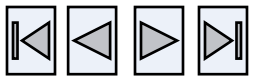
Mobile Global Positioning

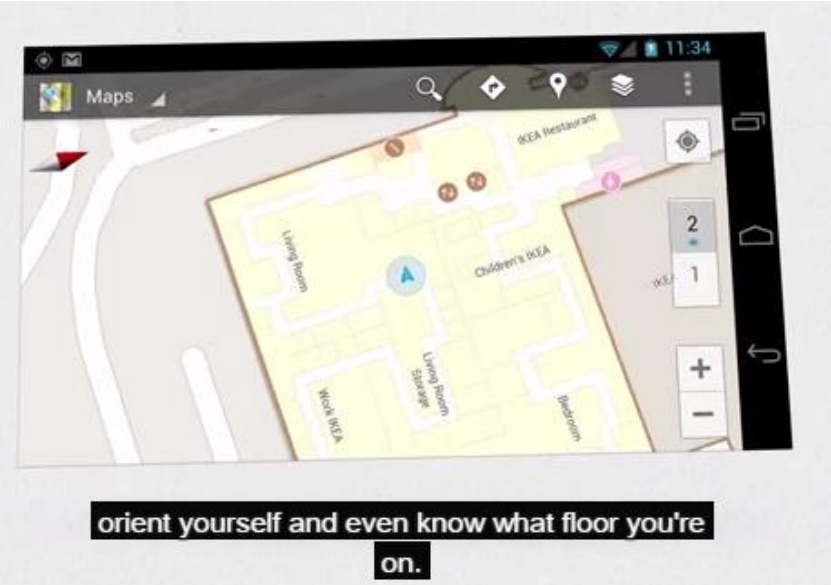
CSCI3310

Mobile Computing & Application Development

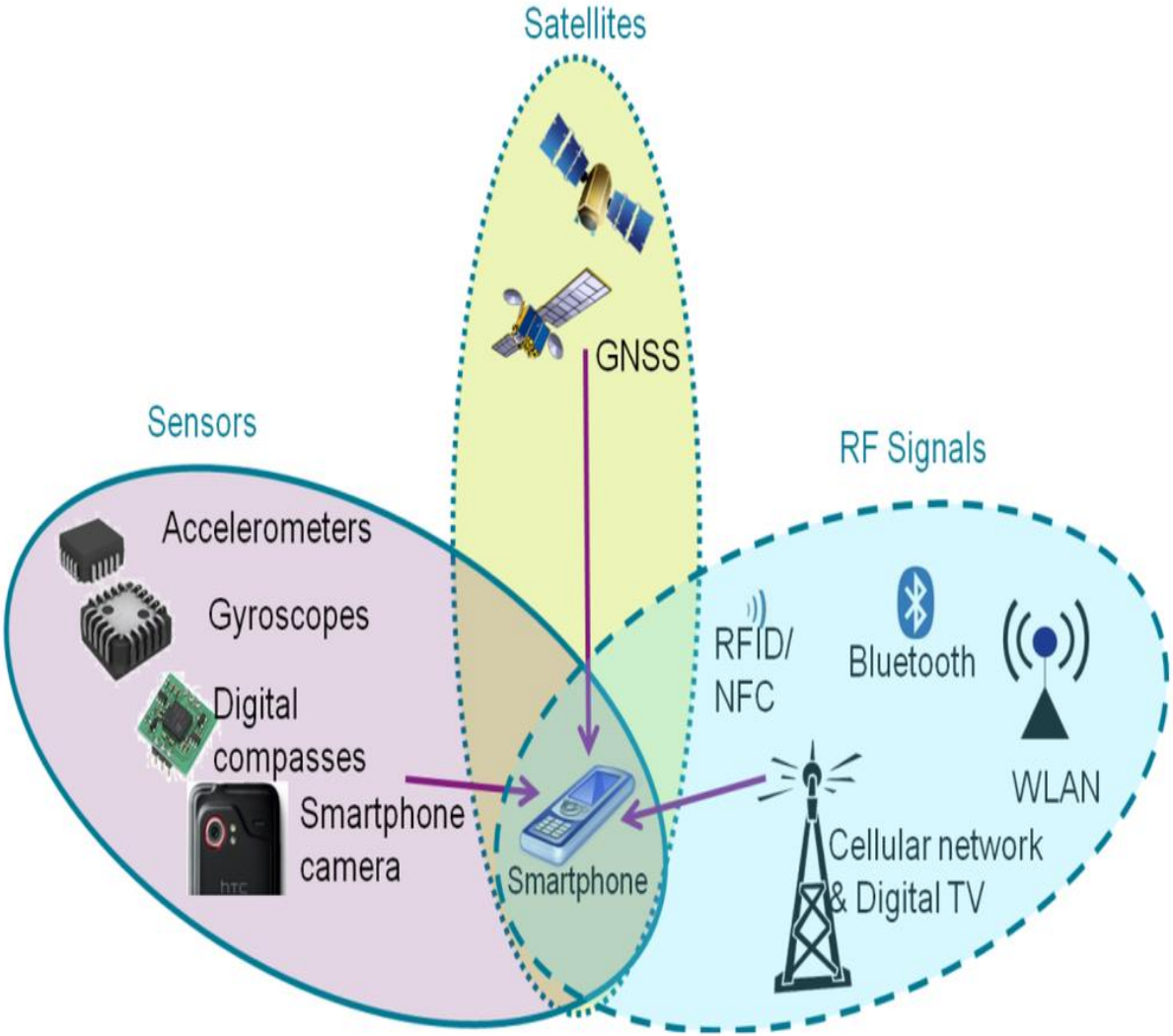


Location-based Application

- Navigation, entertainment etc
- Using location to monitor user position and as a way of connecting nearby user socially
- Can improve overall user experience
- Further classify into outdoor (e.g. GPS) and indoor positioning system

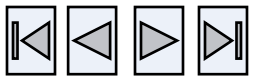


Smartphone Positioning



Outline

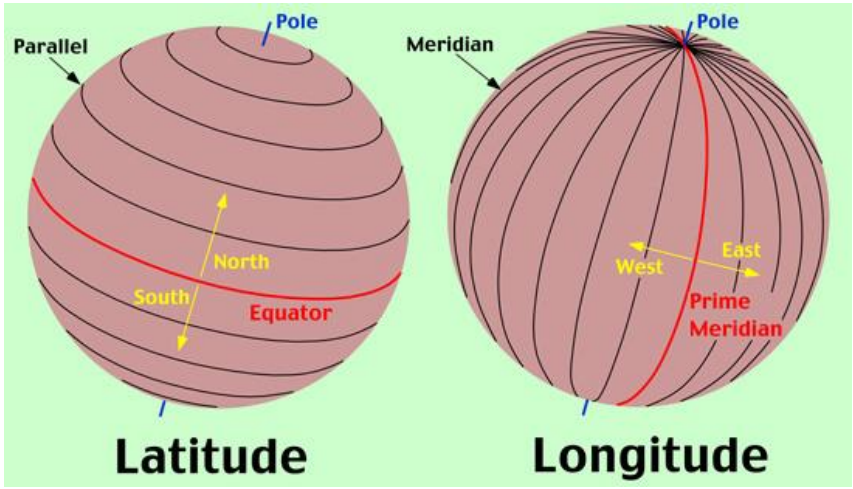
- GPS Basic Mechanism
- Trilateration and Multi-lateration
- Limitations in GPS system
- GPS/AGPS/GNSS for Android
- Remarks on using Map

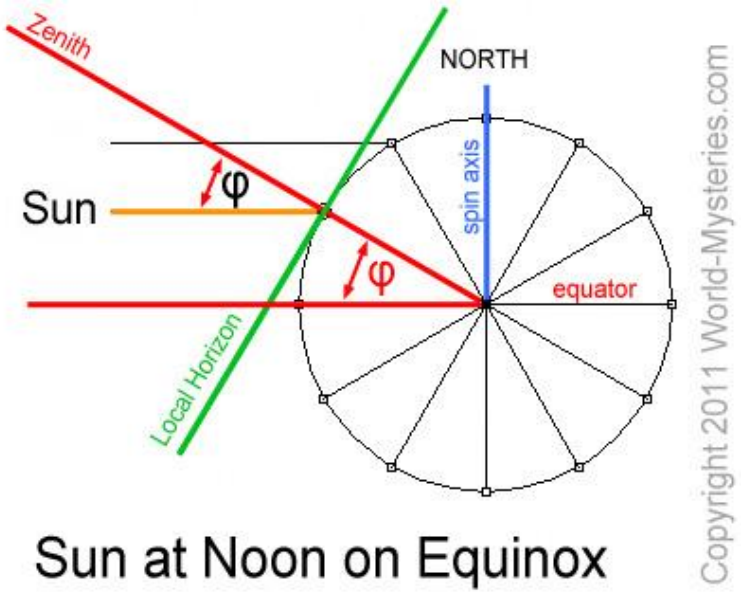


Ancient Wisdom

- On earth, north-south (Latitude), east-west (Longitude) is the important metric

