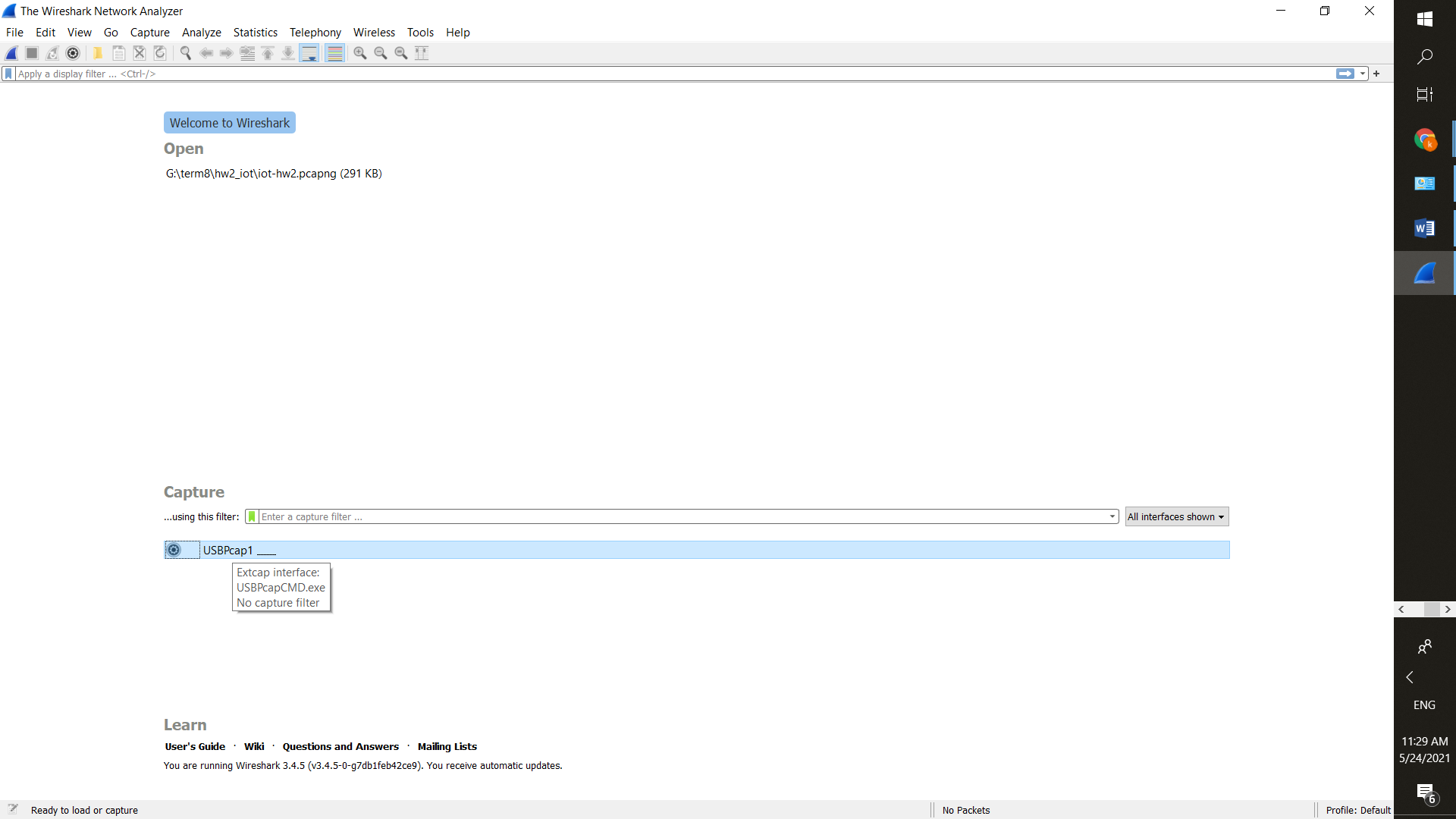
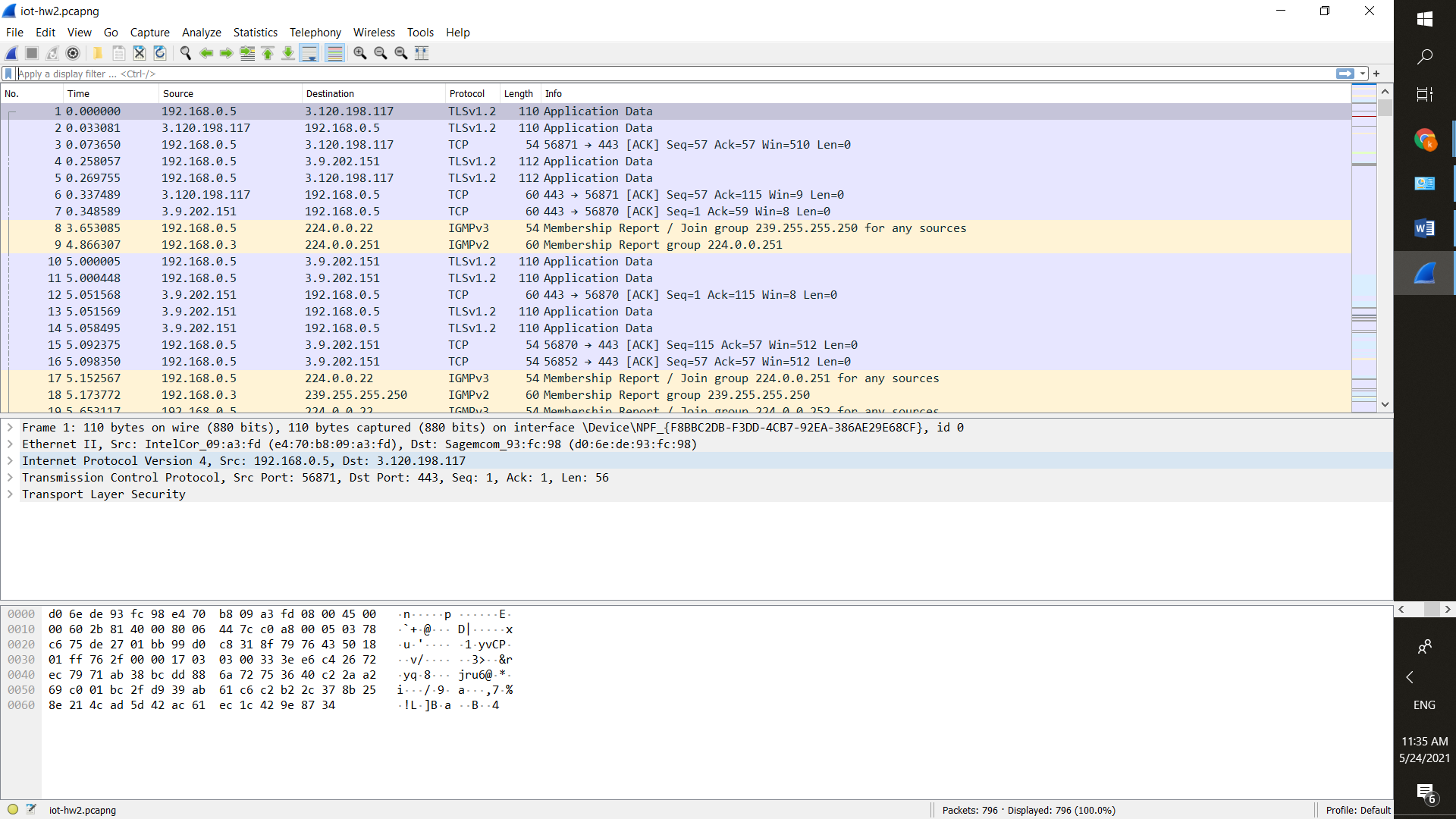
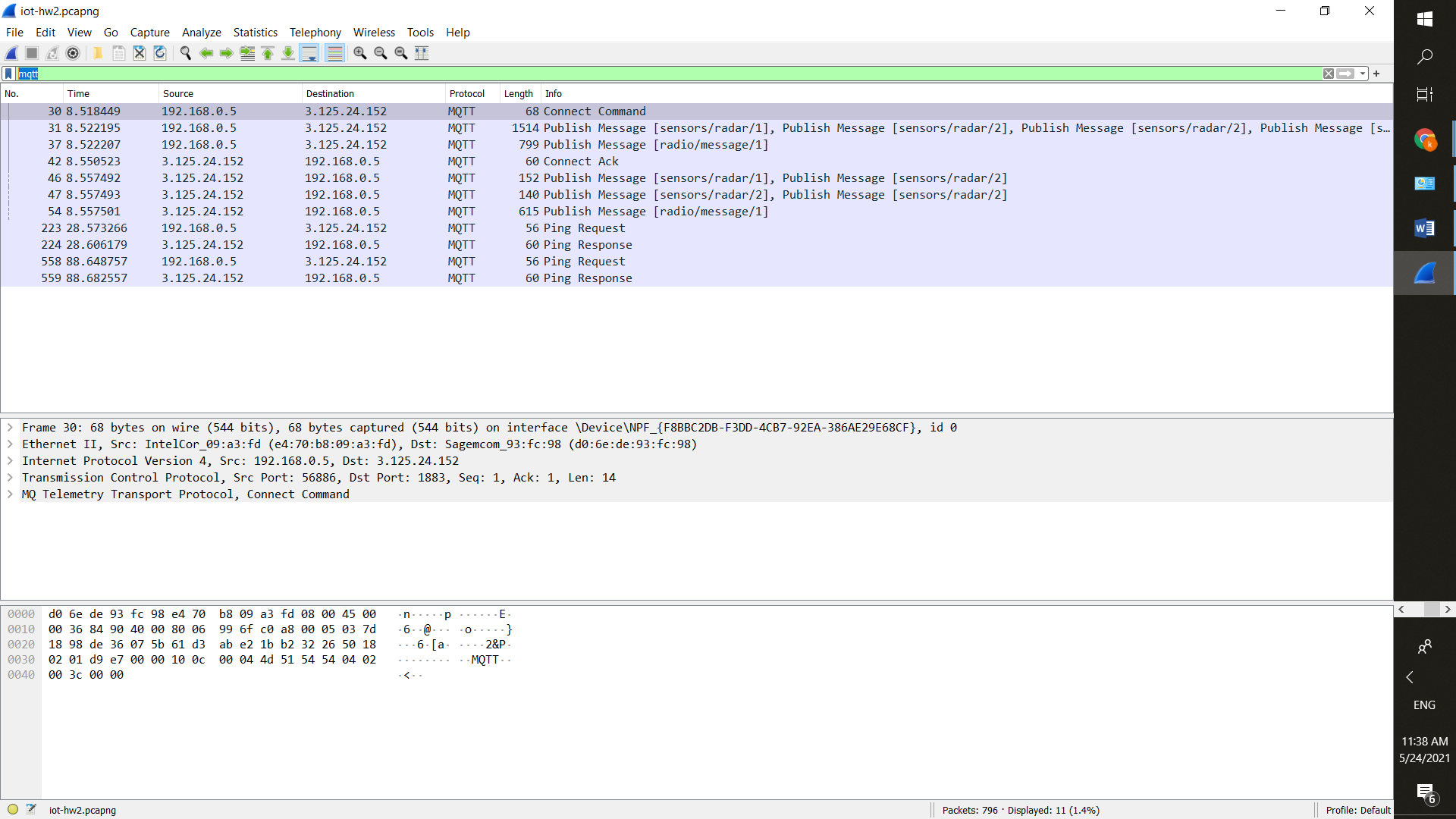
**بخش دوم: تحلیل ترافیک MQTT**

