Introduction

MAPI/ HTTP is slowly replacing RPC/ HTTP. Office 365 has already implemented it and Outlook 2016 use this protocol by default.

The questions we explored at GSX were: does MAPI impact the end-user performance? How does it change the network’s consumption? Will you have performance tickets when switching from one protocol to another?

To best answer these questions, we decided to employ our Robot Users, with different network configurations and the two protocols. Through this article, we will divulge our findings.

MAPI over HTTP: features and benefits

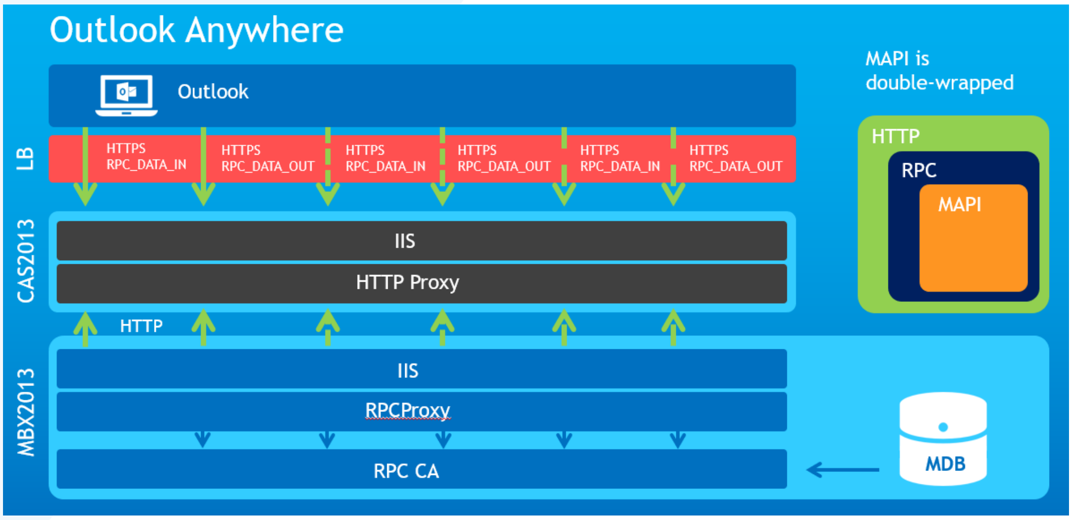
What is MAPI? MAPI stands for Messaging Application Programming Interface. If we had to describe it simply, it is just a way for a program to call functions on a messaging server.

MAPI has been around for a while. In 1991, MAPI 0 was available for MS Mail. At that time, the API was not yet officially called MAPI. First formal release of MAPI was in 1992.

Initially, it was designed just to let a local mail client running on Windows talk to a mail server running on Windows as well in the same network.

With the first MAPI, the problem was that it was designed to be used exclusively on your network, and it was created even before the current Internet existed. It just defined a series of remote procedure calls (RPC) that are also very old and did not address any security concerns. These problems made it particularly unsafe to make it available on the Internet.

Before the most recent release of MAPI, we used RPC over HTTP:

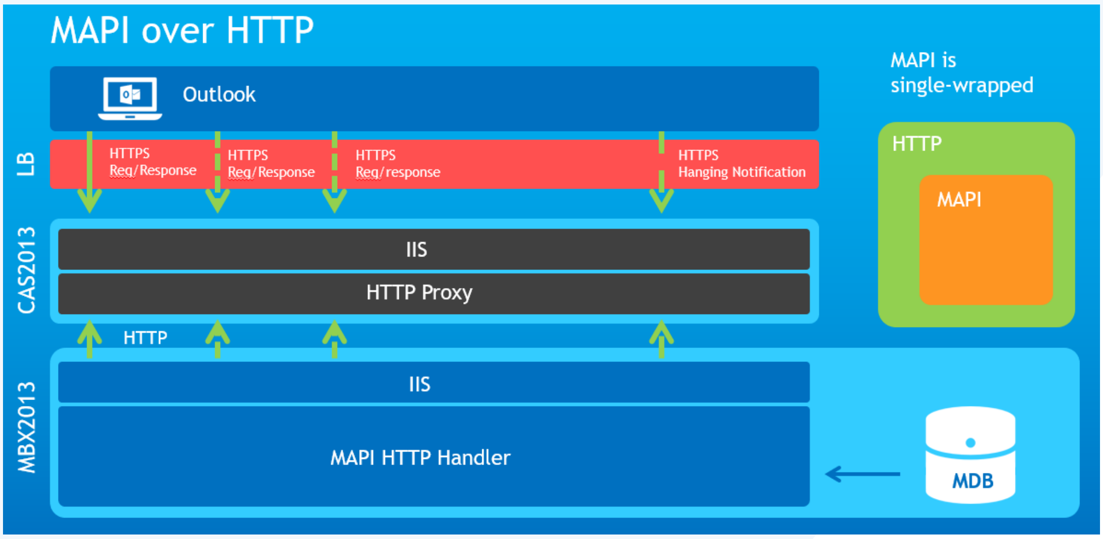


It was still MAPI, just first wrapped with RPC and then wrapped with HTTP, resulting in what we know as “Outlook Anywhere”.

Now we can understand the kind of complexity and number of calls that happen here.

First it is double wrapped and requires multiple connections; 2 long-lived TCP connections for each session between the outlook client and Exchange (Server or Online).

If we switch now on MAPI over HTTP:



We can see that the 2 long-lived TCP connections between the client and Exchange are now longer requested. The new MAPI/HTPP protocol generates a maximum of 2 current connections during this one TCP connection.

It becomes easier to manage this protocol because it is now single wrapped.

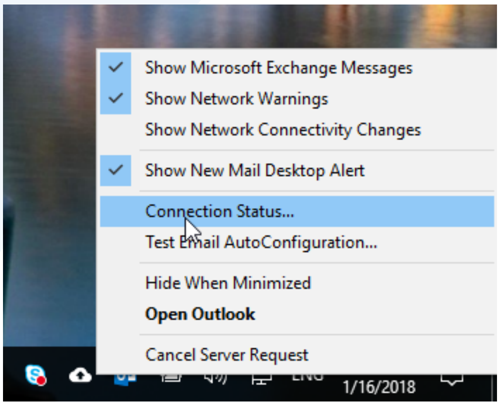
This also decouples the client-server session from the underlying network. When the network connection is lost, the session itself is not reset for 15 minutes or so, and the client can simply reconnect.

The scalability on the server side was also improved, because reestablishing a lot of these RPC over HTTP connections could strain the resources on the mailbox server at the time. Moreover, the configuration was very complex when you needed to setup RPC on 10 or 20 different servers. All this extra work was eliminated.

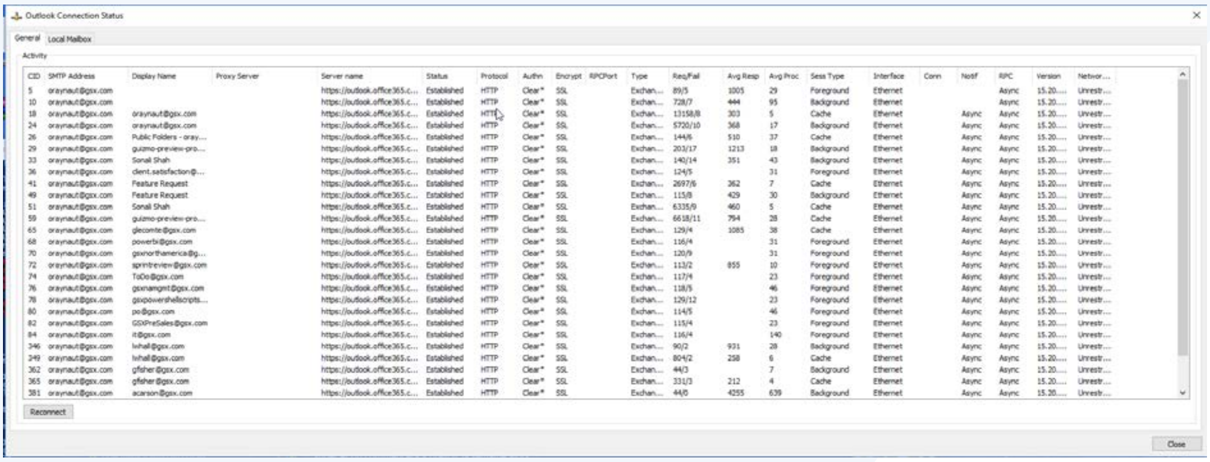
Tips and tricks

How can you find out if a client is using RPC or MAPI?

