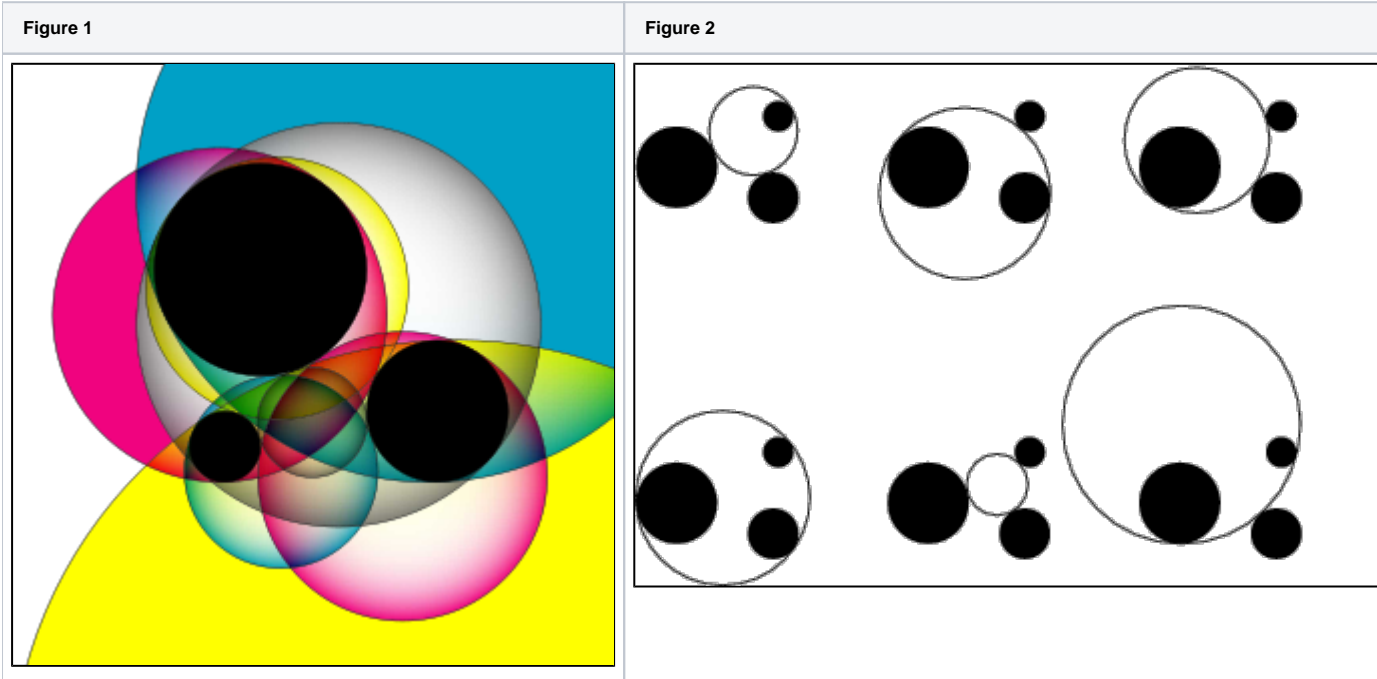


# TULIP Algorithm Apollonius Technique and its Performance

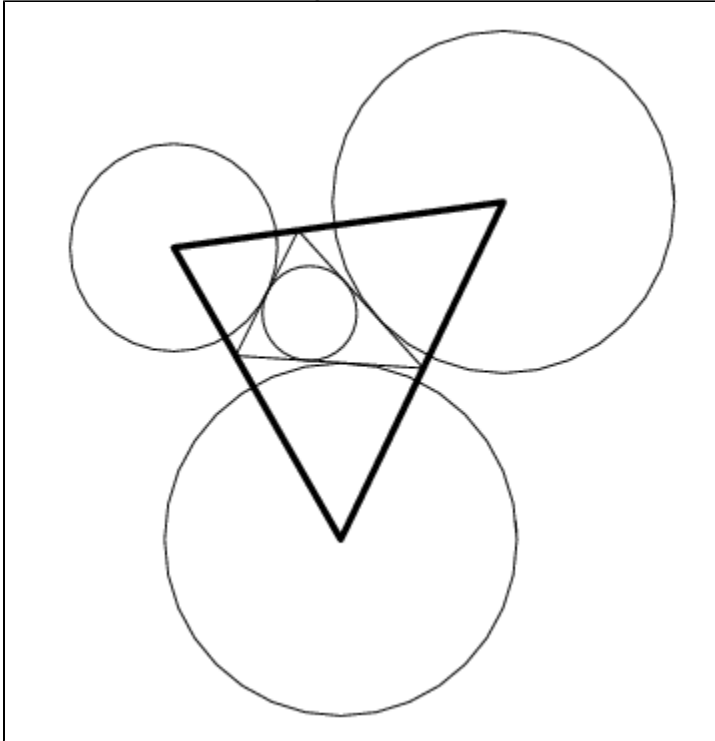
Another quite efficient geo location technique is Apollonius which also relies on RTT delay data from the known location landmarks to the unknown location target to estimate the target location. The main difference between Trilateration and Apollonius is the way in which the two techniques estimate the location. Contrary to Trilateration, the Apollonius technique doesn't take the overlapping region of the three circles into consideration; instead it draws the tangent circles touching all three landmark circles. In this way, Apollonius can result in the formation of multiple ( $\leq 8$  or sometimes even no) solution circles. Figure 1 below shows, in black, the three initial circles whose radii are derived from the RTT measurements and centers are the known landmark circles. Figure 1 below shows, in black, the three initial circles whose radii are derived from the RTT measurements and centers are the known landmark circles. The colored circles show the various solution circles that can be drawn tangential to the three black circles. There can be up to 8 such solution circles. An example of 8 such solution circles tangential to the measured black initial circles is shown in Figure 2. Figure 3 shows all the circles super-imposed



A key part is then to select one candidate circle out of these multiple solution circles. In this technique, the center of the finally selected Apollonius circle is considered as the target location. The circle selection in Apollonius is a critical part and is explained here ( <https://confluence.slac.stanford.edu/display/IEPM/Circle+selection+for+Apollonius+results> )

There are many ways to solve construct Apollonius circle.

1. Solve for the three initial circles that are the **excircles** of a triangle, see figure below. As shown in the figure below, every triangle has 3 excircles. A circle inside three circles and tangent to them is the minimal solution circle of Apollonius.



2. Solve three arbitrary circles. i.e at any position (except they are not nested within each other) and find solution circles that can be at max 8 circles. and one of these eight solution circles is the final Apollonius circle. (Figure 1, Figure 2, Figure 3)

We have implemented the second case. i.e used the three measured initial circles to find up to 8 Apollonius solution circles that are tangent to the three initial circles. One of these eight circles is the final Apollonius Circle (we use the cluster method to find that circle).

