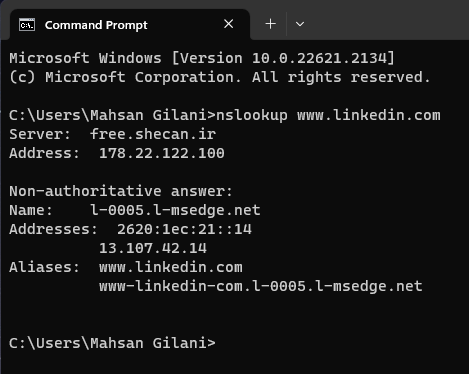
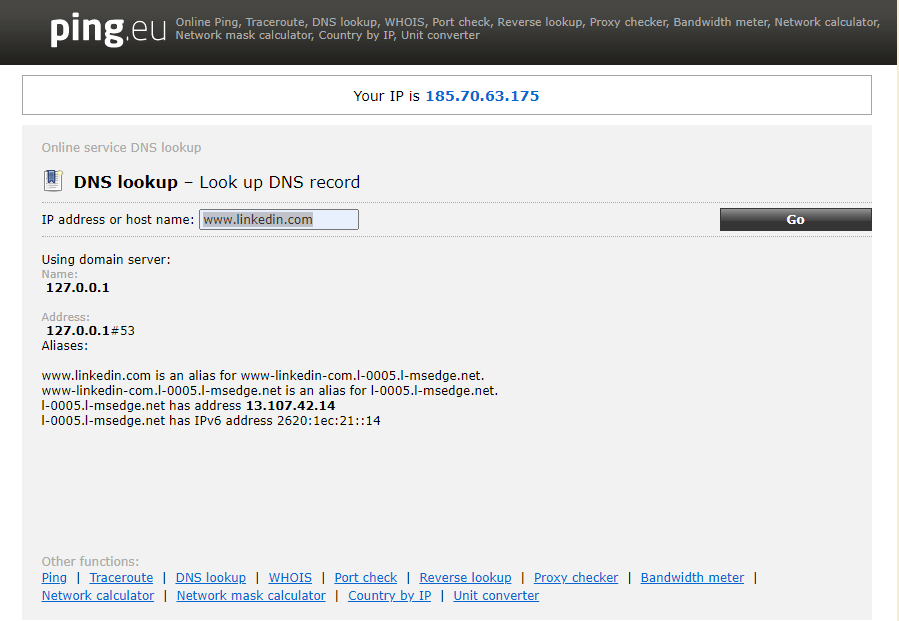
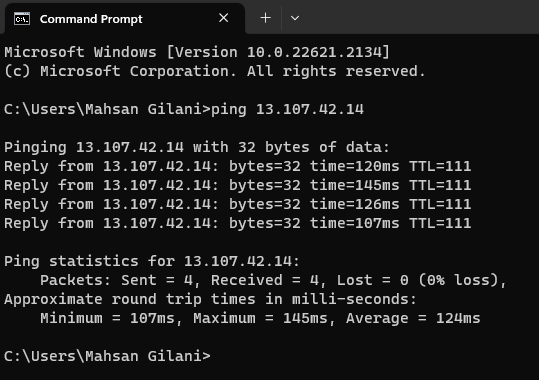
سوال 1)