It is very easy to do this. Simply right click on the Outlook icon:


✓ Show Microsoft Exchange Messages
✓ Show Network Warnings
Show Network Connectivity Changes
✓ Show New Mail Desktop Alert
Connection Status...
Test Email AutoConfiguration...
Hide When Minimized
Open Outlook
Cancel Server Request
ENG
1/16/2018

Then you can see the connection status:



CID	SMTP Address	Display Name	Proxy Server	Server name	Status	Protocol	Authn	Encrypt	RPCPort	Type	Req/Fail	Avg Resp	Avg Proc	Sess Type	Interface	Conn	Notif	RPC	Version	Networ...
5	oraynaut@gsx.com			https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	89/5	1005	29	Foreground	Ethernet			Async	15.20....	Unrestr...
10	oraynaut@gsx.com			https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	728/7	444	95	Background	Ethernet			Async	15.20....	Unrestr...
18	oraynaut@gsx.com	oraynaut@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	13158/8	303	5	Cache	Ethernet		Async	Async	15.20....	Unrestr...
24	oraynaut@gsx.com	oraynaut@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	5720/10	368	17	Background	Ethernet		Async	Async	15.20....	Unrestr...
26	oraynaut@gsx.com	Public Folders - oray....		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	144/6	510	37	Cache	Ethernet		Async	Async	15.20....	Unrestr...
29	oraynaut@gsx.com	guizmo-preview-pro....		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	203/17	1213	18	Background	Ethernet		Async	Async	15.20....	Unrestr...
33	oraynaut@gsx.com	Sonali Shah		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	140/14	351	43	Background	Ethernet		Async	Async	15.20....	Unrestr...
36	oraynaut@gsx.com	client.satisfaction@...		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	124/5		31	Foreground	Ethernet		Async	Async	15.20....	Unrestr...
41	oraynaut@gsx.com	Feature Request		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	2697/6	362	7	Cache	Ethernet		Async	Async	15.20....	Unrestr...
49	oraynaut@gsx.com	Feature Request		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	115/8	429	30	Background	Ethernet		Async	Async	15.20....	Unrestr...
51	oraynaut@gsx.com	Sonali Shah		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	6335/9	460	5	Cache	Ethernet		Async	Async	15.20....	Unrestr...
59	oraynaut@gsx.com	guizmo-preview-pro....		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	6618/11	794	28	Cache	Ethernet		Async	Async	15.20....	Unrestr...
65	oraynaut@gsx.com	glecomte@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	129/4	1085	38	Cache	Ethernet		Async	Async	15.20....	Unrestr...
68	oraynaut@gsx.com	powerbi@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	116/4		31	Foreground	Ethernet		Async	Async	15.20....	Unrestr...
70	oraynaut@gsx.com	gsxnorthamerica@g...		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	120/9		31	Foreground	Ethernet		Async	Async	15.20....	Unrestr...
72	oraynaut@gsx.com	sprintreview@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	113/2	855	10	Foreground	Ethernet		Async	Async	15.20....	Unrestr...
74	oraynaut@gsx.com	ToDo@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	117/4		23	Foreground	Ethernet		Async	Async	15.20....	Unrestr...
76	oraynaut@gsx.com	gsxnamgmt@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	118/5		46	Foreground	Ethernet		Async	Async	15.20....	Unrestr...
78	oraynaut@gsx.com	gsxpowershellscripts....		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	129/12		23	Foreground	Ethernet		Async	Async	15.20....	Unrestr...
80	oraynaut@gsx.com	po@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	114/5		46	Foreground	Ethernet		Async	Async	15.20....	Unrestr...
82	oraynaut@gsx.com	GSXPreSales@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	115/4		23	Foreground	Ethernet		Async	Async	15.20....	Unrestr...
84	oraynaut@gsx.com	it@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	116/4		140	Foreground	Ethernet		Async	Async	15.20....	Unrestr...
346	oraynaut@gsx.com	lwhall@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	90/2	931	28	Background	Ethernet		Async	Async	15.20....	Unrestr...
349	oraynaut@gsx.com	lwhall@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	804/2	258	6	Cache	Ethernet		Async	Async	15.20....	Unrestr...
362	oraynaut@gsx.com	gfisher@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	44/3		7	Background	Ethernet		Async	Async	15.20....	Unrestr...
365	oraynaut@gsx.com	gfisher@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	331/3	212	4	Cache	Ethernet		Async	Async	15.20....	Unrestr...
381	oraynaut@gsx.com	acarson@gsx.com		https://outlook.office365.c...	Established	HTTP	Clear*	SSL		Exchan...	44/6	4255	639	Background	Ethernet		Async	Async	15.20....	Unrestr...