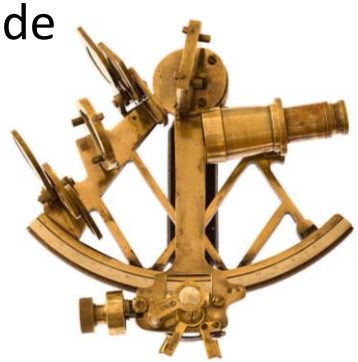
- North-south position (latitude) can be deduced by measuring angle with the sun at noon on prescribed day

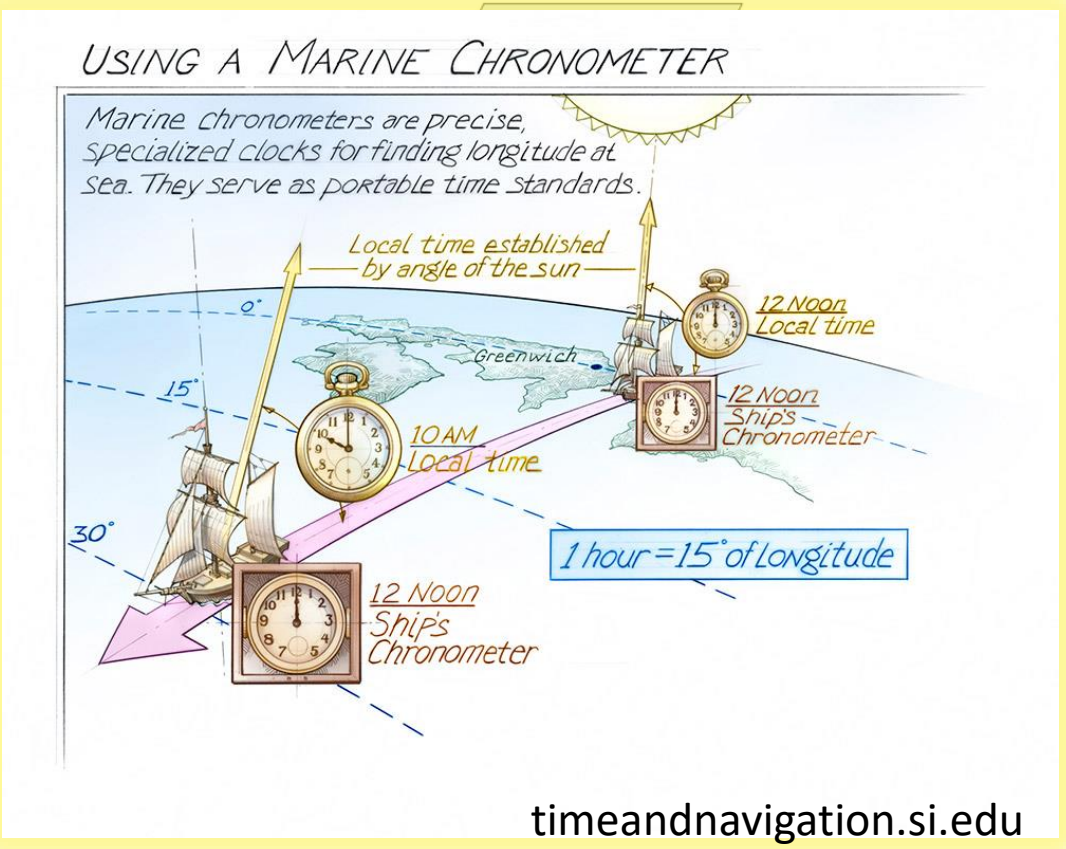




Past Wisdom

- Longitude estimation is much more difficult
- Turns out it need precise timing information
- a position line is drawn from a sight by sextant of any celestial body, crosses the observer's assumed latitude





timeandnavigation.si.edu

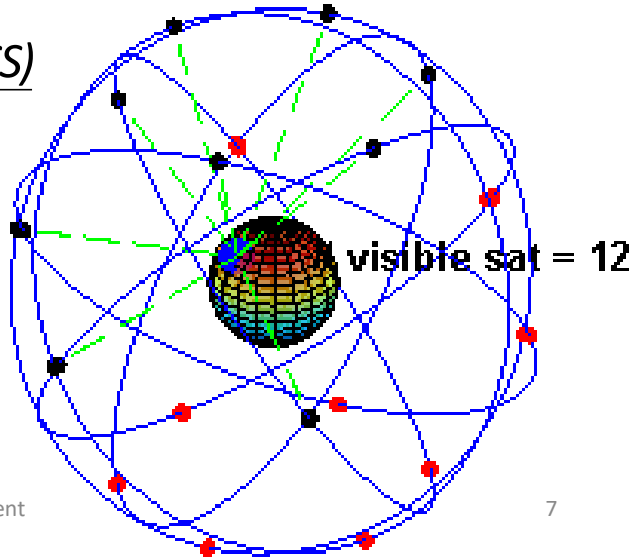
Global Navigation Satellite System (GNSS)

United States' Global Positioning System (GPS)

- provides location and time information in all weather, anywhere on or near the Earth
- launched in 1973, fully operational in 1995
- Originally run with 24 satellites, now 31 operational

Similar projects worldwide:

- Russian *GLObal NAvigation Satellite System (GLONASS)*
- European *Union Galileo positioning system*,
- China's *BeiDou Navigation Satellite System (BDS)*,
- India's *Regional Navigation Satellite System*, and
- Japan's *Quasi-Zenith Satellite System*

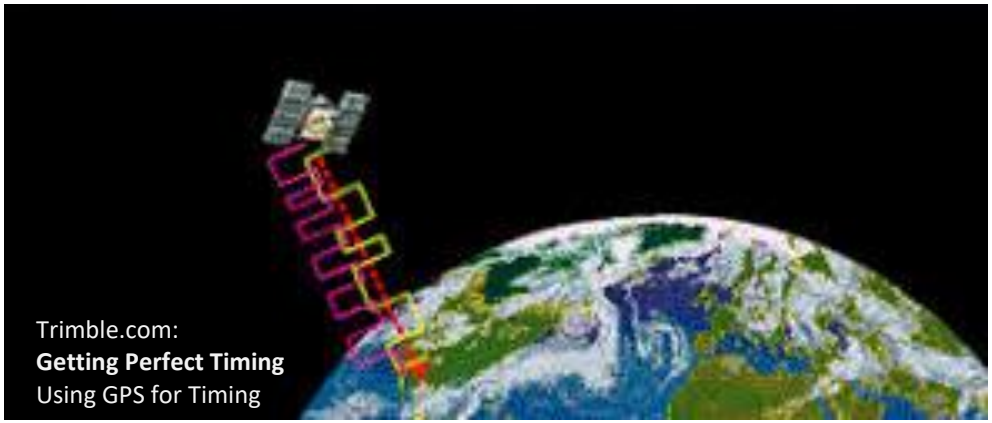


Global Positioning System

Calculates its position by precisely timing the signals sent by GPS satellites

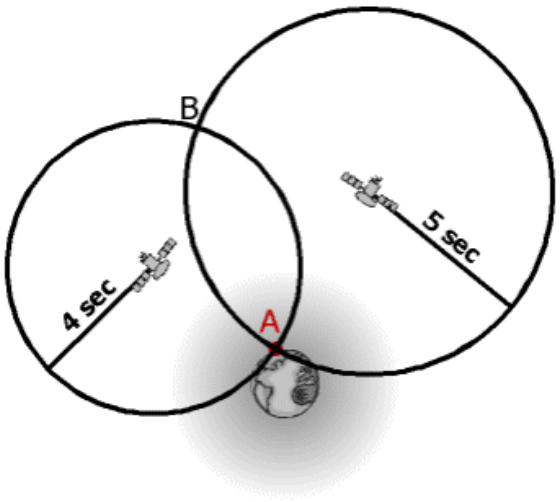
Each satellite continually transmits messages that include