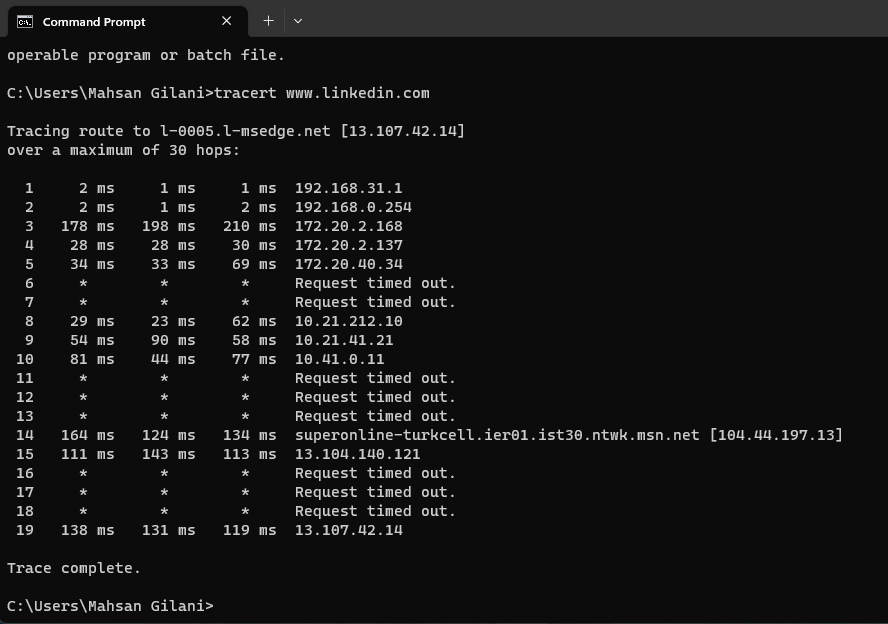




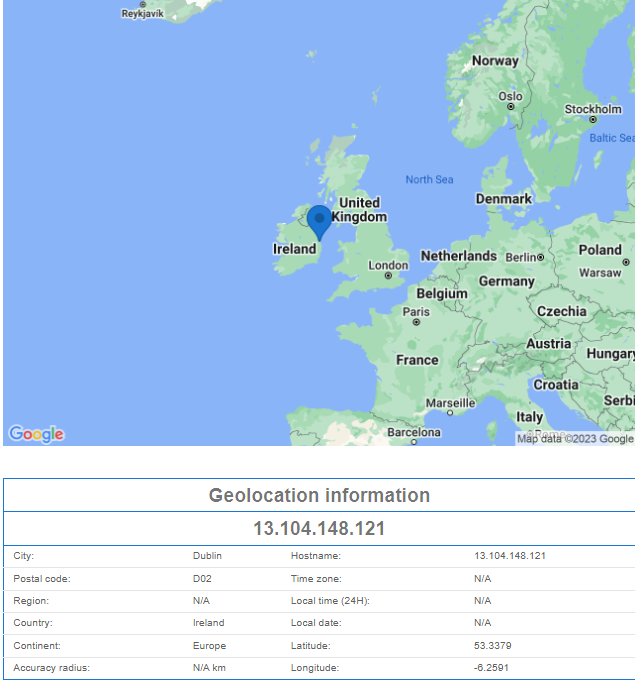
نتیجه دستور nslookup و سایت ping.eu یکسان بودند.

نتیجه‌ی پینگ:  


برای بدست آوردن مسیر دستور traceroute برای command prompt کار نمیکرد و برای ویندوز میبایست از دستور tracert میرفتم.

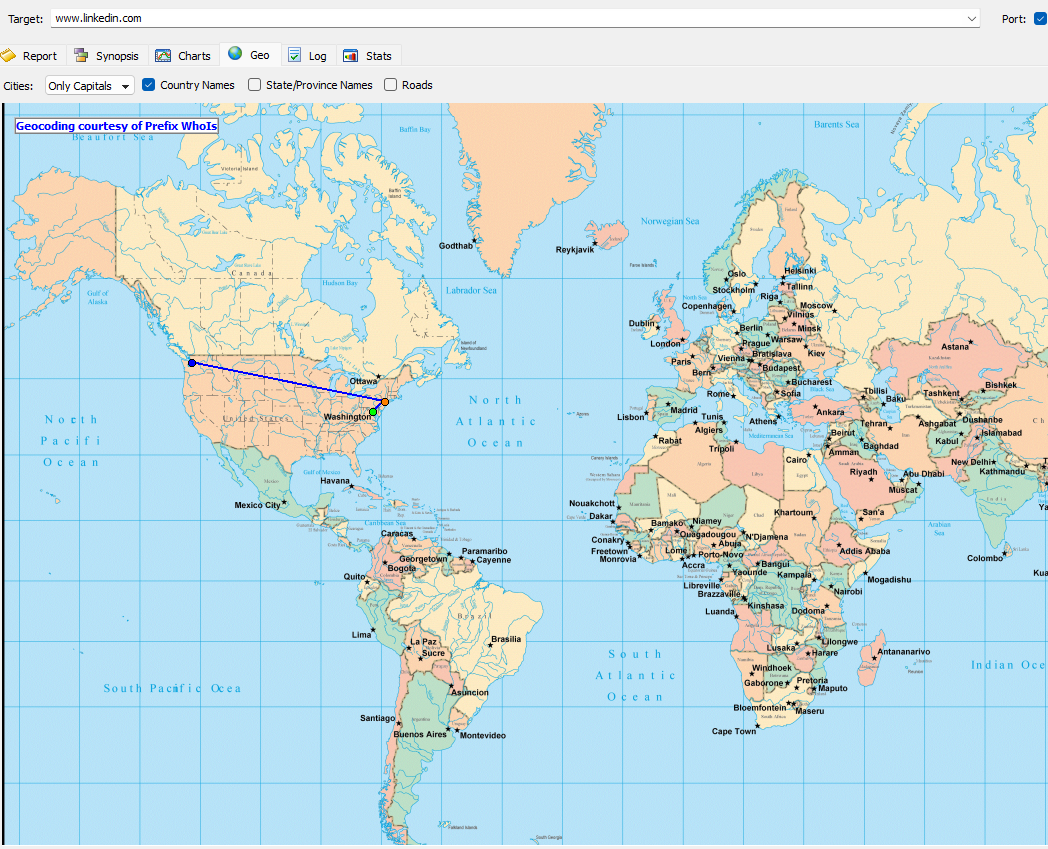


Ipهایی که با 192 و 10 شروع می‌شوند، همگی local و private هستند و مابقی همگی global هستند.