Microsoft has added a protocol column where you can see if you’re currently using RPC/HTTP or MAPI/HTTP (it would show HTTP). This feature is only in Outlook 2013 and 2016.

What version of Outlook can do MAPI?

All versions starting from Outlook 2010, but it is necessary to have the right updates: service pack 2 and 10 cumulative update that you need to apply manually.

If you’re on Outlook 2013, you just need the service pack one and Outlook 2016 will do MAPI/ HTTP by default.

Exchange 2019 is really planned to ship without RPC.

So let’s go now to our experiment and their results in determining which of these protocol is faster.

Our experiment

We achieved 3 different use cases running MAPI/ HTTP versus RPC/ HTTP.

- A first deployment was on Outlook 2016 client with Windows 10
- The second was on Outlook 2016 with Windows 10 but in a network that experienced issues (packet loss, latency, etc.)
- And the last one was on Outlook 2013 with Windows 7, as we know that many companies still use this configuration.

Let’s now move on to the data.

So how do we collect this data?

As explained in other articles, GSX provides Robot Users that perform end-user experience tests as well as network latency checks. Their actions aim at collecting metrics about the end-user experience and the underlying network. Theses end-user scenarios are specific to each suite of Office 365 services.

In this article, we will only speak about Microsoft Exchange.

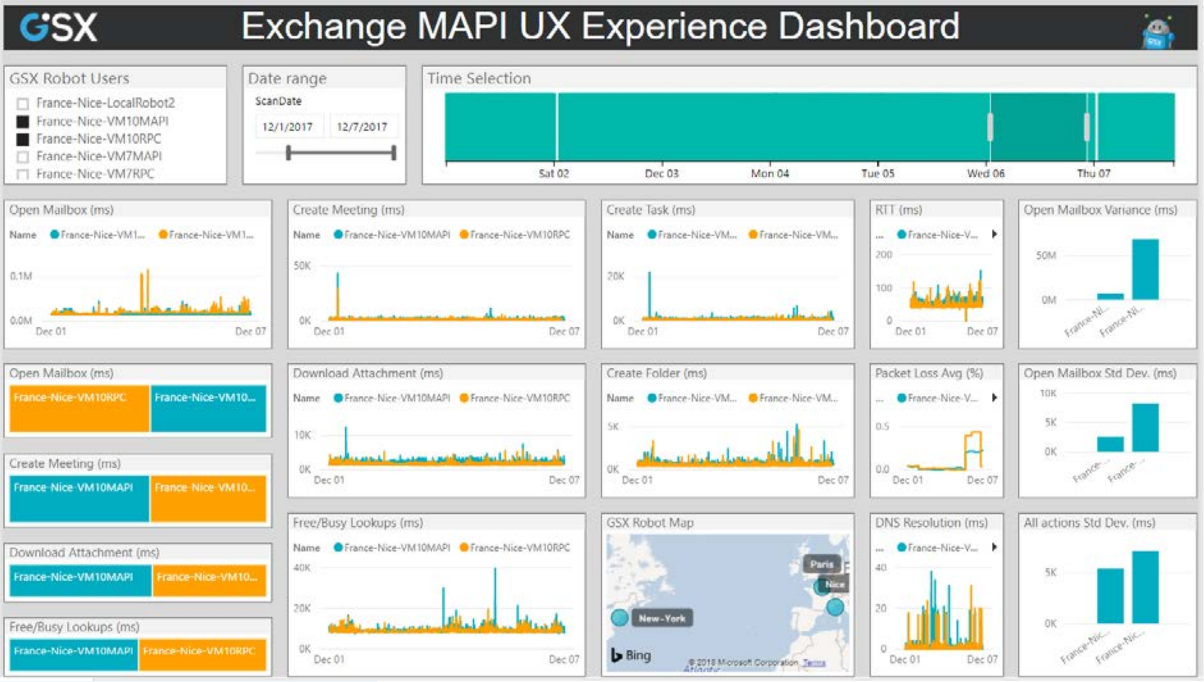
Read more about GSX Robot Users here >>