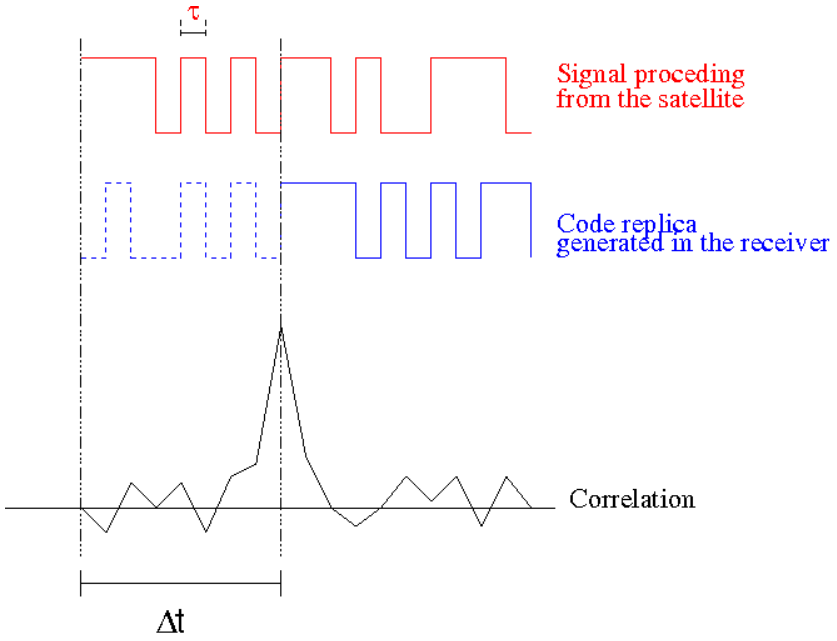
- the time the message was transmitted
- precise orbital information (the ephemeris)
- the general system health and rough orbits of all GPS satellites (the almanac).



Global Positioning System

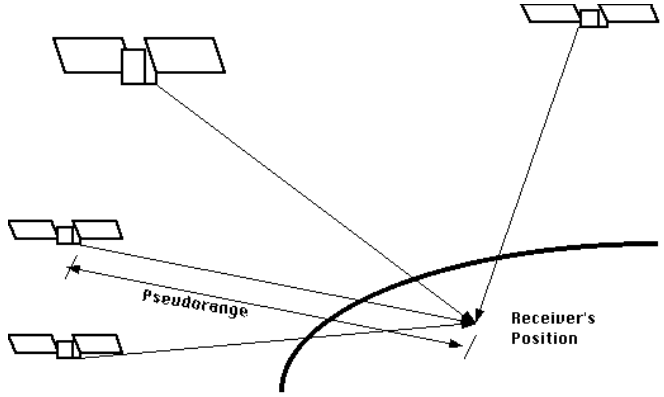
- The receiver determine the transit time of each message, aka **time of arrival (TOA)** and computes the distance to each satellite
- Along with the satellites' locations and with the possible aid of **trilateration**, to compute the position of the receiver.

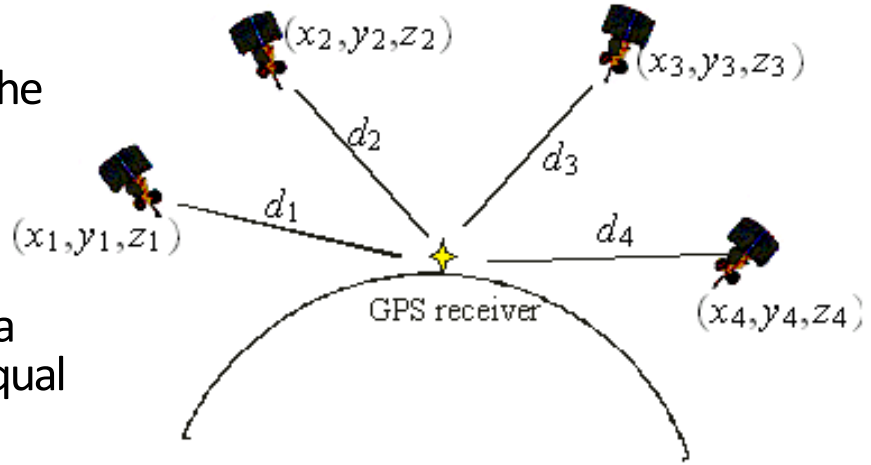




Global Positioning System

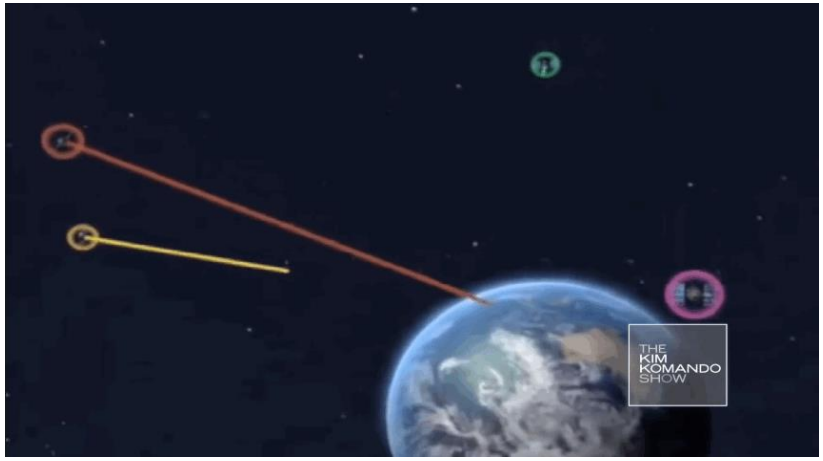
- $x, y,$ and z components of position, and the time sent by satellite are designated as (x_i, y_i, z_i, t_i) subscript i is the satellite number $\{1, 2, 3, \text{or } 4\}$
- transit time of the message as $\hat{\delta}_t = \bar{t}_r - t_i$ with the time the message was received
- A **pseudo-range**, $\rho_i = (\bar{t}_r - t_i)c$, would be the traveling distance of the message, assuming it traveled at the speed of light, c.
- A satellite's **position** and **pseudo-range** define a sphere, centered on the satellite, with radius equal to the pseudo-range.

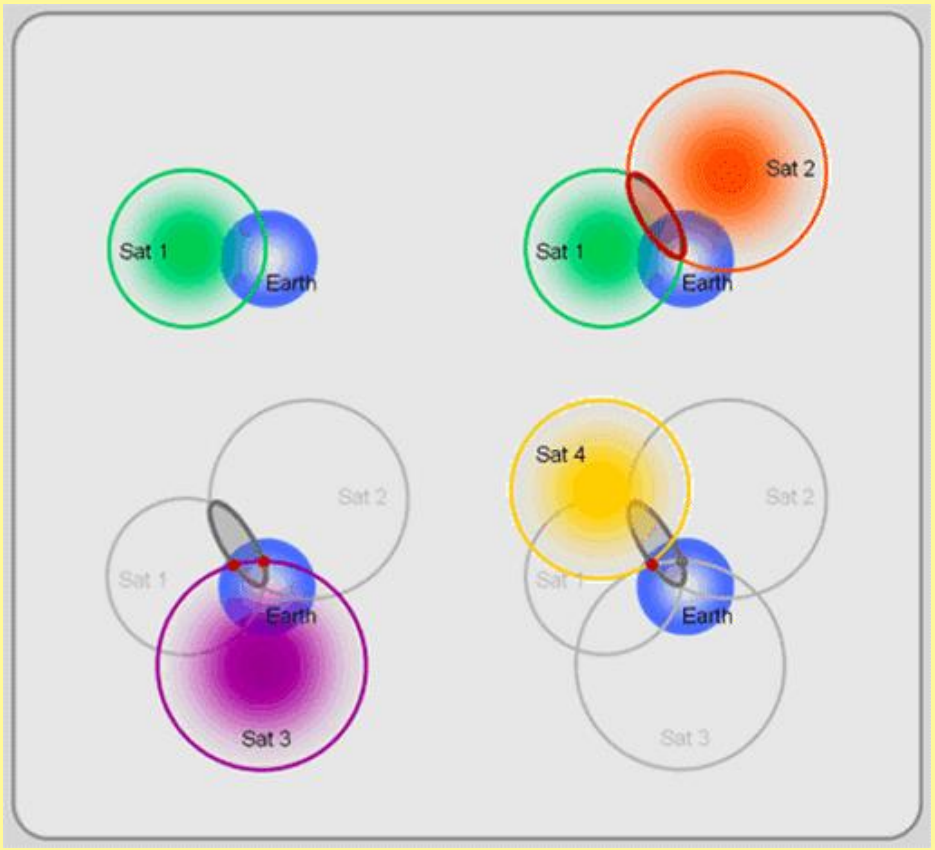




How many satellites in need?

