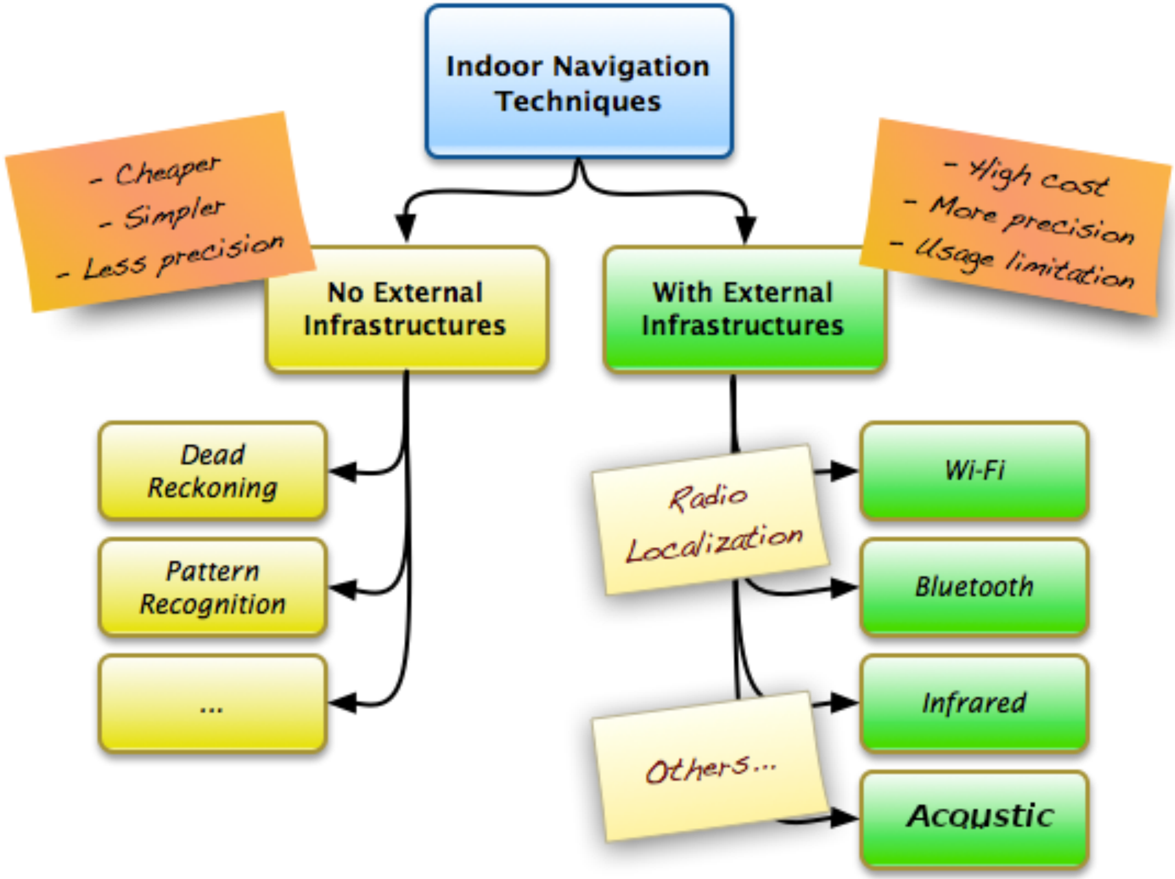
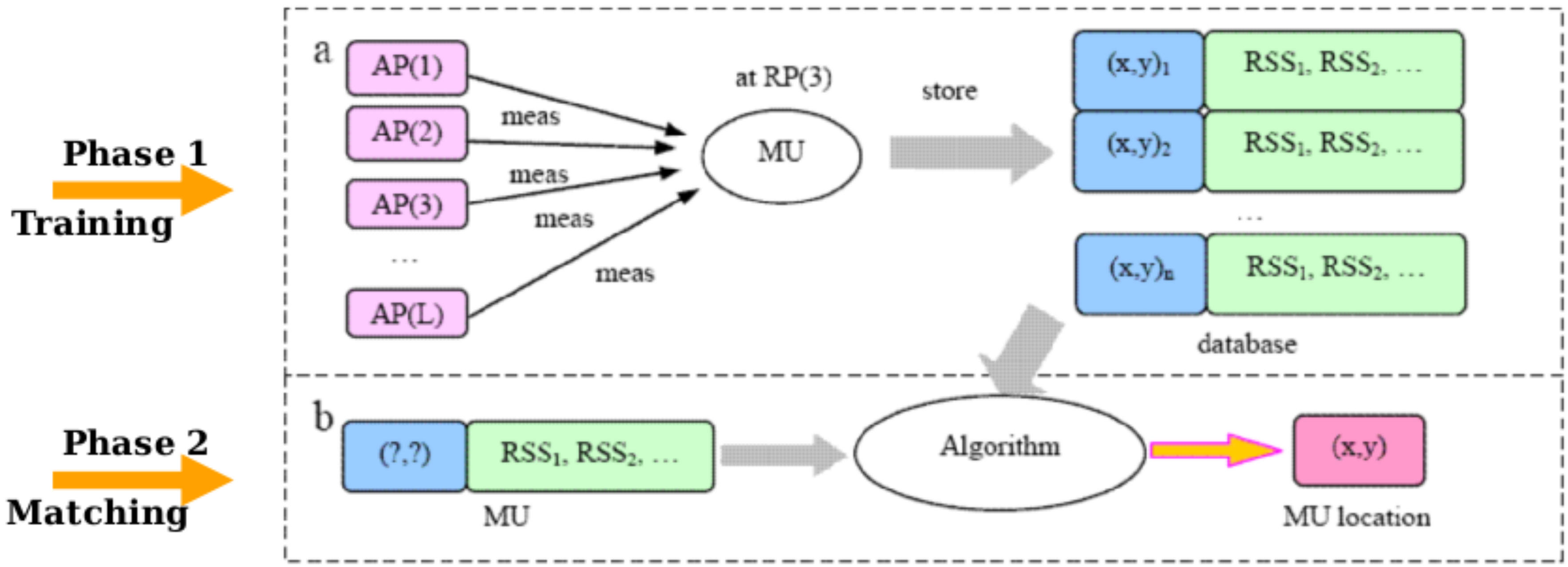