We are then using PowerBI to display the data to report and analyze the information.

In order to have more information about the type of graph we are using in PowerBI and how to read them, please read our dedicated blog here >>

So let’s come back to the results.

Results in a clean environment with Outlook 2016 & Windows10



GSX Robot Users
- □ France-Nice-LocalRobot2
- ■ France-Nice-VM10MAPI
- ■ France-Nice-VM10RPC
- □ France-Nice-VM7MAPI
- □ France-Nice-VM7RPC

Date range
ScanDate 12/1/2017 12/7/2017

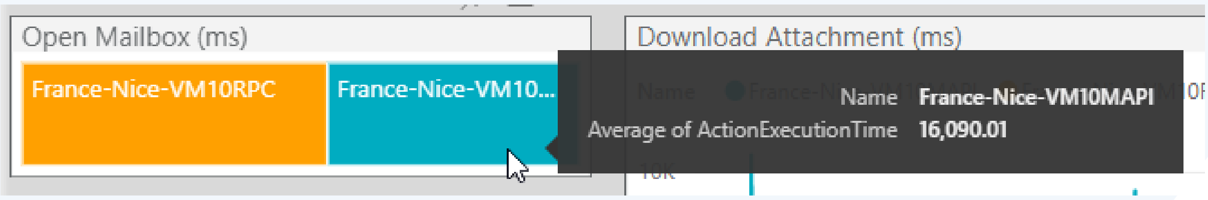
The dashboard we see here provides the results of the 2 first Robot Users we set up in order to compare the performance between RPC/HTTP (yellow statistics) and MAPI/HTTP (blue statistics).

This dashboard gathers several days of these metrics. First, we looked at the open mailbox that performs a full auto discover to retrieve all the different settings, connect to that mailbox and authenticate to access the mailbox.

But checking an “Open Mailbox” action is not enough. We have seen in other RoboTech articles that performance can greatly vary, for example between an Open Mailbox and a Free/Busy lookup. Therefore, our tests simulate multiple scenarios with the GSX Robot Users to assure the most accurate results.

For these tests we add the actions: download an attachment, perform a free/busy lookup and create a meeting.

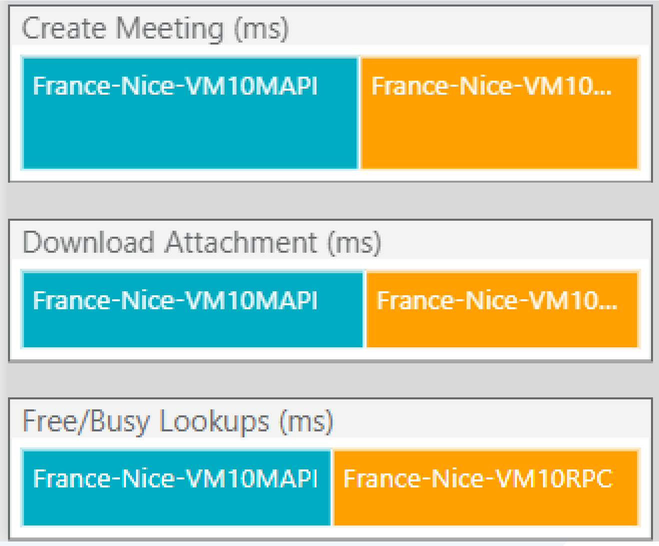
We expected to see better overall performance with MAPI as the protocol can better reconnect to a session. Indeed, we see this with the Open Mailbox.