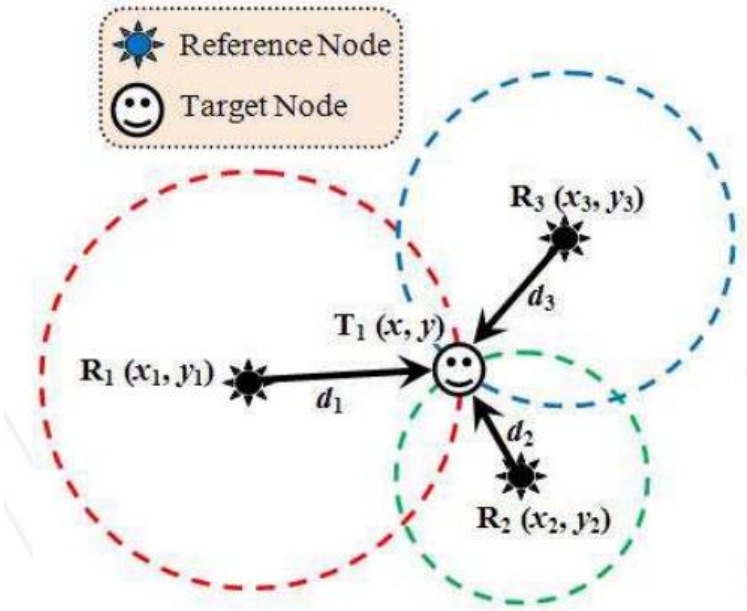
- Use four or more satellites to solve for both the receiver's location and time

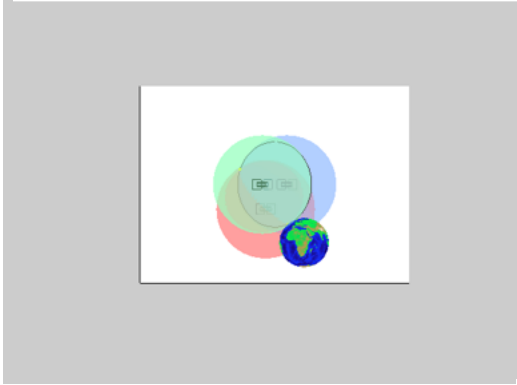




Trilateration (2D-case)

- Localization from reference locations using distance estimates
- Distance between reference locations and target location consider as radii of circles having centre at reference location
- Unknown location is the intersection of all sphere surface

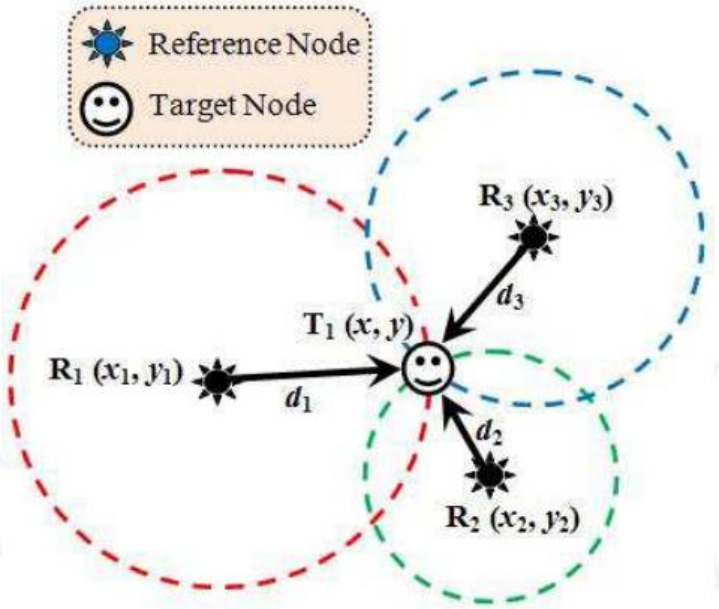




Trilateration (2D-case)

- Assume target node at exactly the intersection with given reference node coordinates

$$x = \frac{AY_{32} + BY_{13} + CY_{21}}{2(x_1Y_{32} + x_2Y_{13} + x_3Y_{21})}$$
$$y = \frac{AX_{32} + BX_{13} + CX_{21}}{2(y_1X_{32} + y_2X_{13} + y_3X_{21})} \quad (1)$$



where

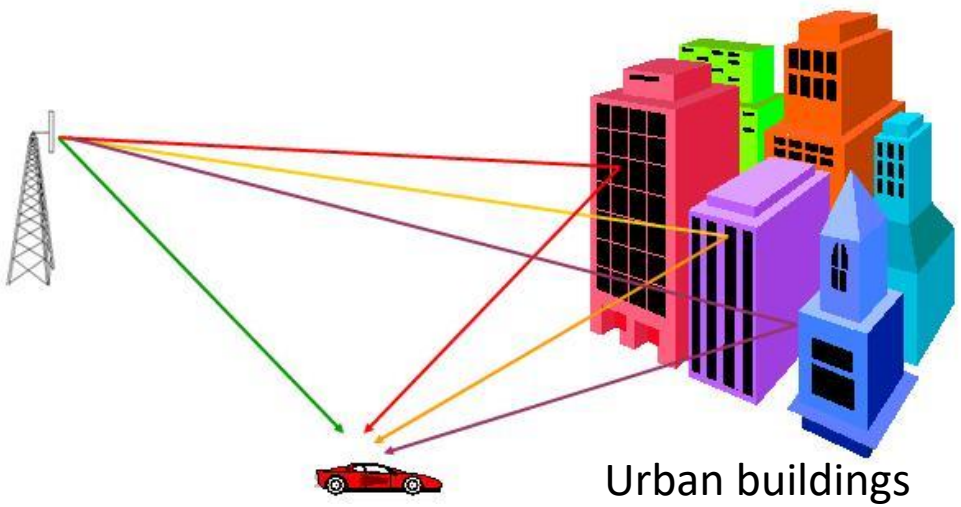
$$A = x_1^2 + y_1^2 - d_1^2$$
$$B = x_2^2 + y_2^2 - d_2^2$$
$$C = x_3^2 + y_3^2 - d_3^2$$

and

$$X_{32} = (x_3 - x_2)$$
$$X_{13} = (x_1 - x_3)$$
$$X_{21} = (x_2 - x_1)$$
$$Y_{32} = (y_3 - y_2)$$
$$Y_{13} = (y_1 - y_3)$$
$$Y_{21} = (y_2 - y_1)$$

GPS problem

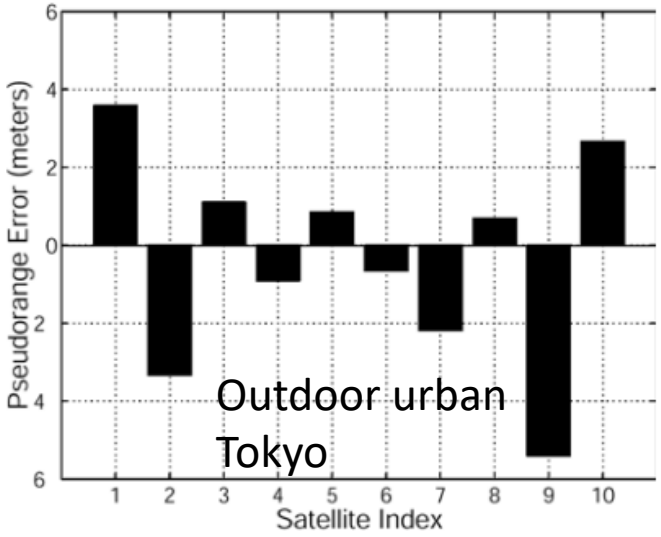
- Signal received in urban environment face the problem of multi-path effect
- propagation phenomenon that results in radio signals reaching the receiving antenna by two or more paths
- Magnitudes of the signals arriving by the various paths have a distribution (Rayleigh or Rician fading)