Lesson 3

17 November 2014

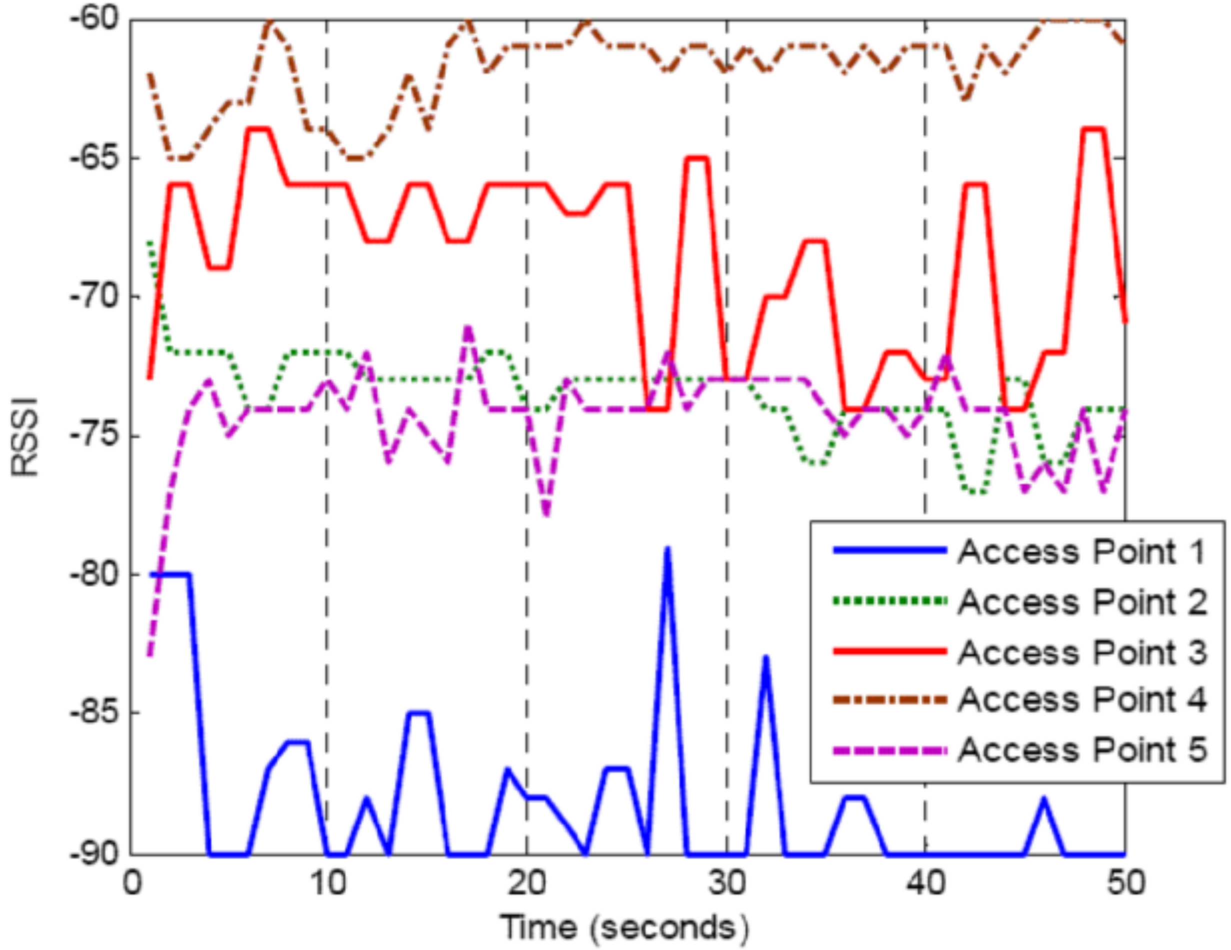