Name France-Nice-VM10MAPI
Average of ActionExecutionTime 16,090.01

As you can see here, there is a gain of about 25% performance in establishing the connection between the client and the mailbox when using MAPI over HTTP.

However, when comparing this to the performance of the other actions, we then realize that they are 5 to 20% slower when using the MAPI protocol.

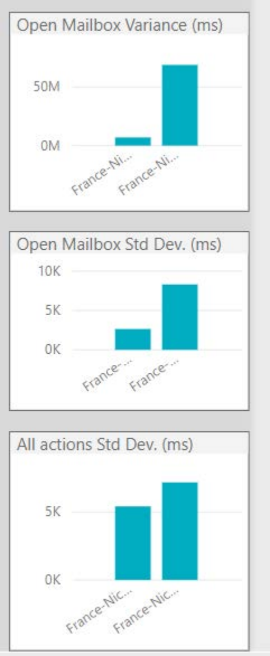


How can we explain this discrepancy?

First, it needs to be understood that we are looking at the raw protocol performance in a very static configuration. This means that these scenarios are run on virtual machines that are not being disconnected from the network, not running on wireless, and not suffering from any problems that could arise when working at home or at a cafe.

Secondly, no matter the performance, it is important to consider their stability. On the right side of the dashboard you can see some statistics calculations (Variance and Standard Deviation) that provide some insight about the stability of performances.

If you want more information about Variance and Standard Deviation please read our blog here >>