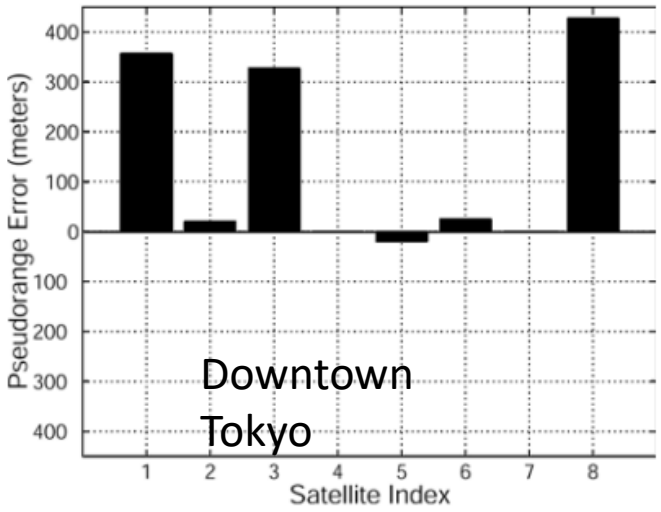


Urban buildings

GPS problem

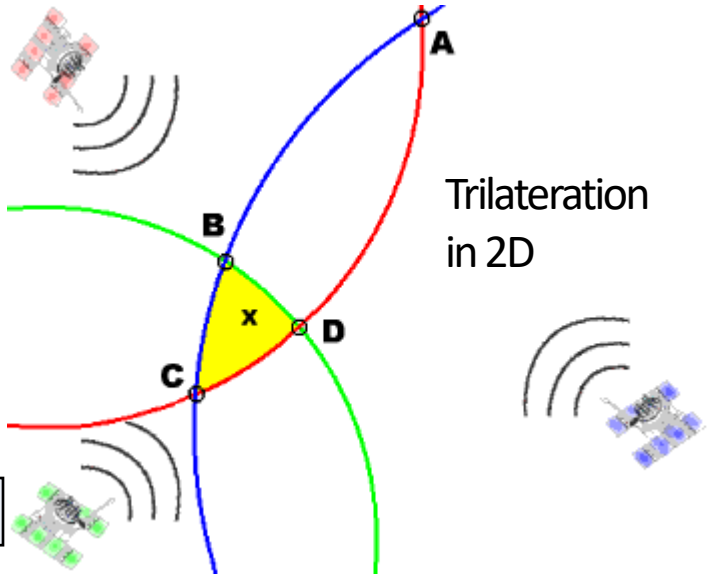
- Chart shows the effect of urban environment on pseudo-range error
- Probably due to multi-path and signal scattering by building



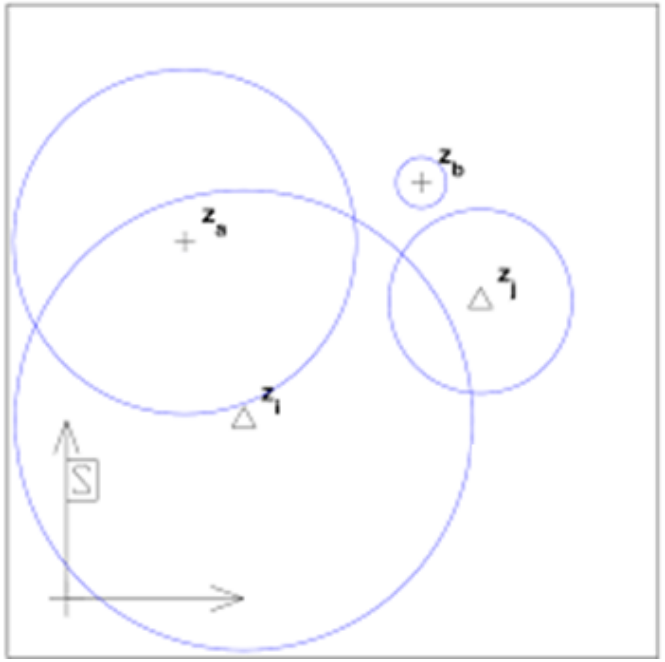


Maximum Likelihood

- How about the spheres are not intersecting in one point as expected?
- In reality, timing error exists so no exact solution might be found
- In the worse scenario, no spheres are intersecting with



Trilateration in 2D



Noise Compensation using Least Squares

- In real world, it seldom exists a solution (x,y) that satisfies the analytical form,

$$(x - x1)^2 + (y - y1)^2 = r1^2$$
$$(x - x2)^2 + (y - y2)^2 = r2^2$$
$$(x - x3)^2 + (y - y3)^2 = r3^2$$

- So, we wish to find the position (x∗,y∗) that minimizes the sum of squared differences of all three equations:

$$(x^*, y^*) = \underset{(x,y)}{\operatorname{argmin}} \sum_{i=1}^{3} [ri^2 - (x - xi)^2 + (y - yi)^2]^2$$

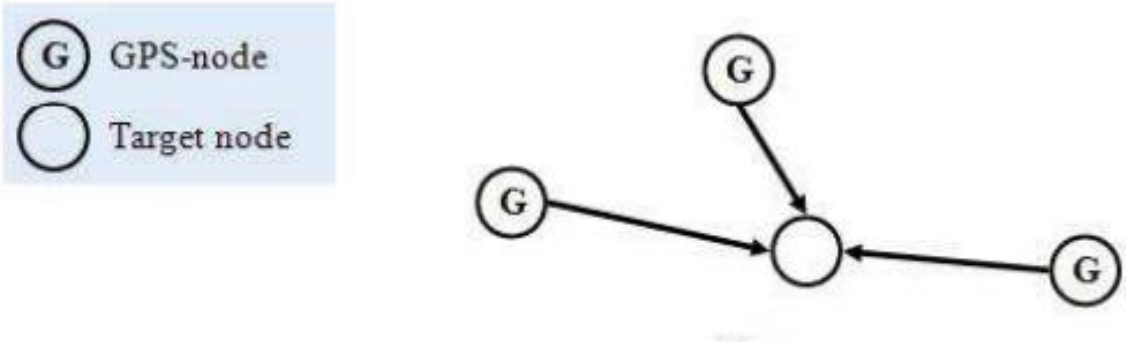
Trilateration (con't)

- For large scale network, the position information for sensors without GPS installed thus depends on whether any GPS enabled sensor nearby

- In cases where less than 3 GPS-enabled sensors nearby, localization is done via adoptive approaches:
  - Iterative
  - collaborative

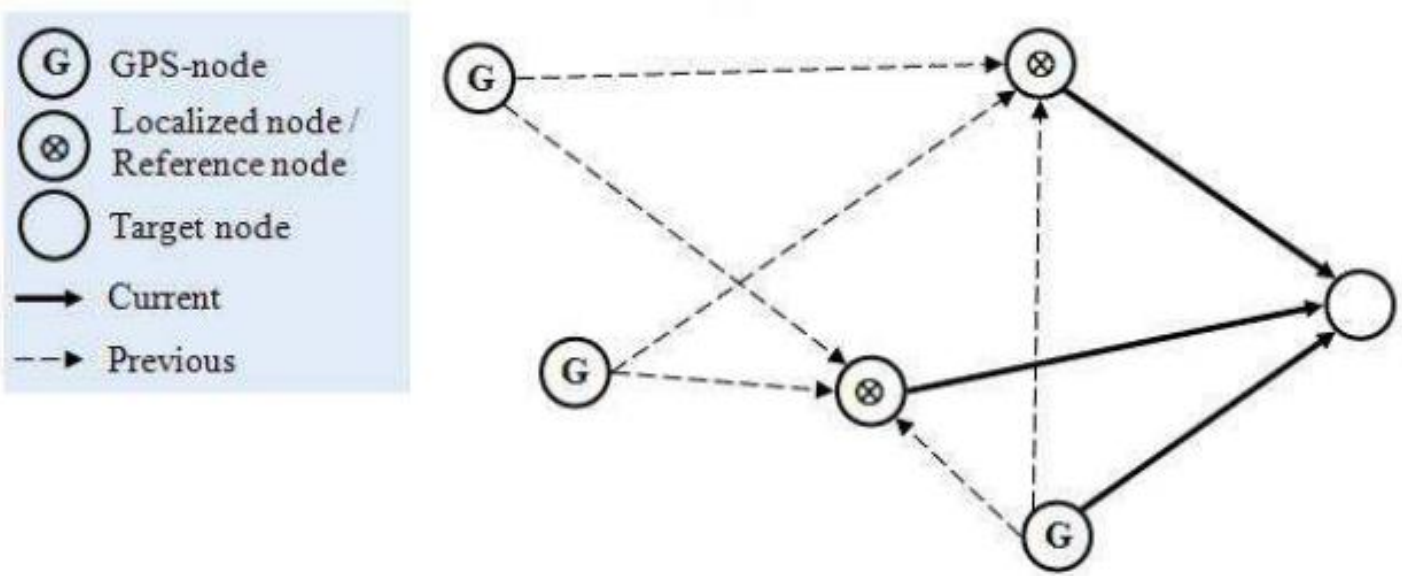
Atomic Localization (trilateration)