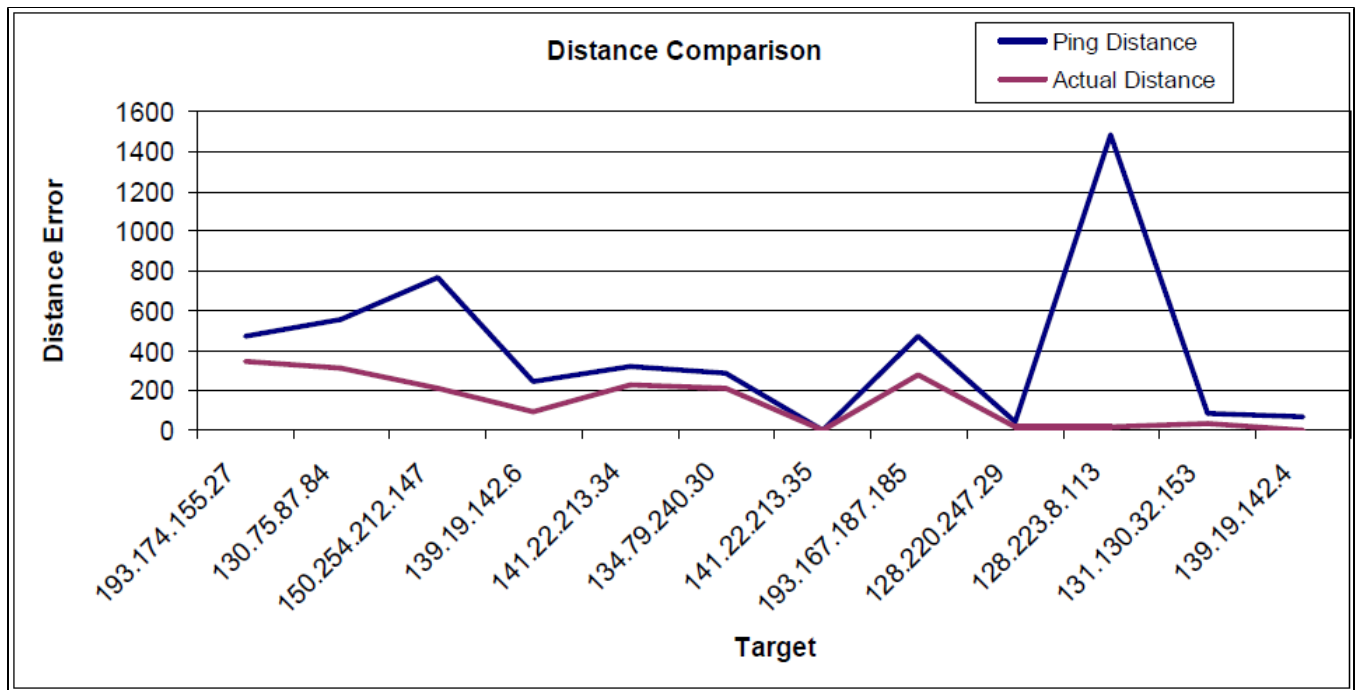
## Application

In our algorithm to solve IP based Geolocation we have embedded the Apollonius technique to locate the required target. To get the initial three circles of Apollonius we used the minRTTs from the three landmarks with the lowest minRTTs obtained from pinging the target node. Each of these three minRTTs is then used to estimate the  $distance = minRTT (ms) * alpha * 100 \text{ km}$  to get the radius of the circle. In this way we get the three initial circles of Apollonius which are used by our algorithm to find the solution circles that are tangent to the three initial circles. The resultant solution circles are up to eight in number or less. After getting eight or fewer solution circles we need to find the one final circle that is near our target. We have used the cluster approach to find the final circle and then compared this technique with a manual method to view the performance of our cluster approach. (<https://confluence.slac.stanford.edu/display/IEPM/Circle+selection+for+Apollonius+results>)

## Performance of Apollonius

Like in trilateration, in Apollonius we are using  $distance = RTT * alpha$  to get the radius of initial three circles of Apollonius. We can not accurately analyse the efficiency of Apollonius Algorithm using delay to distance mapping. So in order to see the actual performance of Apollonius technique we performed a test. We selected known landmarks and a target (which was also a landmark, but acting as target for other landmarks) then calculated distance (<http://www.movabletype.co.uk/scripts/latlong.html>) between each landmark to that target and hardcoded this distance to code i.e replaced e.g distance= 786.00.

By doing so we do not need delay measures (RTT). Purpose of doing this task was to test Apollonius efficiency in case it does not have to suffer the ping error.



Doing analysis on results showed that 60-70% of Distance Error is because of pinging error. Hence Apollonius can perform far better if there is no pinging error.

SpreadSheet for [Results](#) and [Comparison](#)

## Reference

[http://en.wikipedia.org/wiki/Circles\\_of\\_Apollonius](http://en.wikipedia.org/wiki/Circles_of_Apollonius)<http://mathworld.wolfram.com/ApolloniusCircle.html>

<http://mathworld.wolfram.com/ApolloniusCircle.html>

[http://upload.wikimedia.org/wikipedia/commons/2/2a/Apollonius\\_circle\\_definition\\_labels.svg](http://upload.wikimedia.org/wikipedia/commons/2/2a/Apollonius_circle_definition_labels.svg)