هدف کلی از این بخش آشنایی بیشتر شما با پکت های MQTT و تحلیل آن است. همراه صورت پروژه، فایل pcapng در اختیار شما قرار گرفته شده است که مربوط به ترافیک MQTT از قبل ضبط شده بین دو سیستم میباشد. شما باید به کمک Wireshark فایل pcapng را خوانده و پیام قابل خواندن منتشر شده را از پکت های MQTT استخراج کنید. برای تحویل این بخش گزارشی مختصر از نحوه استخراج پیام آماده کرده و به همراه پیام در پوشه گیت قرار دهید .

برای تحلیل ترافیک MQTT ابتدا نرم افزار wireshark را نصب کردیم و بعد فایل pcap موجود را ریکورد میکنیم 



حال فیلتر میکنیم



* حال info ها را تحلیل میکنیم

با توجه به info ها به طور کل میتوان گفت :

* پروتکل MQTT مبتنی بر TCP / IP است و هر دو client و broker باید یک پشته TCP / IP داشته باشند.
* publisher و subscriber هر دو به عنوان مشتری MQTT در نظر گرفته می شوند.
* ارتباط MQTT همیشه بین client و broker برقرار می شود ، هیچ client مستقیماً به client دیگری متصل نیست.
* به محض برقراری اتصال ، تا زمانی که client دستور قطع ارتباط را ارسال نکند یا اتصال را قطع کند ، broker آن را باز نگه خواهد داشت.
* MQTT از دستگاه های پشت دستگاه NAT (برای Ex-Router یا Firewall) پشتیبانی می کند.
* شناسه client (ClientId کوتاه) شناسه هر مشتری MQTT است که به یک کارگزار(broker) MQTT متصل می شود.
* Keep Alive یک بازه زمانی است ، client متعهد می شود به ارسال پیام های PING Request منظم به broker. این broker با PING Response پاسخ می دهد و این مکانیسم به هر دو طرف امکان می دهد تشخیص دهند که دیگری هنوز زنده است و قابل دسترسی است.
* پیام هایی که از نوع publish هستن در ان ها دیتا موجود هست و در ان دیتا جابه جا میشود اما بقیه پیام ها در جهت کنترل بسته هستند.
* اتصال از طریق client ارسال پیام "connect command" به broker آغاز می شود. پاسخ broker با "Connect Ack" و کد بازگشت است.

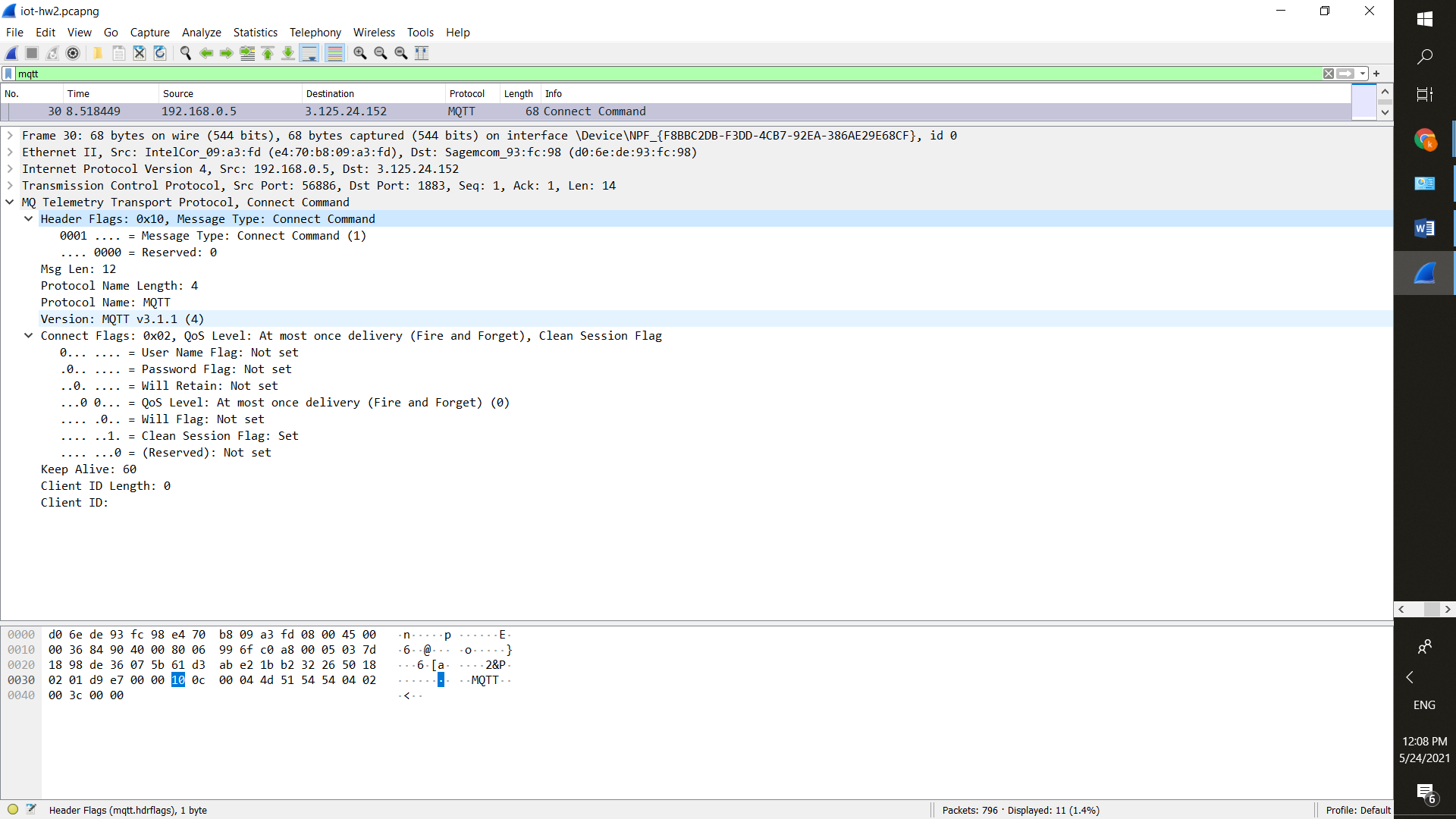
حال به بررسی هر کدوم از پکت ها طبق تعاریف بالا میپردازیم