- At least 3 GPS-nodes needed
- Standard trilateration



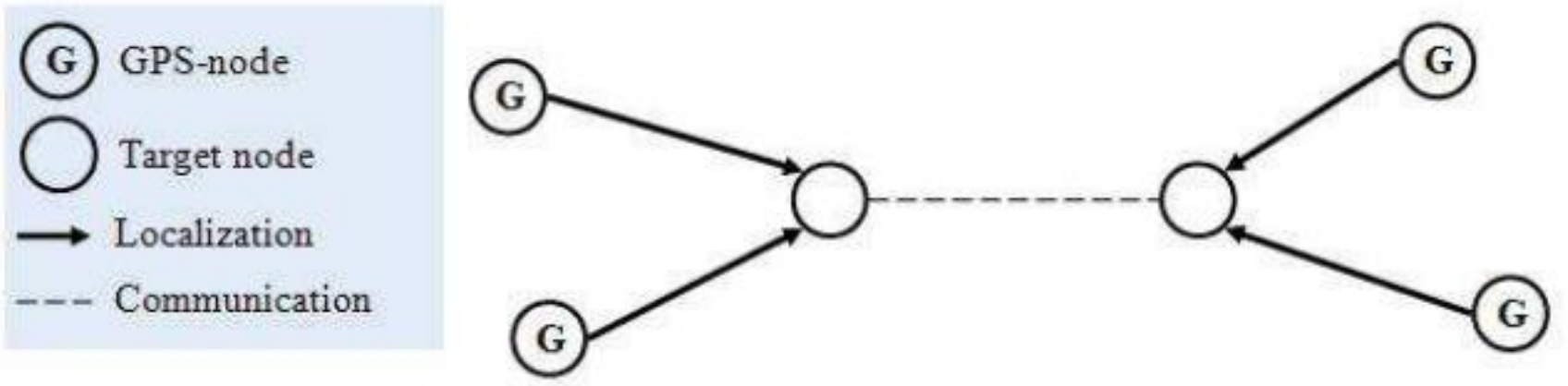
Iterative Localization (multilateration)

- Some nodes are too far away from GPS-node
- Sensor nodes converted to reference node after localization
- On next iteration, these nodes are used to localize rest of unreachable nodes
- Process until all nodes localized



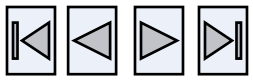
Collaborative Localization (multilateration)

- In large scale sensor network, nodes are randomly placed at start
- All nodes cannot have enough GPS-nodes at initial state
- Two sensor nodes are close to each other, but they have only 2 GPS-node nearby
- Exchanging location information between each other



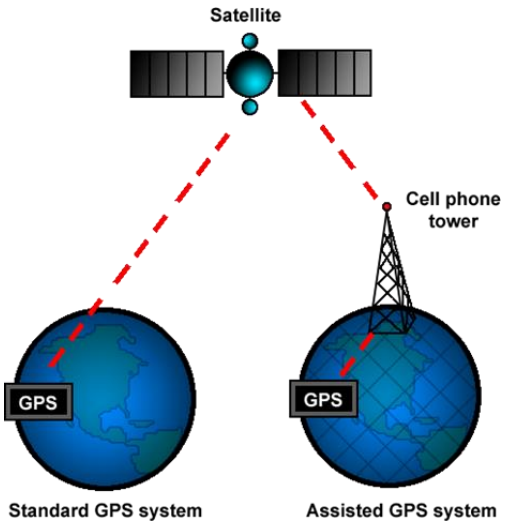
GPS improvements

- Standalone GPS provides first position in approximately 30-40 seconds data rate of the satellite signal is only 50 b/s,
- Downloading orbital information directly from satellites typically takes a long time
- Old GPS algorithm require a lot of time to compute startup information
- Assisted GPS improve the startup performance, or time-to-first-fix (TTFF) of a GPS satellite-based positioning system



Assisted GPS (AGPS)

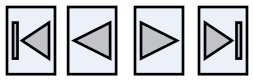
- AGPS servers download the following information from the satellite and store it in the database
  1. Time stamp $\hat{t}_i$: estimate of time at which GPS signal capture initiated
  2. Approximate location of receiver $\hat{\ell}$, typically taken as the base station position
  3. Ephemeris information
  4. Satellite clock corrections
  5. Differential corrections
  6. Navigation data
- AGPS capable device connect to these servers and download these information using Mobile Network



LocationManager (Android)

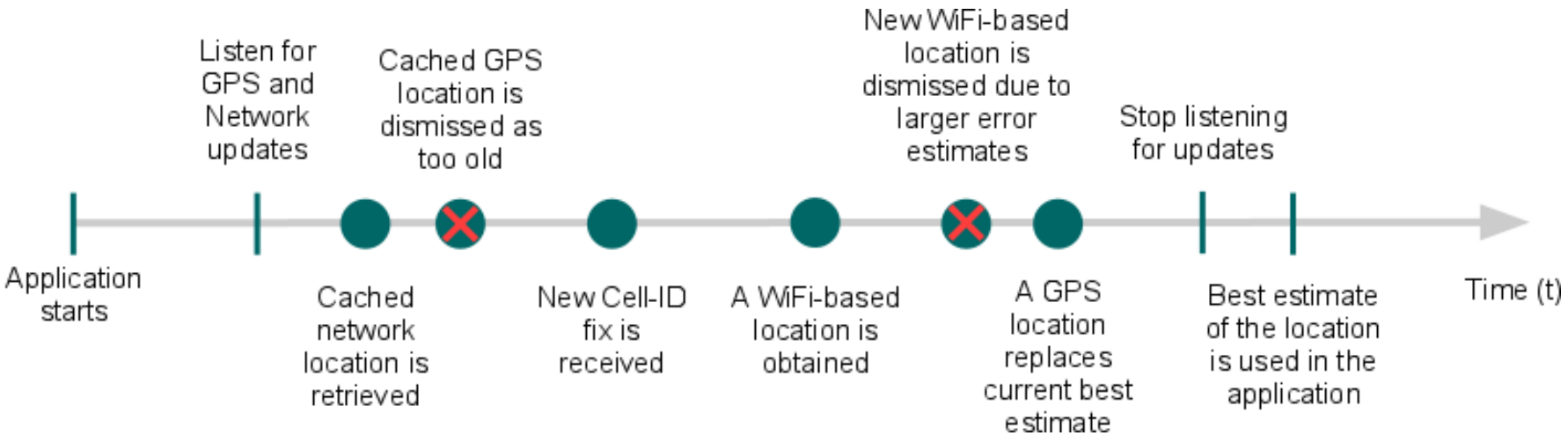
- provides access to the system location services.
- obtain periodic updates of the device's geographical location, or to fire an application-specified Intent when the device enters the proximity of a given geographical location.
- LocationProvider class
  - GPS： GPS Location
  - AGPS： Assisted GPS

```
locationManager= (LocationManager)getSystemService(Context.LOCATION_SERVICE);
List<String> providers= locationManager.getAllProviders();
for(String p:providers){
  // processing
}
```