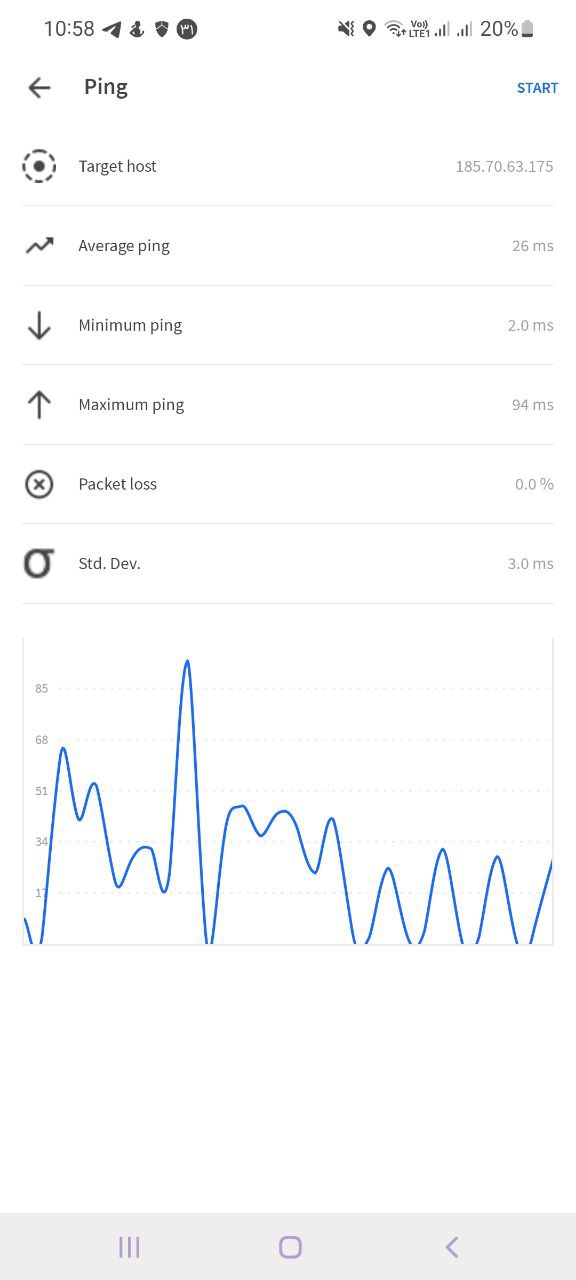
به عنوان مثال برای چک کردن محل جغرافیایی و اطلاعات مالکیت گره ی 13.104.148.121 از سایت <https://gsuite.tools/ip-location> استفاده می‌کنیم و نتیجه به صورت زیر است:  


مسیر ارتباطی با سرور لینکدین از طریق path analyzer pro :

بدون فیلترشکن بهم ارور میداد. در نتیجه با فیلترشکن نتیجه اش شد، و بخاطر همین مبدا رو از آمریکا نشون داده.



سوال 2)



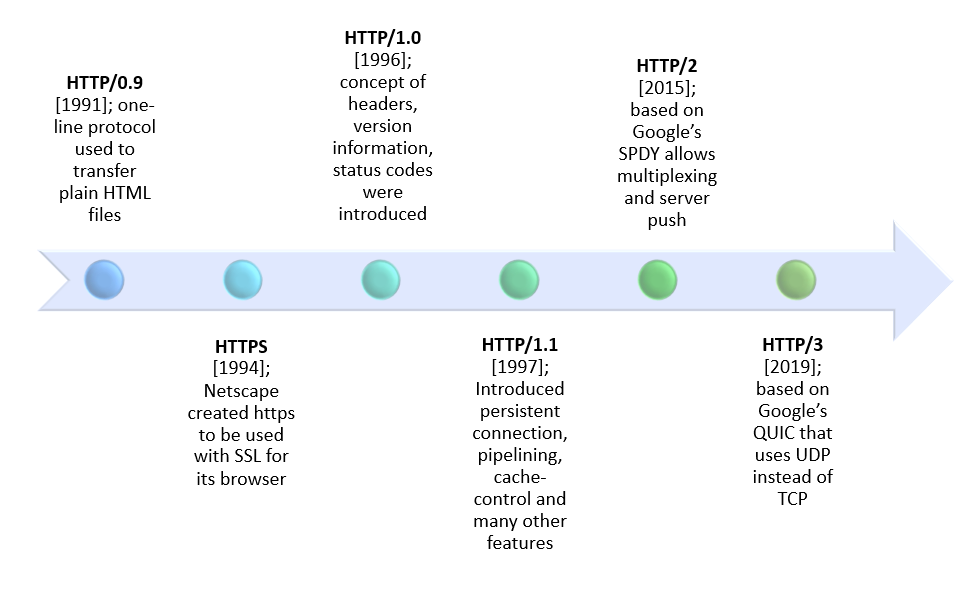
سوال 3)