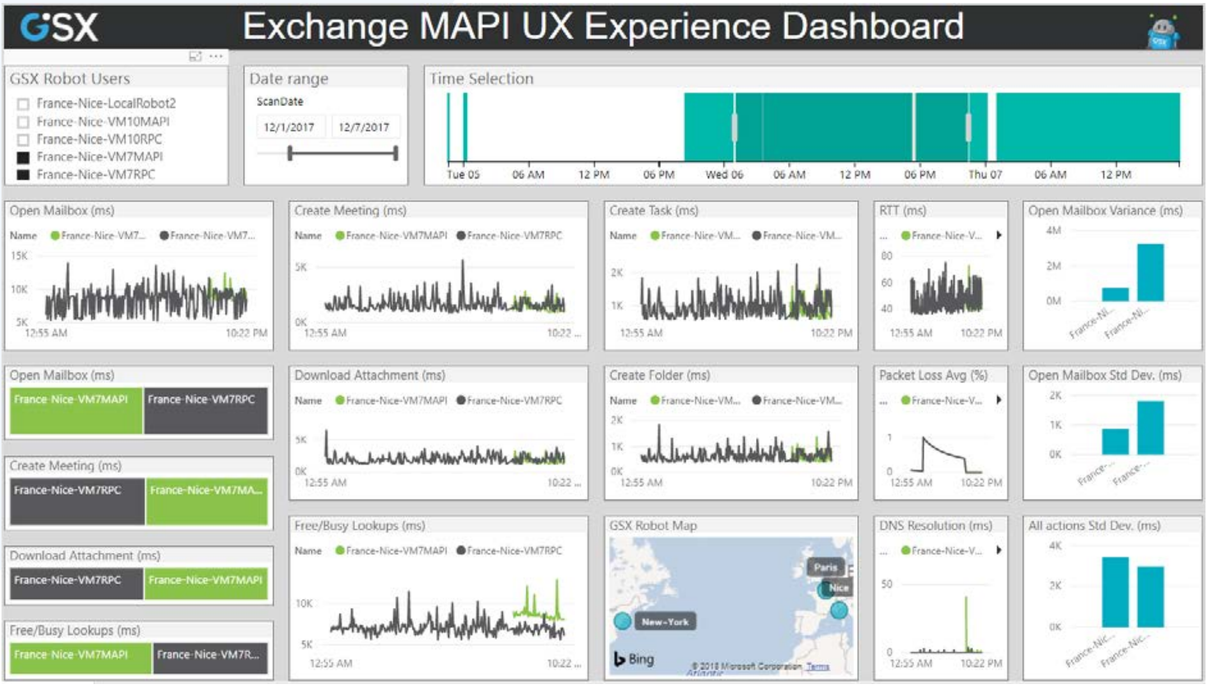


Here you can see the dramatic difference between MAPI/HTTP and RPC. The connection is much more stable with MAPI, and does not vary as much on MAPI than RPC.

To sum up these findings, MAPI brings more stability and speed when connecting to the mailbox, but does not necessarily improve performance for other daily actions that are performed by an Outlook client.

Let’s see our second set of tests.

MAPI/HTTP versus RPC/HTTP with Outlook 2013 on Windows 7 machines

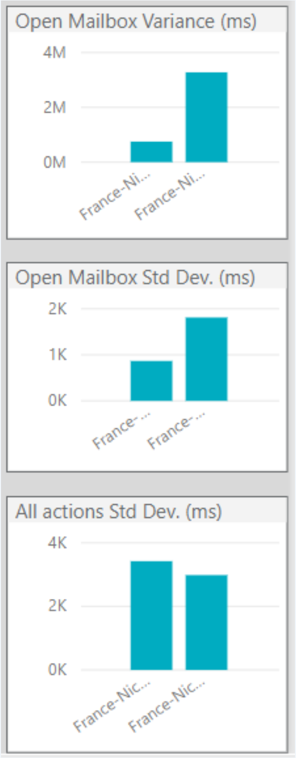


As with the previous tests, we have one Robot User using MAPI/ HTTP (Grey statistics) and the second one running RPC/ HTTP (Green statistics).

On the performance side, we saw surprisingly contrasting sets of results. The time taken for connections and other actions varied a lot.

Depending on the timeframe we analyzed, sometime it seemed as if RPC was faster when initiating the connection to the mailbox and slower on the other actions, while sometimes MAPI seemed to have a better performance…

However, we can confirm the former conclusion that MAPI is much more stable by checking the statistics calculations on the right side of the dashboard.