* اغاز اتصال از طریق client به brocker و ارسال پیام "connect command"

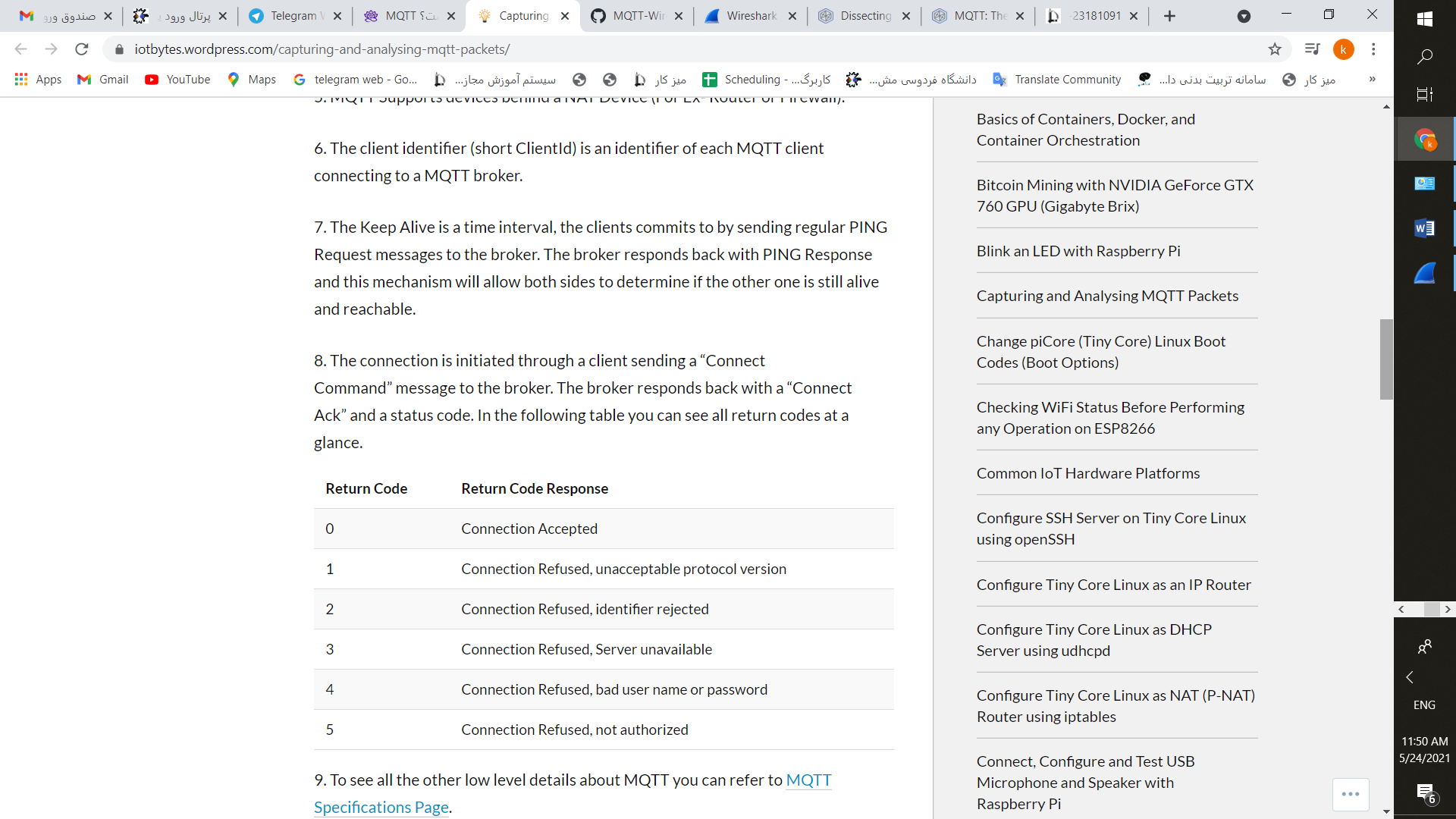
ارتباط mqtt بین مشتری و کارگزار است و هرگز مستقیماً با مشتری دیگری ارتباط برقرار نمی کند. شروع این اتصال با استفاده از یک دستور اتصال ارسال شده از مشتری به کارگزار است. اتصال ، پس از برقراری ارتباط ، تا زمانی که فرمان قطع ارتباط از مشتری را دریافت نمی کند ، باز نگه داشته می شود و قطع نمیشود .

Port مقصد 1883 است

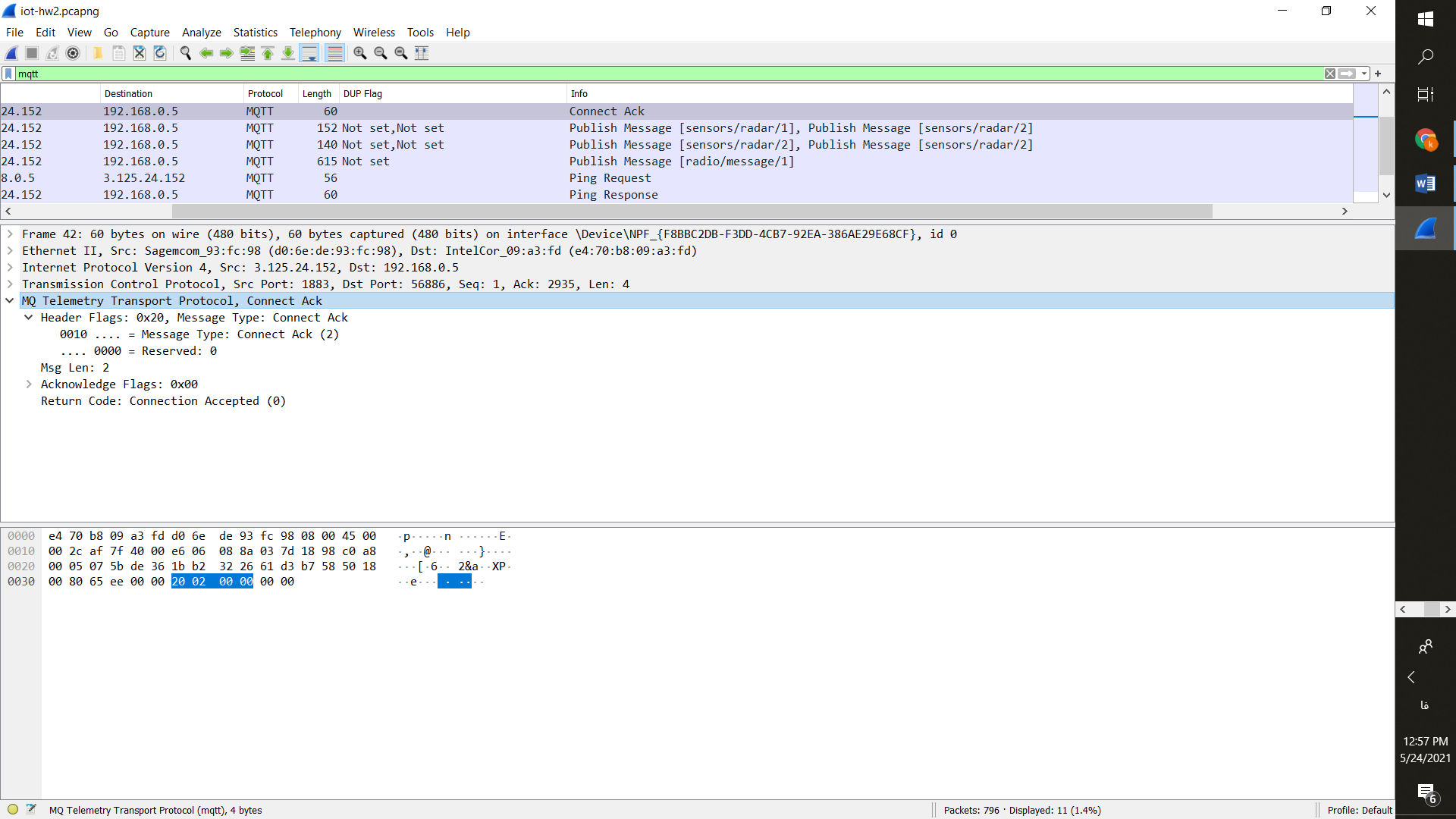
* header flags: اطلاعات مربوط به نوع بسته کنترل mqtt را در خود نگه می دارد.
* Connect flags: بایت پرچم اتصال شامل پارامترهایی است که رفتار اتصال mqtt را مشخص می کند. این وجود یا عدم وجود فیلدها در محموله را نشان می دهد.
* Clean session: بیت 1 پرچم های اتصال. این پرچم نشانگر کارگزار است که آیا با مشتری می خواهد یک اتصال مداوم برقرار کند یا خیر. پرچم ، هنگامی که روی "true" تنظیم می شود ، منجر به یک clean session می شود که در آن قطع اشتراک ها در هنگام قطع اتصال است و هنگامی که روی "false" تنظیم می شود ، می توان در صورت باقی ماندن اشتراک ها و ارسال پیام های با کیفیت بالا در هنگام اتصال مجدد ، یک اتصال بادوام برقرار شود
* will flag: بیت 2 از پرچم های اتصال. این پرچم ، هنگام تنظیم ، به این معنی است که ، در صورت پذیرفته شدن درخواست اتصال ، پیام باید در سرور ذخیره شود. یک پیام یک پیام mqtt با یک موضوع و یک پیام است. این در حالی است که به سایر مشتریان در مورد قطع ارتباط اطلاع داده می شود. با قطع ارتباط مشتری ، کارگزار این پیام را از طرف وی ارسال می کند. هنگامی که پرچم will به 1 تنظیم شود ،  will qos و will retain زمینه هایی را در پرچم های اتصال که توسط سرور استفاده می شود حفظ می کند.
* will qos: بیت 4 و 3 از پرچم های اتصال. سطح qos را نشان می دهد تا هنگام انتشار پیام will استفاده شود.
* Will retain: بیت 5 از پرچم های اتصال. اگرRetain روی 0 تنظیم شده است ، سرور باید پیام will را به عنوان پیام غیرقابل نگهداری منتشر کند و وقتی روی 1 تنظیم شود ، پیام will به عنوان یک پیام retained منتشر می شود.
* Username , password: بیت 7 و 6 پرچم اتصال به ترتیب. هنگامی که این زمینه ها تنظیم می شود ، انتظار می رود اعتبار در محموله بارگیری شود. mqtt اجازه می دهد تا نام کاربری و رمز عبور را برای احراز هویت مشتری و مجوز ارسال کنید. اگر رمز عبور در زیر رمزگذاری نشده باشد ، به متن ساده ارسال می شود.
* Keep alive: ازalive timer استفاده می شود تا بدانند مشتری mqtt در شبکه ای است که مشتری پیامهای درخواست پینگ منظم را به کارگزار ارسال می کند. کارگزار با پاسخ پینگ پاسخ می دهد.
* Client id: شناسه ای است برای هر مشتری mqtt که به یک کارگزار mqtt متصل می شود. این باید برای هر کارگزار منحصر به فرد باشد.
* payload: payload شامل شناسه مشتری است ، در قسمت عنوان ، پیام ، نام کاربر و قسمتهای رمز عبور که حضور آنها توسط پرچم ها تعیین می شود ، خواهد بود.