Indoor Localization



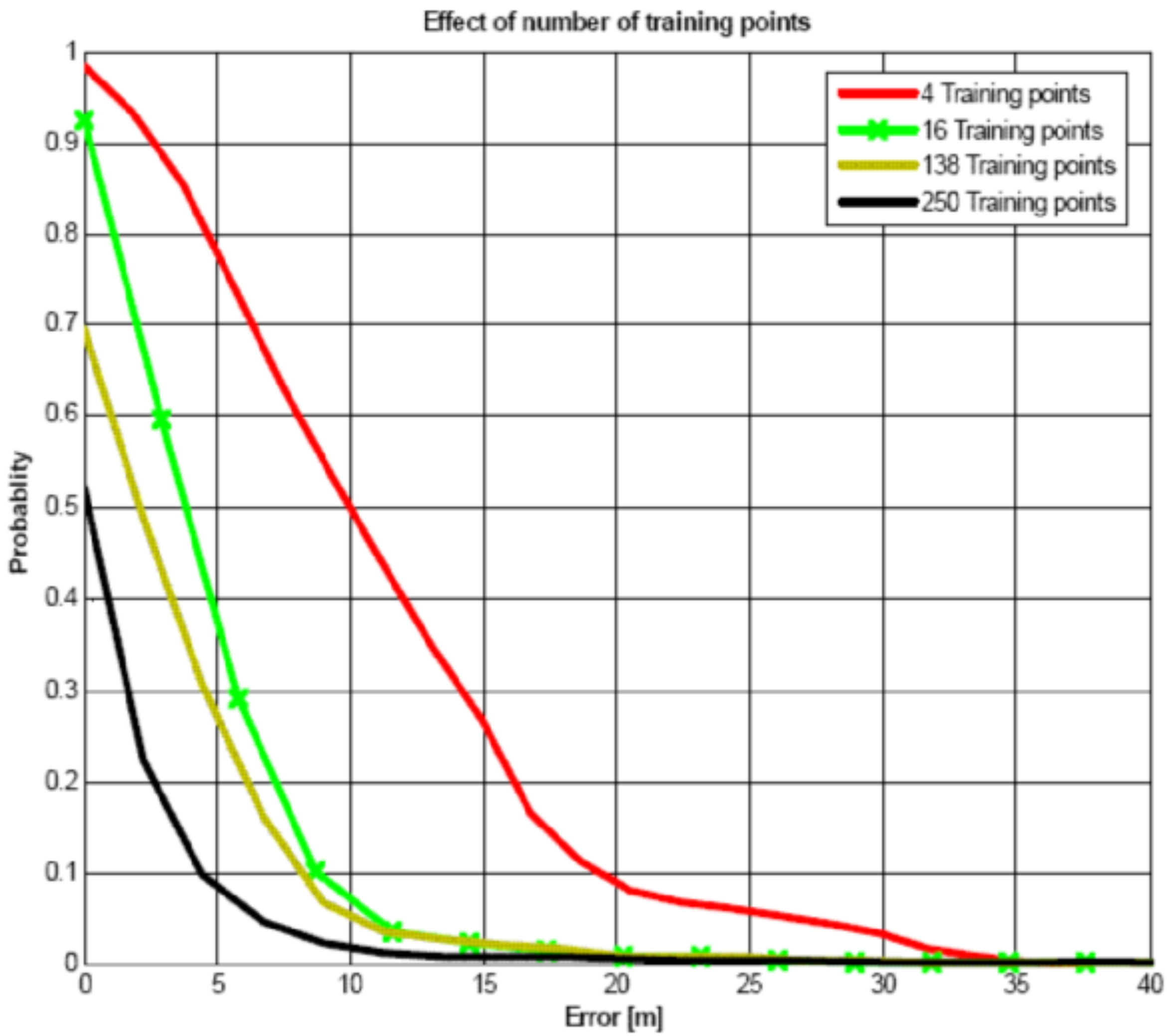