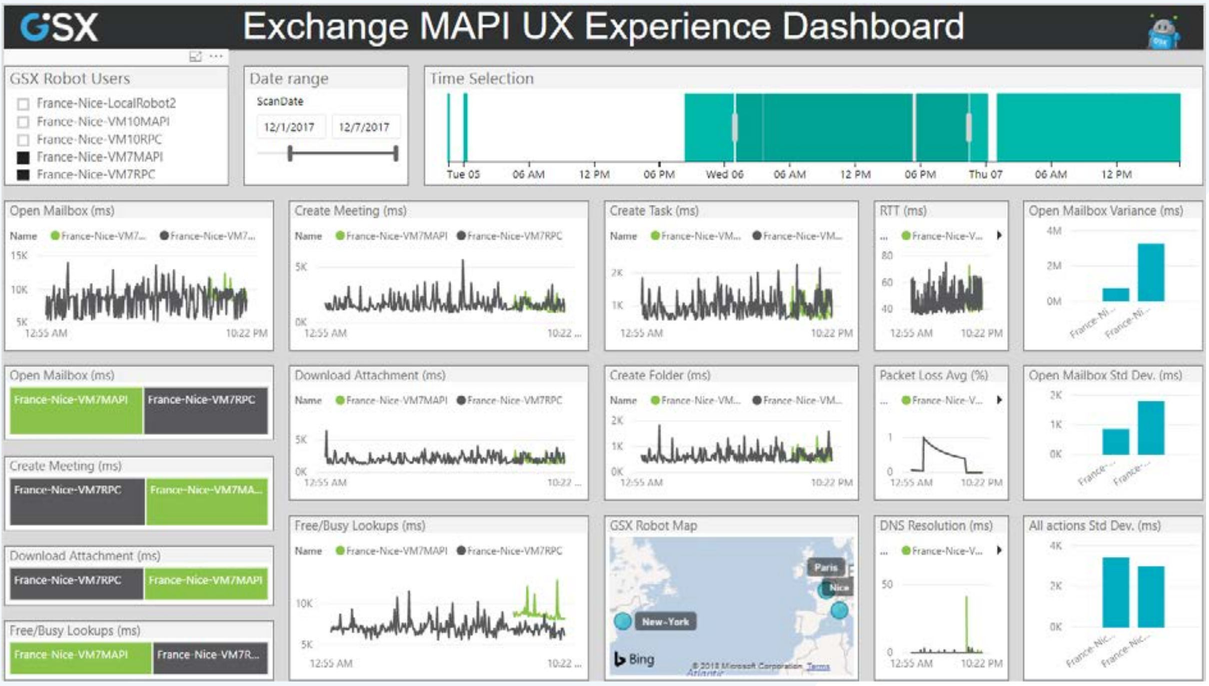


In the next phase, we aimed to see how both the protocol react when the network is in trouble.

MAPI/HTTP versus RPC/HTTP while running on poor network performance

For this test we set up different network environments with packet loss, latency, throttling, packet duplication, and packets out of order, to see how the protocol would respond. The purpose was to simulate realistic situations that could happen in the everyday life of a company with multiple remote locations and people working from home or on the road.

Results showed 5% packet loss and 50 milliseconds of added latency, knowing that it is already going across the Atlantic (Robot User in Philadelphia accessing a tenant in Europe).



Once again it confirmed the first set of data we presented.

The connection time was always slower for RPC. The difference was not extreme (10%) but further confirmed our first findings.

From most articles, one would think that in these environments MAPI/ HTTP would perform much better than RPC.

While MAPI/HTTP was designed for better resiliency, which it does provide, it did not make much difference in other areas.

In order to find a solution, we started to take down a router for about 2 second out of every 6, and then we put a proxy server in the middle that was throwing away packets.

But no matter what we tried, we struggled to actually come up with a scenario where, other than being better at reconnecting the mailbox, MAPI/HTTP actually had a better and faster overall performance.

Data Volume

When we looked at the 2 Robot Users in Philadelphia and isolated the amount of data coming out, we saw that there were definitely fewer TCP sessions established, but that they also were sending large amounts of data when compared to the RPC protocol, occasionally even twice as much data volume.

Despite these interesting results, it was expected to be the contrary, as MAPI is supposed to be a thinner protocol.

Even if the MAPI/HTTP protocol sometimes had to move twice the volume of data in contrast to the RPC protocol, it only had a maximum of 10% difference in performance compared to the RPC protocol. When moving the data, MAPI is very efficient, but that doesn’t necessarily mean a smaller volume of data.

This corresponds to MAPI's design, which is to improve connection resiliency, not to send a smaller amount of data.

Conclusion of our tests

The shift to MAPI is already in motion. If you are on Office 365, you're most likely already using this protocol, and if you move from on-premises to Exchange 2016 that is definitely the protocol that you will want to use.

From a protocol standpoint, MAPI/ HTTP is more in line with the modern world, as we are always connecting from different networks and locations nowadays.

As usual, before implementing any change in your environment, you need to do some careful planning.

And that is what the GSX Robot Users are designed for.

MAPI is intended to provide a better connection and reconnection to the mailbox, which is what we found while running our tests. It has not been designed to improve the performance of every action you can perform with an Outlook client in your day-to-day life. You should check the impact of this new protocol on your end-user experience before rolling it out.

From a data perspective, you should expect a greater amount of data flowing on your network when moving to MAPI.

Of course, the amount of increase all depends on your network configuration and the way your messaging environment is used. Once again, it is a good idea to measure it in order to anticipate any bandwidth consumption increase in the future.

Finally, moving to a new protocol is not necessarily only a question of performance. One of the biggest benefits of MAPI, besides its resiliency, is better, more modern authentication. So if you're doing multi-factor authentication, you still may want to take a look at it..