**Case 1 : Client and Server on Google Cloud**

**Chart, application, line chart

Description automatically generated**

In the above situation the NTP Server was run on Google Cloud and Client on a local machine. We can see that delays are anywhere between 22ms to 45ms. This is pretty normal for a roundtrip delay. More consistent the better.

In the offset, we see that initially there’s barely any offset and over time during different bursts, we see the offset increase and then go back to being minimal. This is probably because of the clock on the cloud instance synchronizing with a different time server periodically. So the local client was almost 1s off the NTP server at one point.

More charts and raw data in the excel file.

**Case 2 : Client and Public NTP Server**

**Chart, line chart

Description automatically generated**

For this test, time.google.com server was used. We can observe that the delay is very consistent other than a couple outliers. When it comes to offset we see a similar story. As out client starts synchronizing with the NTP server itself, we can see how the client local time drifts off over time about 80ms every 4 minutes. This is again probably due to the time server itself synchronizing itself and the client clock synchronizing with a different time server.

**Case 3 : Client and Server on Local**

**Graphical user interface, chart, application

Description automatically generated**

In this scenario the NTP server and client were run locally. We can see that the offset keeps growing over time as we saw in other cases. The Client and Server run under WSL which synchronizes with the host machine which in turn synchronizes with an NTP server. This multi-layer synchronization causes a drift in the offset.

In all 3 cases, we can see that with a smaller and also importantly symmetric round trip delay will give the most accurate estimation of the current time since we are dividing the delay / 2 to estimate the server’s time during the synchronization instant.