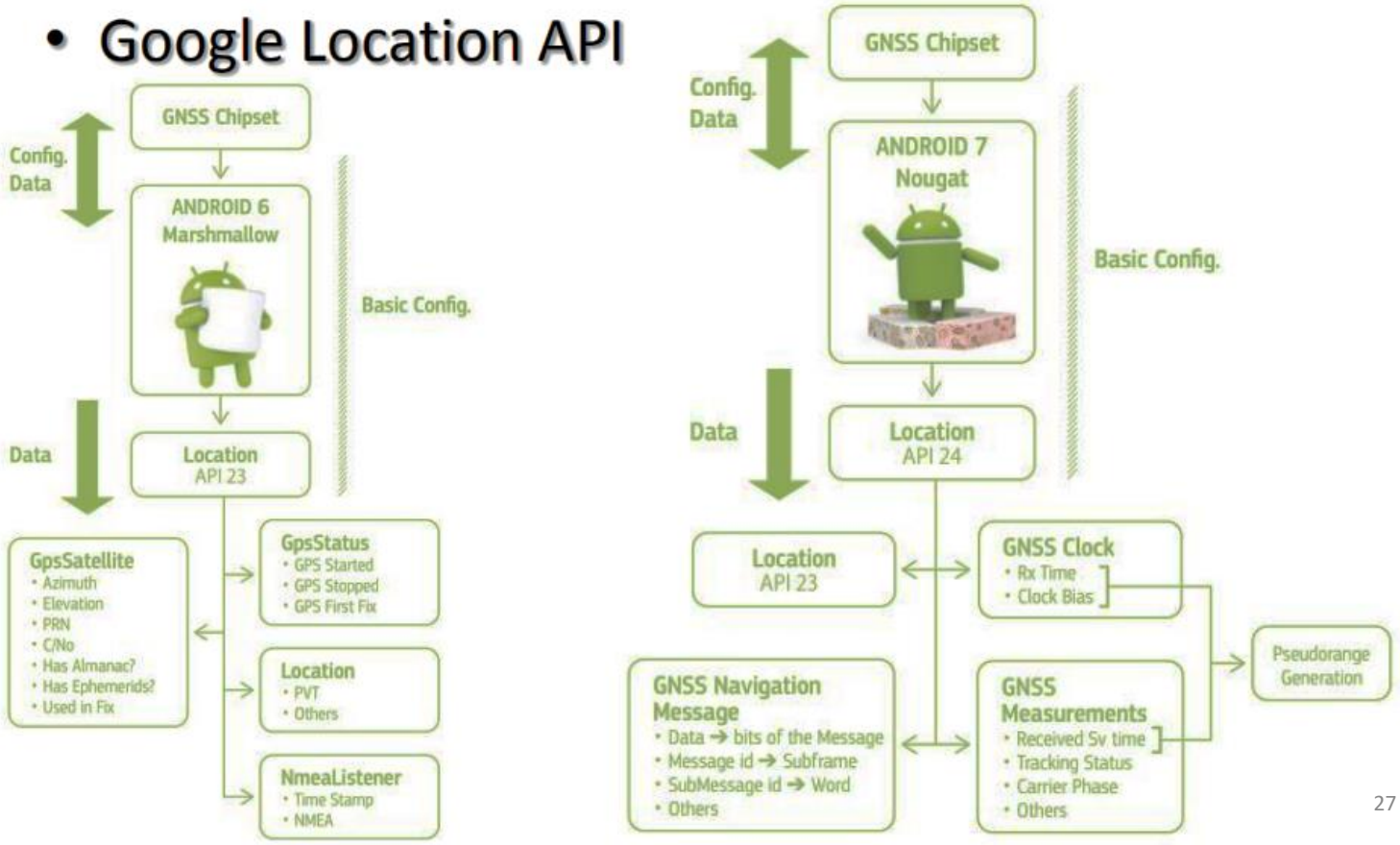
Location (Android)

- With position information coming from GPS, cellular station and WiFi, iOS is also similar
- Android suggests to use a filtering approach to discard outdated estimate and maintain a current best estimate



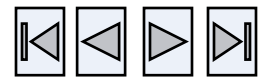
Getting GNSS signal from Android

- From Android 7, developer can obtain raw GNSS measurements for more precise global positioning apps.



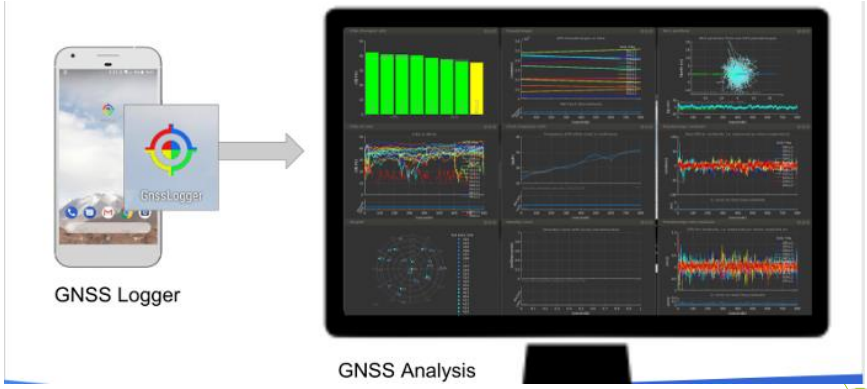
Accessing raw GNSS measurements

Model	Android version	Pseudo-range data	Navigation messages	Accumulated delta range	HW clock	Global systems
Huawei Honor 9	7.0	yes	yes	yes	yes	GPS GLONASS
Samsung S8 (Exynos)[1]	7.0	yes	yes	yes	yes	GPS GLONASS GALILEO BDS
Samsung S8 (QCOM)[2]	7.0	yes	no	no	yes	GPS
Huawei P10	7.0	yes	yes	yes	yes	GPS GLONASS GALILEO BDS
Huawei P10 Lite	7.0	yes	no	no	yes	GPS
Huawei Honor 8	7.0	yes	yes	yes	yes	GPS GLONASS BDS
Huawei Mate 9	7.0	yes	yes	yes	yes	GPS GLONASS BDS
Huawei P9	7.0	yes	yes	yes	yes	GPS GLONASS BDS
Pixel XL	7.0	yes	no	no	yes	GPS
Pixel	7.0	yes	no	no	yes	GPS
Nexus 6P[3]	7.0	yes	no	no	no	GPS
Nexus 5X[3]	7.0	yes	no	no	no	GPS
Nexus 9 (non cellular version)[4]	7.1	yes	yes	yes	yes	GPS GLONASS



Using GNSS in Android

1. Pick some Android devices that support raw GNSS measurements
2. Log raw measurements
3. “Cooked” raw measurements
4. Compute precise location based on least-square based multi-lateration and/or additional cleaning processes.



Google map

- Google Maps APIs allows you to include maps and customize mapping information in your app
- Available to **BOTH *Android*** and ***iOS***

In Android

- Maps can be embedded into an activity as a fragment with XML
- But the Android API is part of Google Play services platform now (closed source assets of Google)

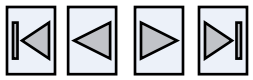
- Thus you must install Google Play Services SDK on AS in order to start

Apply GOOGLE KEY (Android)

Need to apply google key to use it

- https://code.google.com/apis/console
- Register an account, and get a key.
- In AndroidManifest.xml, add the following element as a child of the <application> element, by inserting it just before the closing tag </application>:

```
<meta-data
   android:name="com.google.android.maps.v2.API_KEY"
   android:value="YOUR_API_KEY"/>
```