* Connect ack پاسخ broker هست که با کد بازگشتی همراه است. که کد بازگشتی 0 هست یعنی accept connection
* header flags: اطلاعات مربوط به نوع بسته کنترل mqtt را در خود نگه می دارد.
* session present: بیت 0 اتصال که ack byte پرچم جلسه فعلی است. این پرچم نشان می دهد که آیا کارگزار از تعاملات قبلی جلسه مداوم مشتری را دارد یا خیر.
* Return code: مقادیر و پاسخ های کد بازگشت



* payload: بسته اتصال هیچ محموله ای ندارد.



* ما در اینجا 5 تا مسیج داریم و قابل خواندن هستن چون از نوع publish هستند و پیام ان ها مشخص هست.

هنگامی که مشتری mqtt به کارگزار متصل شد ، می تواند پیام ها را منتشر کند.

* header flags: اطلاعات مربوط به نوع بسته کنترل mqtt را در خود نگه می دارد.
* dup flag: وقتی که dup flag صفر باشد ، به این معنی است که این اولین تلاش برای ارسال این بسته انتشار است. اگر dup flag 1 باشد ، نشانگر تلاش مجدد برای ارسال پیام است.
* qos: سطح qos سطح اطمینان پیام را تعیین می کند.
* retain flag: اگر retain flag روی 1 تنظیم شده باشد ، سرور باید پیام و qos آن را ذخیره کند ، تا بتواند اشتراک های آینده را با موضوع مطابقت دهد. هنگامی که یک بسته انتشار به یک مشتری مشترک ارسال می شود ، اگر بسته به دلیل اشتراک جدید ارسال شود ، سرور باید پرچم نگهدارنده را روی 1 تنظیم کند. هنگام ارسال بسته ، سرور باید پرچم نگهدارنده را روی 0 تنظیم کند زیرا صرف نظر از اینکه پرچم هنگام دریافت پیام تنظیم شده باشد ، با یک اشتراک ثابت مطابقت دارد.
* Topic name: رشته utf-8 که در صورت نیاز به ساختار سلسله مراتبی ، می تواند حاوی اسلش های رو به جلو نیز باشد. پیام انتشار باید حاوی موضوعی باشد که توسط کارگزار برای فیلتر کردن موضوع استفاده می شود. به این ترتیب کارگزار برای مشتریانی که در این موضوع مشترک شده اند پیام می فرستد.
* message: پیام payload همراه با موضوع است که شامل داده های واقعی برای انتقال است. از آنجا که mqtt داده های تجربی است ، می توان محموله را بر اساس مورد استفاده ساخت.
* payload: محموله شامل پیامی است که منتشر می شود.
* Decode کردن پیام های payload پکت ها :

1. روی چنل sensors و radar مقدار یک را پابلیش کن

Message: 206465746563746564206f626a6563742033206d696c6c696f6e206b6d

detected object 3 million km

1. روی sensor و radar مقدار دو را پابلیش کن

Message:

7261646172206465746374696f6e2c20636865636b696e672075706c696e6b

radar detction, checking uplink

1. روی sensor و radar مقدار دو را پابلیش کن

Message:

6465746563746564206461746173747265616d

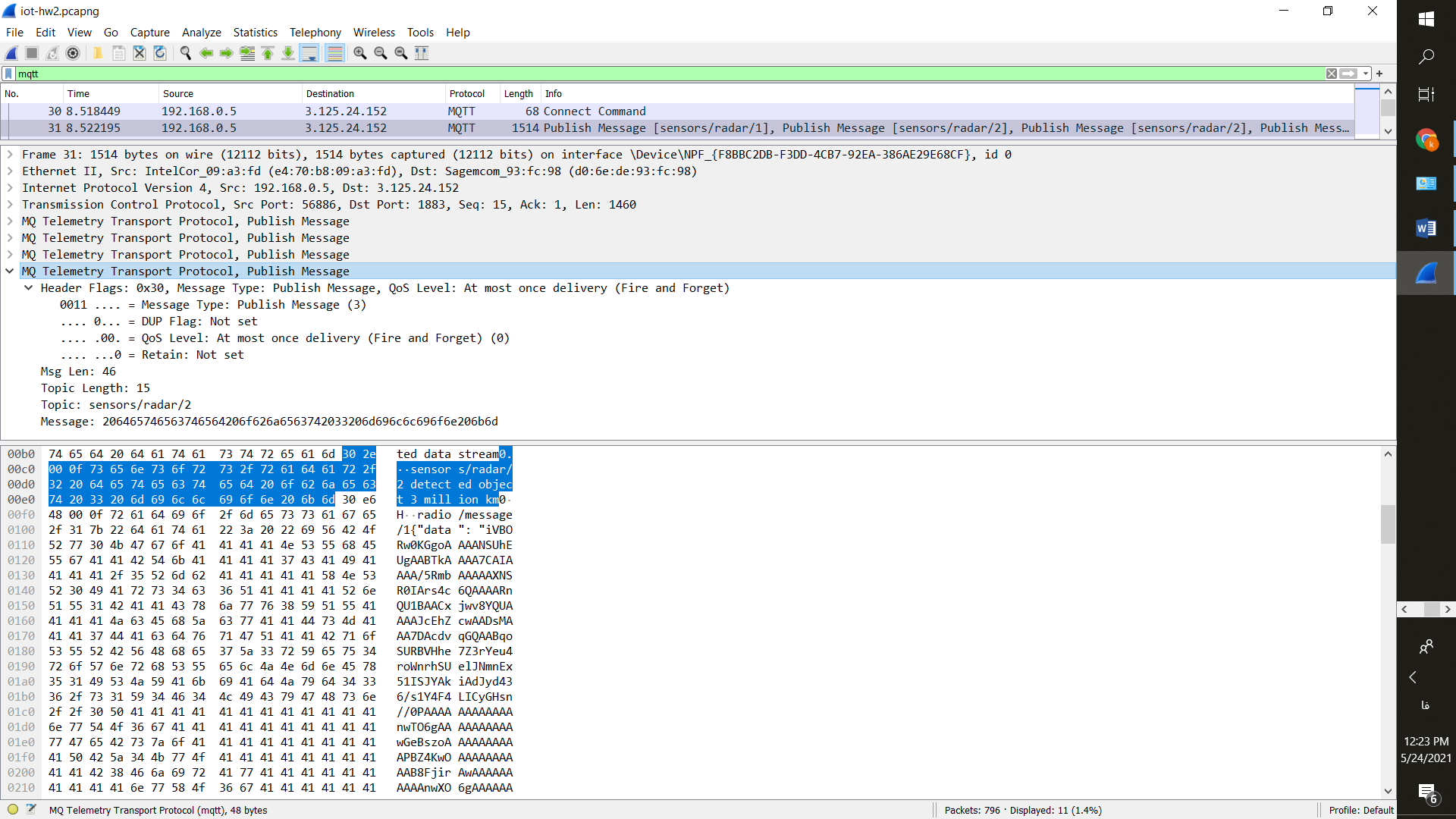
detected datastream

1. روی sensor و radar مقدار دو را پابلیش کن

Message:

206465746563746564206f626a6563742033206d696c6c696f6e206b6d

detected object 3 million km

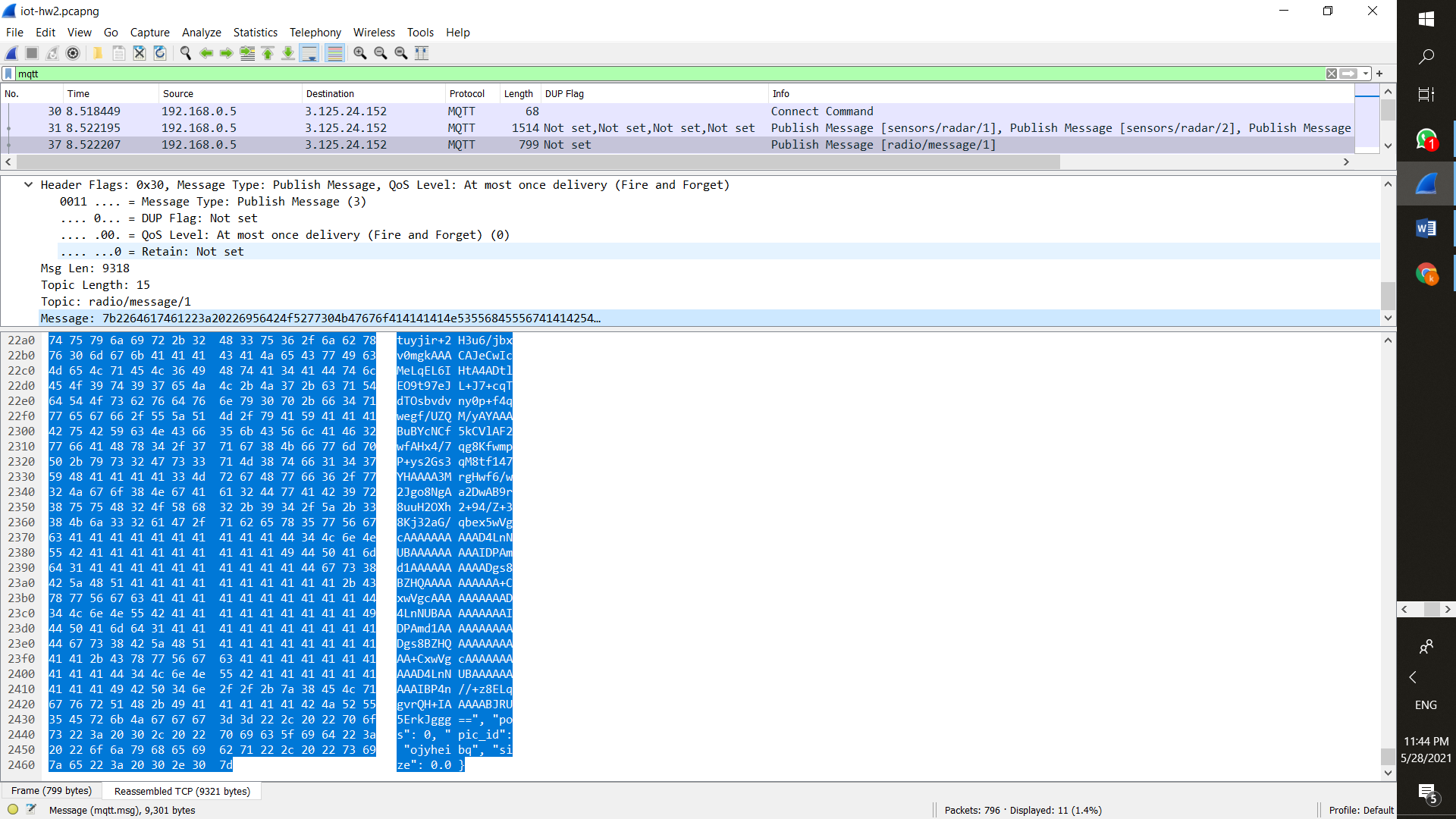


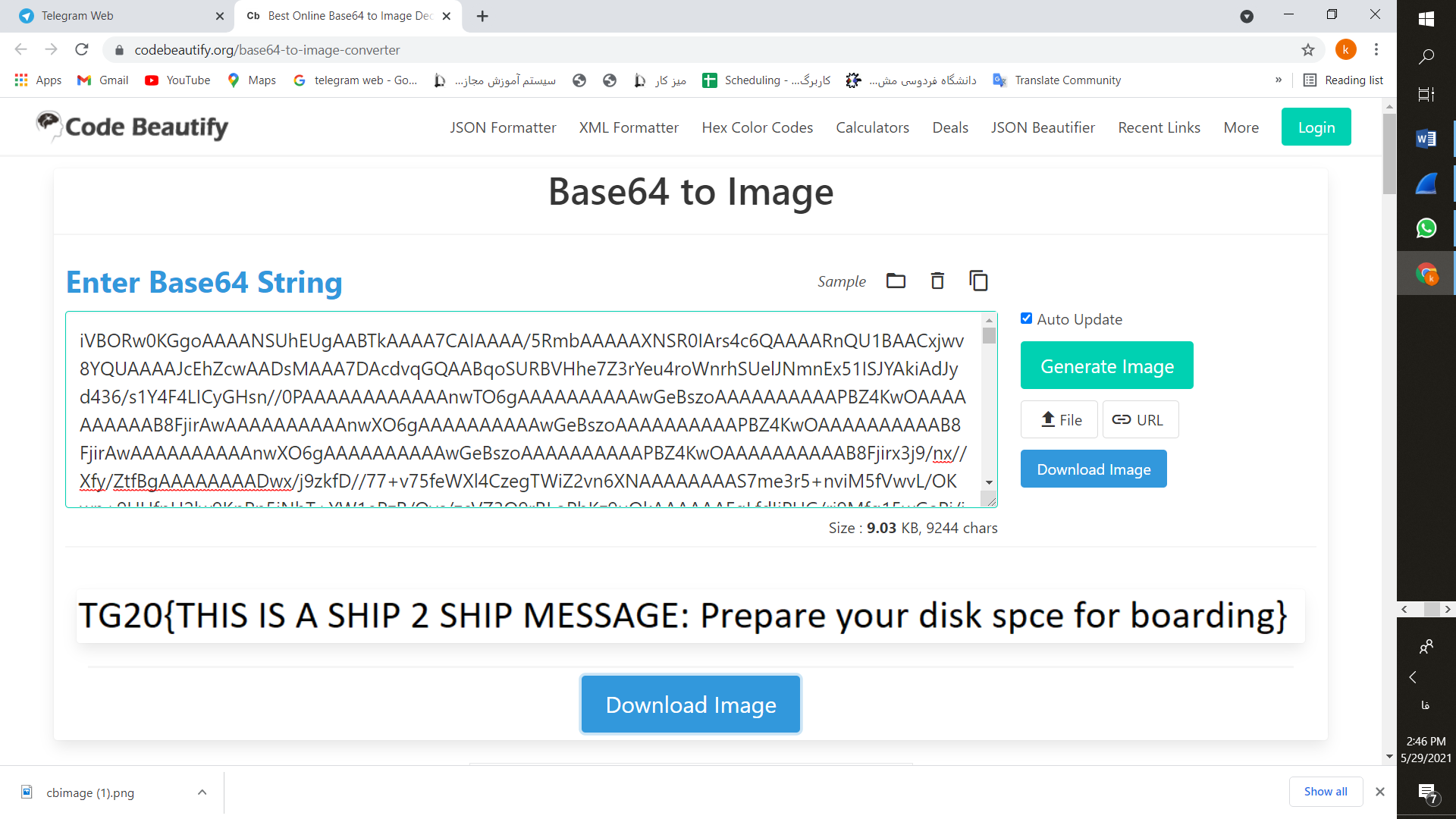
1. روی چنل sensors و message مقدار یک را پابلیش کن