پروتکل HTTP مجموعه قوانینی است که روی TCP/IP اجرا می‌شود و مشخص میکند فایل‌ها چگونه باید منتقل شوند.

تفاوت HTTP2 , HTTP1.1

HTTP2 امکان ارسال و دریافت چندگانه را دارد. یعنی میتواند روی یک اتصال TCP چندین request/response آن هم به شکل ناهمزمان انجام دهد. HTTP2 الگویی به نام server push را ارائه می‌کند. یعنی سرور پیش‌بینی می‌کند که client به چه منابعی نیاز خواهد داشت و مجموعه‌ای از آن‌ها را برای client می‌فرستد قبل از اینکه request از جانب client ارسال شود. البته client هم قابلیت این را دارد که server push را نادیده بگیرد. این کار در خیلی از مواقع باعث افزایش بهره وری می‌شود. اما HTTP1.1 فقط به درخواست هایی که ارسال شده پاسخ می‌دهد و چیز اضافه‌ای ارسال نمی‌کند. همچنین اینکه HTTP2 درخواست ها را به شکل باینری ارسال و دریافت می‌کند یعنی فقط 0و1 و این پیام‌ها در لایه باینری به قالب‌هایی که Data یا Header هستن تفکیک می‌شوند. این ویژگی کارایی را از نظر فشرده‌سازی و امنیت افزایش می‌دهد. HTTP2 برای فشرده‌سازی header از الگوریتم HPACK استفاده می‌کند (ترکیبی از کد Huffman یک دیکشنری از سمبل های از پیش تعریف شده و یک دیکشنری داینامیک از هدرهای واقعی که در هنگام اتصال داریم.) در حالی که HTTP1.1 فشرده‌سازی نمی‌کند.

HTTP3 در سال 2019 ارائه شده تفاوت اصلی با نسخه های قبلی در این است که از QUIC بجای TCP استفاده میکند. که سریع‌تر و ایمن تر هست. QUIC یک پروتکل برای لایه انتقال شبکه هست که ترکیبی از TCP و TLS رو ارائه میکند.



1xx: اطلاعات – به این معنی که درخواست، پذیرفته شده است یا فرایند همچنان ادامه دارد.

2xx: تایید می‌کند که عمل با موفقیت انجام شد و یا با موفقیت دریافت شد.

3xx: معمولاً این status code ها زمانی ارسال میشن که برای تکمیل کردن request نیاز به انجام دادن کارهای بیشتری توسط client هست. مثل ریدایرکت 301

4xx: خطای سرویس‌گیرنده (سمت کاربر) که نشان می‌دهد درخواست قابلیت تکمیل ندارد یا اینکه کاربر چیزی را نادرست وارد می‌کند. مثل 403 یا 404

5xx: خطای سرویس‌دهنده (سمت سرور) که حاکی از ناتوانی سرور در تکمیل یا انجام درخواستی است که ظاهراَ معتبر بوده است.

Content-type: محتوایی که ارسال میشود، چه نوعی است؟ مثلا عکس هست یا html یا ... / response

Host : میزبان و شماره پورت سرور که درخواست به آن ارسال شده / request

: Accept-Encodingانتظار میرود که چه نوع Encoding در پاسخ (responce) استفاده شود. /request

: Content-lengthسایز محتوایی که در جواب فرستاده می‌شود، چقدر است / response

Content-Range : میاد مشخص میکنه که یک پیام جزئی به کجا در یک پیام کامل تعلق دارد / response

Last-modified : تاریخ آخرین به روزرسانی محتوای مورد نظر / responce

Location : مشخص میکنه که به کدام صفحه تغییر مسیر میدیم / response

Cashe-Control : اطلاعات موجود در صفحات وب به چه روشی در مرورگر کاربران ذخیره شده و اصطلاحا **کش** شود. / request

Referer : یک فیلد اختیاری HTTP است که آدرس صفحه وب رو از منبع مشخص میکند که که درخواست از کجا شروع شده / request