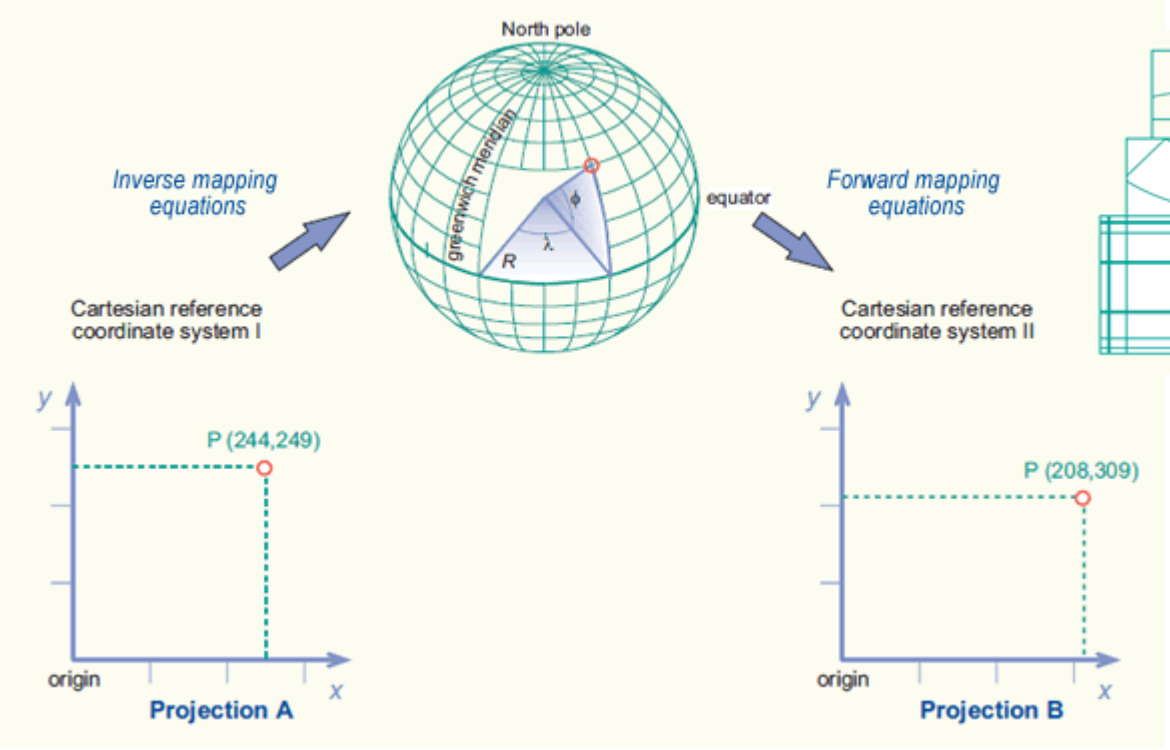


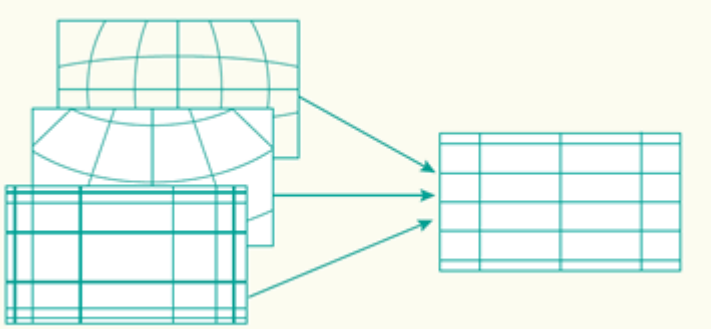
Mercator projection

- Used by Google Map to project the **world's surface** (a sphere) on your **device's screen** (a flat plane) :
  - In the **east** and **west** direction, the map is repeated infinitely as the world **seamlessly wraps** around on itself.
  - In the **north** and **south** direction the map is limited to approximately **85 degrees** north and 85 degrees south.

Coordinate transformations

- Changing map projection between Map and GIS





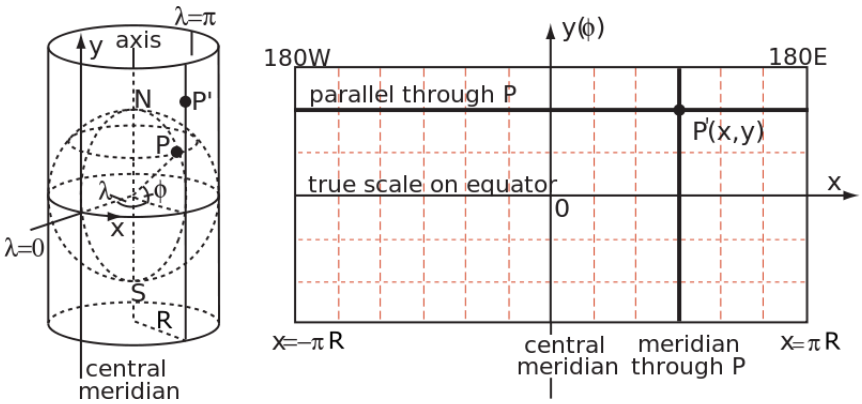
kartoweb.itc.nl

From LatLng to Distance

- Destination point given distance (great circle arc) and bearing from start point:

$$\varphi_2 = asin(\sin \varphi_1 \cdot \cos \delta + \cos \varphi_1 \cdot \sin \delta \cdot \cos \theta)$$
$$\lambda_2 = \lambda_1 + atan2(\sin \theta \cdot \sin \delta \cdot \cos \varphi_1, \cos \delta - \sin \varphi_1 \cdot \sin \varphi_2)$$

where
φ *is latitude,*
λ *is longitude,*
θ *is the bearing (clockwise from north),*
δ *is the angular distance d/R;*
d *being the distance travelled,*
R *the earth's radius (≅6,371km)*



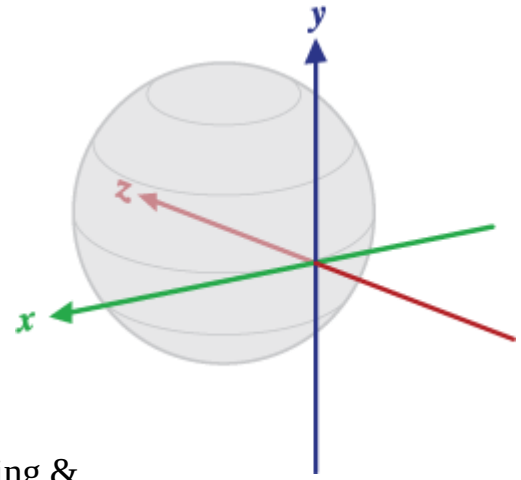
getLastKnownLocation (Android)

- Get final location
- requestLocationUpdate(String provider, long minTime, float minDistance, LocationListener listener)
  - minTime： callback time interval
  - minDistance： callback distance interval

```
locationManager.requestLocationUpdate(locationManager.AGPS_PROV
IDER,60000, 10, new LocationListener(){
  public void onLocationChanged(Location location){
    //update position
  }
}
```

Location Alerts (Android)

- addProximityAlert(double latitude, double longitude, float radius, long expiration, PendingIntent intent)
  - radius：
  - expiration： -1: forever
  - intent： send by broascast
- Create BroadcastReceiver to process the event



Criteria class (Android)

- criteria for selecting a location provider.
- Providers maybe ordered according to accuracy, power usage, ability to report altitude, speed, and bearing, and monetary cost.

```
Private String findProvider(){

  Criteria criteria = new Criteria();

  criteria.setAccuracy(Criteria.ACCURACY_COARSE);
  criteria.setPowerRequirement(Criteria.POWER_LOW);
  criteria.setAlititudeRequired(false);
  criteria.setBearingRequired(false);
  criteria.setSpeedRequired(false);
  criteria.setCostAllowed(true);

  return locationManager.getBestProvider(criteria, true);
}
```

Core Location framework (iOS)

- provides location services for determining a device's geographic location, altitude, orientation based on GPS, Cellular or WiFi

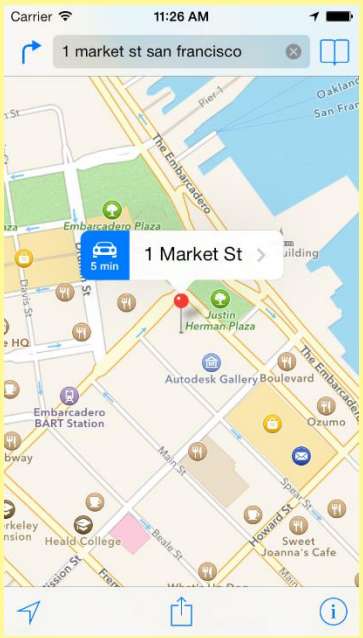
UIRequiredDeviceCapabilities stores strings indicating the features that your app requires.

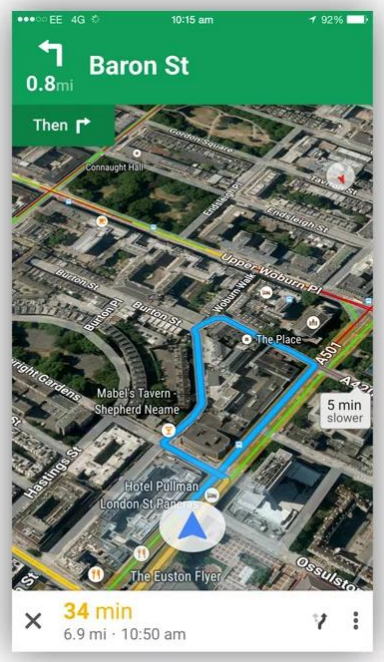
Two strings are relevant to location services:

- **location-services** string if you require location services in general.
- **gps** string if your app requires the accuracy offered only by GPS hardware.

Map Kit framework (iOS)

- Maps in iOS are provided by the **Map Kit** framework, which supports both the display and annotation of maps similar to those found in the Maps app.
- Have to turn on the Maps capability in the Xcode project.
- Apple Map's accuracy is comparable to Google Map while turn-by-turn navigation UI (especially in 3D maps) is clearer





Reference

1. Global Navigation Satellite System Fundamentals
   https://www.ieee.li/pdf/viewgraphs/gnss_fundamentals.pdf
   http://www.unoosa.org/pdf/icg/2013/Ed_GNSS_eBook.pdf

2. Raw GNSS Measurements | Android Developers
   https://developer.android.com/guide/topics/sensors/gnss
   https://developer.android.com/reference/android/location/GnssMeasurement

3. Longitude and Latitude points calculations
   https://www.movable-type.co.uk/scripts/latlong.html#dest-point