# TULIP Calculating distance from latitude and longitude

To compute the physical distance between landmarks from the Lats and Longs of two locations we can use this tool to compute the distance http://www.movable-type.co.uk/scripts/latlong.html.

If you have the lat longs in spreadsheet columns then you could write a function (see the attached spread sheet).

Below is a perl sub you could embed in a perl script. It assumes you have a min RTT and the lat longs of source and destination. You could also extract the code to calculate the distance.

You can get the database information (e.g. name, lat/long) on the landmarks themselves from http://www-wanmon.slac.stanford.edu/cgi-wrap/reflector.cgi?function=landmarks

```
##############################################################################
# Calculate alpha given the lat longs of the source (monitor) and target (remote)
# hosts, together with the minimum measured RTT in msec between them.
# alpha=distance_between_monitor_and_remote(in kms)/(100*min_RTT(in msec)).
# South latitudes are negative, east longitudes are positive. They are in
# decimal degrees. 57.2958=180/pi coverts degrees to rdaians.
# True radius varies from 6357km (polar) to 6378km (equatorial).
# See http://www.planet-source-code.com/vb/scripts/ShowCode.asp?txtCodeId=670&lngWId=6
# If it is not possible to calculate alpha due to an invalid RTT (e.g. no digits,
# just a period (.), then getalpha returns a period.
# Example:
# getalpha($src-latlong, $tgt_latlong, $min_rtt)
# getalpha('31.77 35.23', '9.98 -84.07', 121.223) sub getalpha {
  my @monitor_latlon = split(' ', $_[0]);
  my @remote_latlon  = split(' ', $_[1]);
  for (my $i=0;$i<2;$i++) {
    unless(($remote_latlon[$i]=~/^-{0,1}\d*\.{0,1}\d+$/)
      && ($monitor_latlon[$i]=~/^-{0,1}\d*\.{0,1}\d+/)) {
      return ".";
    }
  }
  my $min_rtt        = $_[2];
  my $alpha;
  my $min_dis=100000; #Initialise to an impossibly large value >> circumference of earth.
  if (defined($min_rtt) && ($min_rtt=~/\d/) && ($min_rtt>0)) {#Is there a valid $min_rtt
     my $distance=((6378.7)*(acos((sin($monitor_latlon[0]/57.2958))
                *(sin($remote_latlon[0]/57.2958))+(cos($monitor_latlon[0]/57.2958))
                *(cos($remote_latlon[0]/57.2958))
                *(cos(($remote_latlon[1]/57.2958)-($monitor_latlon[1]/57.2958))))));
    if($distance>$min_dis) {$distance=$min_dis;}
    $alpha=$distance/(100*$min_rtt);
  }
  else {$alpha=".";}
  return $alpha;
}
```