یک فایل json بر میگرداند

* + - ابتدا کلید های پکت رو پیدا میکنیم همین طور که در تصویر پایین میبینید شامل pic\_id , data , pos , c\_id ,size ("pos": 0, "pic\_id": "ojyheibq", "size": 0.0)هست و از کلید های ان میتوان حدس زد قالب تصویر است پس فایل جیسون رمزگذاری شده و encode شده را decode میکنیم توسط سایت زیر

<https://codebeautify.org/base64-to-image-converter>





و تصویری که به ما میدهد این متن هست

C:\Users\lenovo\Downloads\cbimage.png

1. روی چنل sensors و radar مقدار یک را پابلیش کن

Message:

206465746563746564206f626a6563742033206d696c6c696f6e206b6d

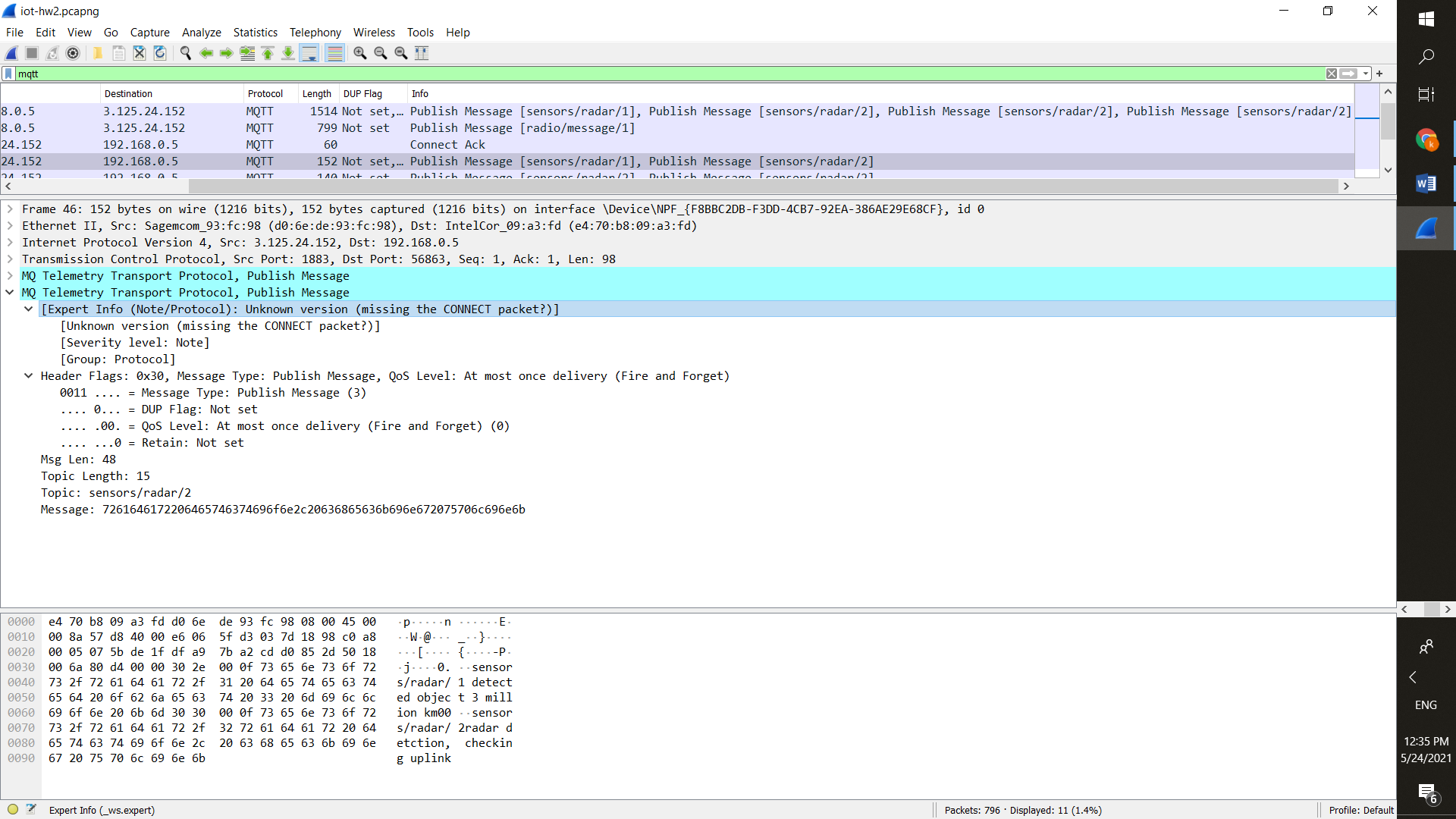
detected object 3 million km

1. روی چنل sensors و radar مقدار دو را پابلیش کن

Message:

7261646172206465746374696f6e2c20636865636b696e672075706c696e6b

radar detction, checking uplink



1. روی چنل sensors و radar مقدار دو را پابلیش کن

Message:

6465746563746564206461746173747265616d

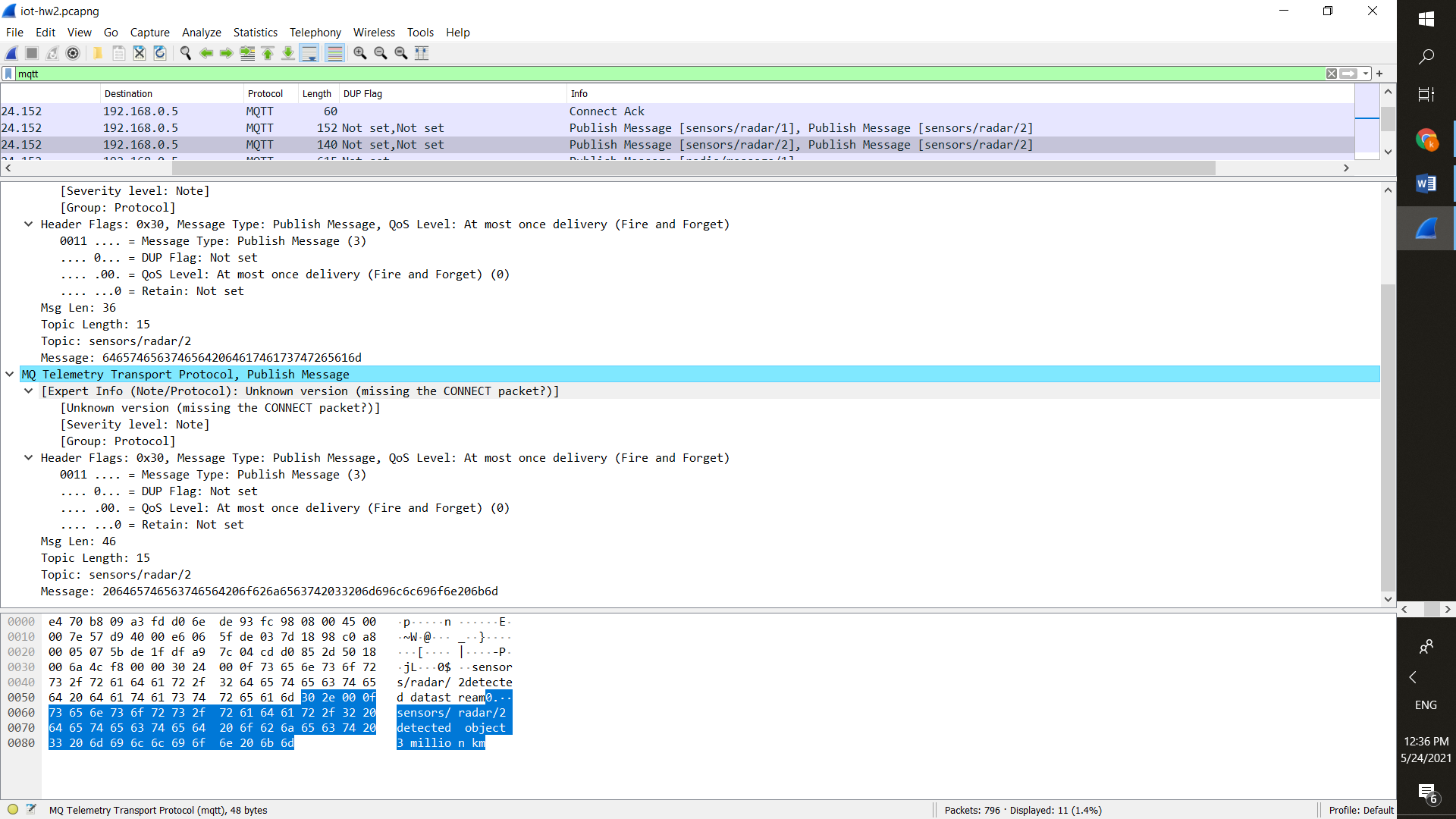
detected datastream

1. روی چنل sensors و radar مقدار دو را پابلیش کن

Message:

206465746563746564206f626a6563742033206d696c6c696f6e206b6d

detected object 3 million km

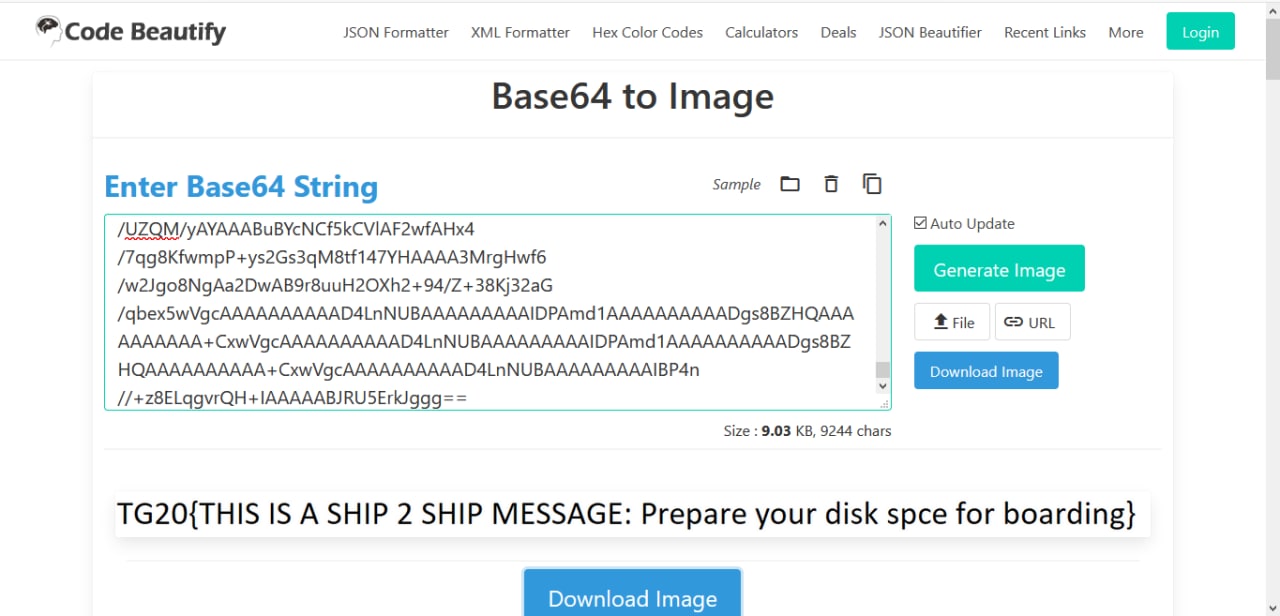


1. روی چنل sensors و radar مقدار یک را پابلیش کن

یک فایل json برمیگردونه.

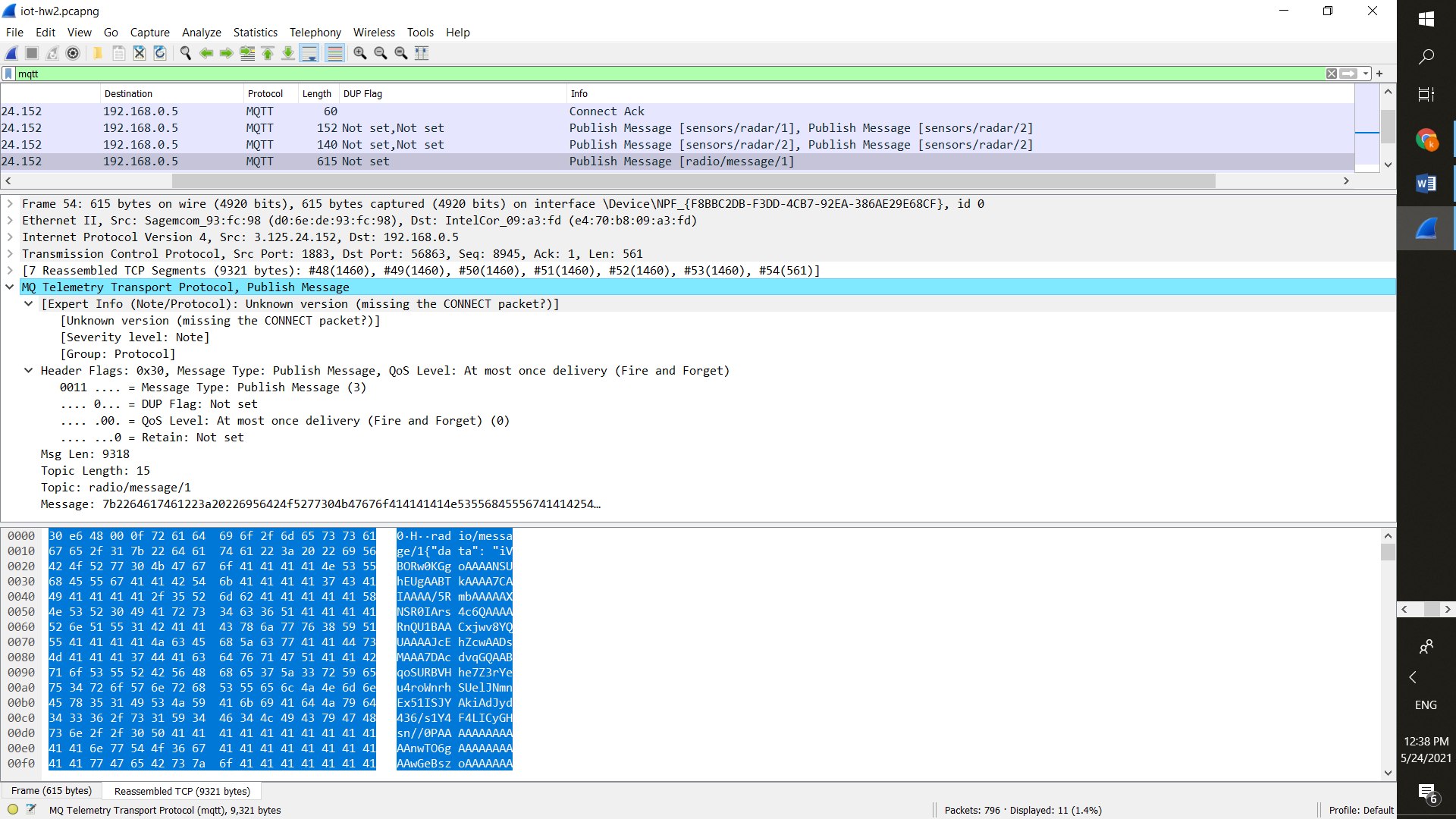
* + - ابتدا کلید های پکت رو پیدا میکنیم همین طور که در تصویر پایین میبینید شامل pic\_id , data , pos , c\_id ,size ("pos": 0, "pic\_id": "ojyheibq", "size": 0.0)هست و از کلید های ان میتوان حدس زد قالب تصویر است پس فایل جیسون رمزگذاری شده و encode شده را decode میکنیم توسط سایت زیر

<https://codebeautify.org/base64-to-image-converter>



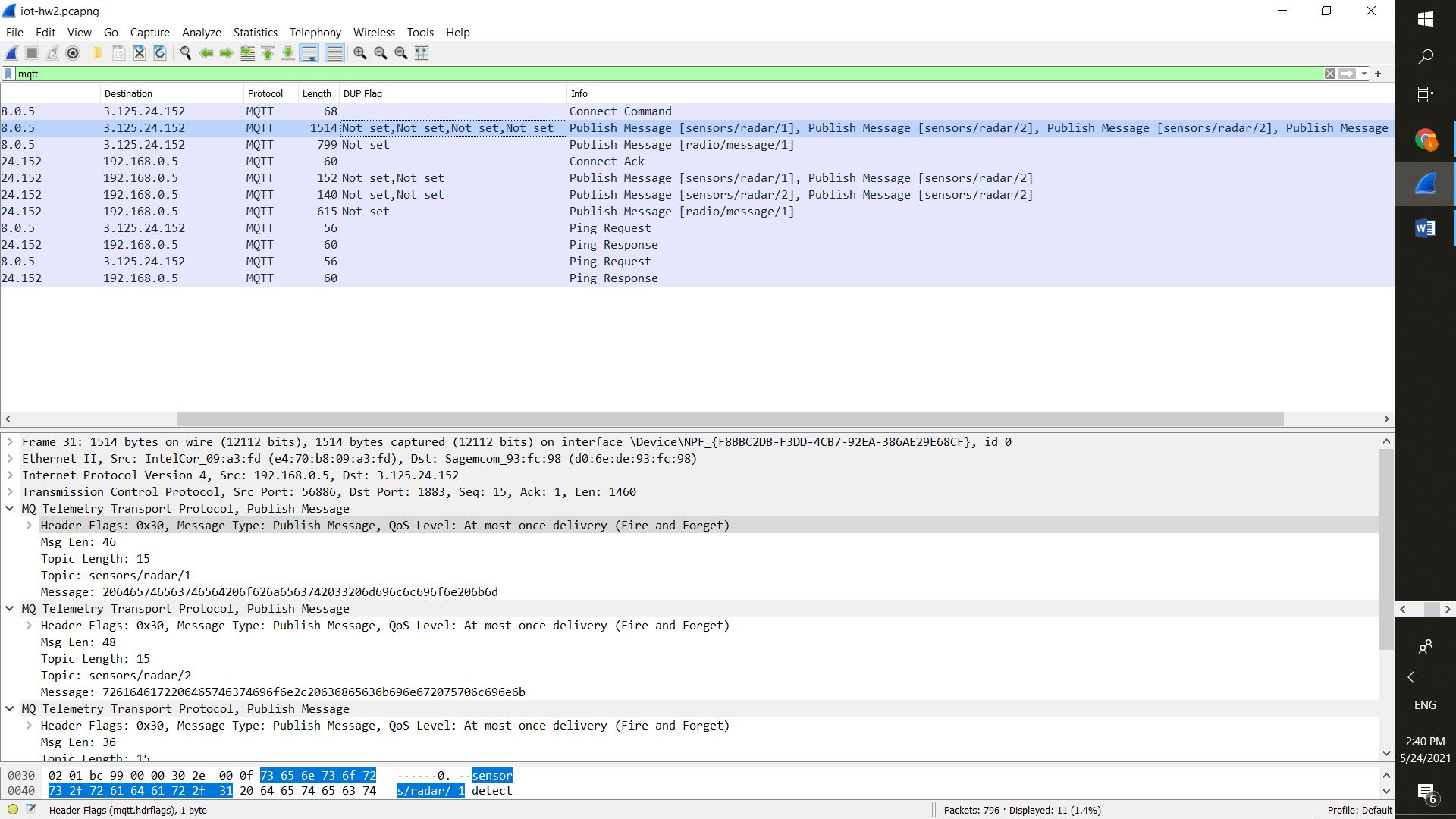
و تصویری که به ما میدهد این متن هست

C:\Users\lenovo\Downloads\cbimage.png



* قطع اتصال پیام بسته کنترل نهایی است که توسط مشتری به کارگزار ارسال می شود. این به معنای قطع ارتباط پاک توسط مشتری است.
* header flags: اطلاعات مربوط به نوع بسته کنترل mqtt را در خود نگه می دارد.
* payload: بسته disconnect فاقد محموله است.
* Mqtt keep alive

Keep alive با پیام های pingreq و pingresp حفظ می شود.

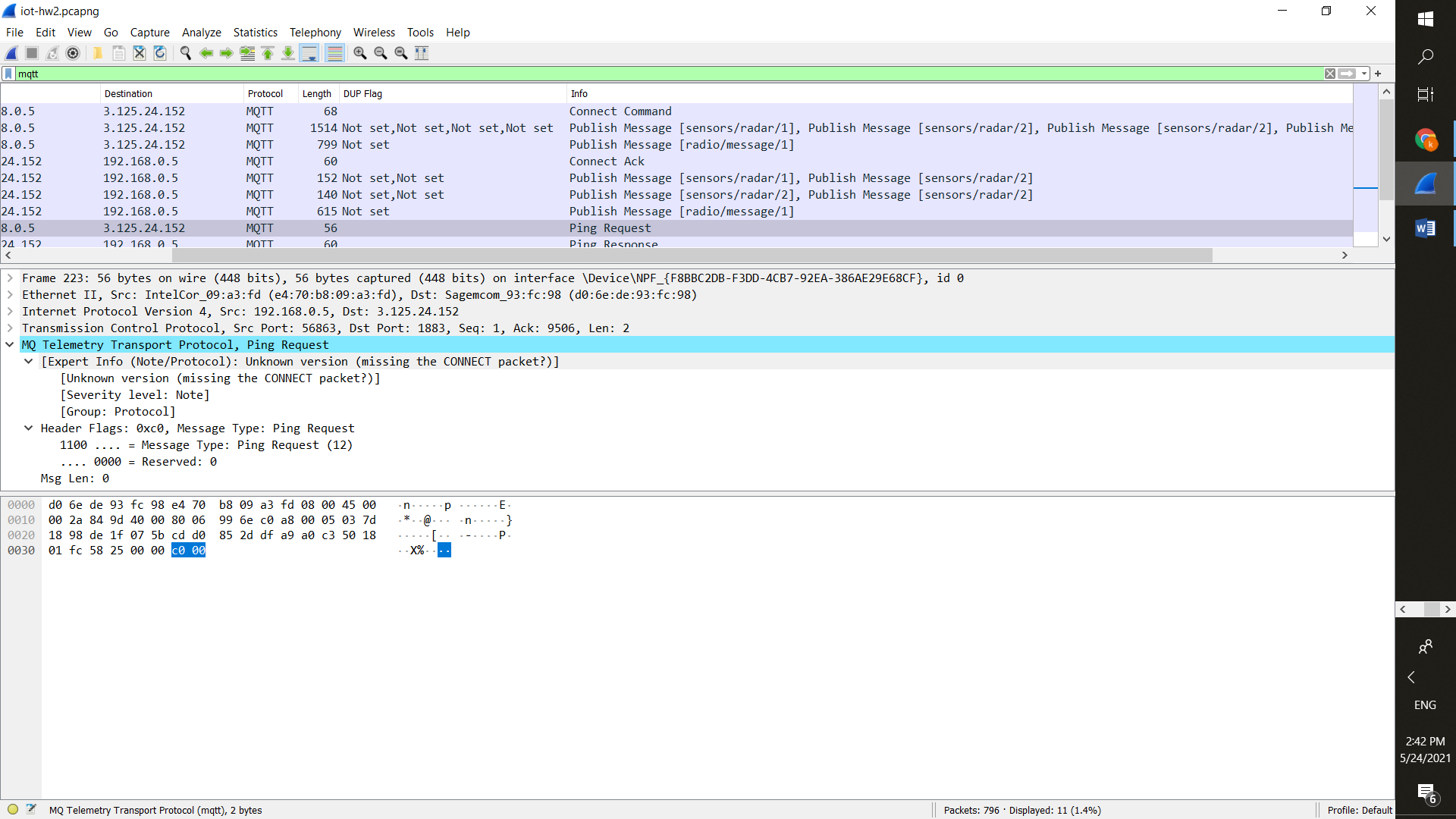


* پیام ping req که مشتری در یک بازه زمانی این پیام را به broker ارسال میکند . که ببیند هنوز کارگزار یوجود دارد یا نه .

specification mqtt می گوید:

"این مسئولیت مشتری است که اطمینان حاصل کند که فاصله بین بسته های کنترل ارسال شده از حد زنده نگه ندارد. در صورت عدم ارسال بسته های کنترل دیگر ، مشتری باید یک بسته pingreq ارسال کند.

* header flags: اطلاعات مربوط به نوع بسته کنترل mqtt را در خود نگه می دارد.
* payload: بسته pingreq فاقد محموله است.



* پیام ping res که کارگزار در یک بازه زمانی این پیام را به مشتری ارسال میکند. این نشان دهنده در دسترس بودن کارگزار برای مشتری است.
  + header flags: اطلاعات مربوط به نوع بسته کنترل mqtt را در خود نگه می دارد.
  + payload: بسته pingresp فاقد محموله است.