WIFI Fingerprinting



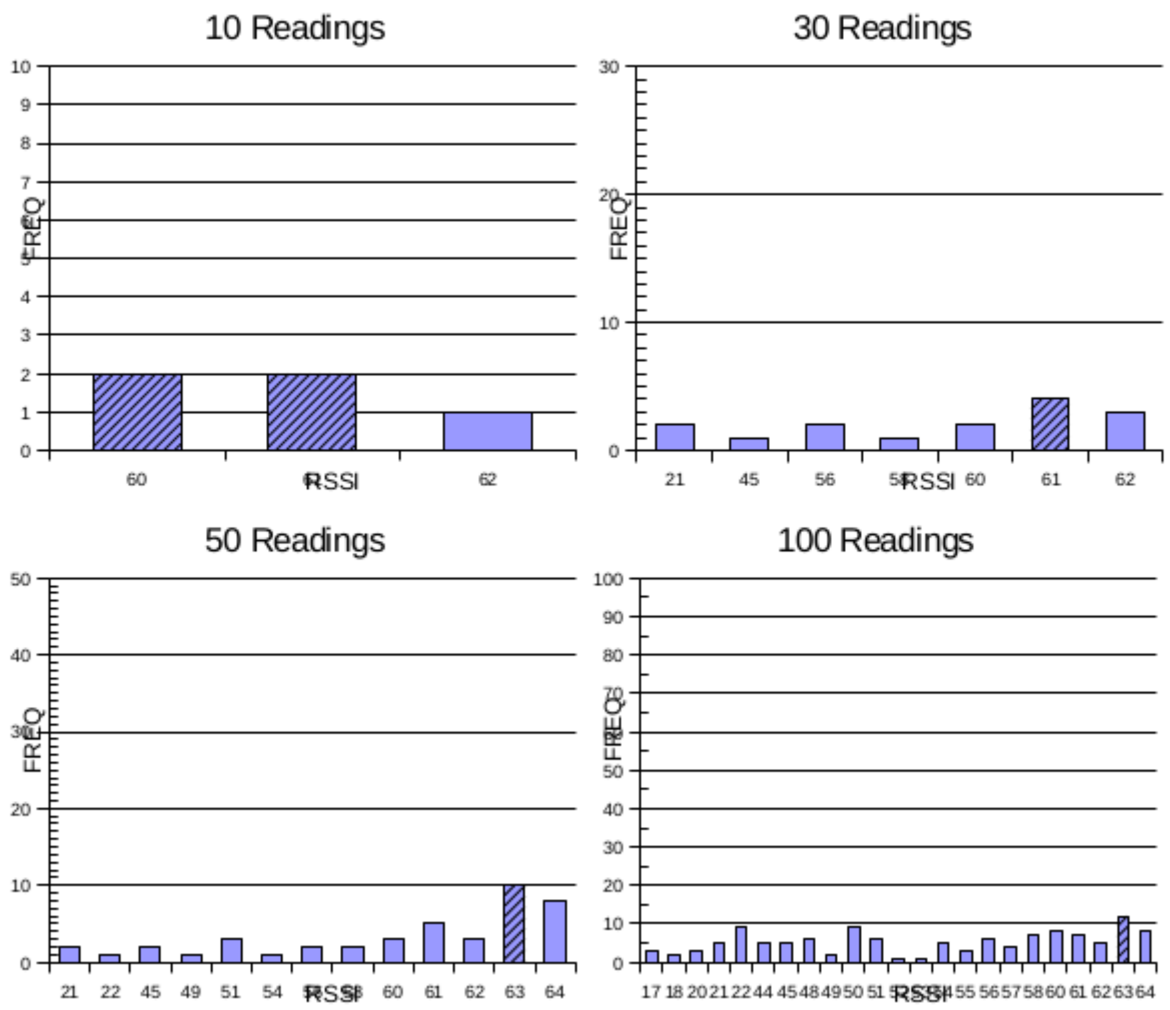
Signal instability



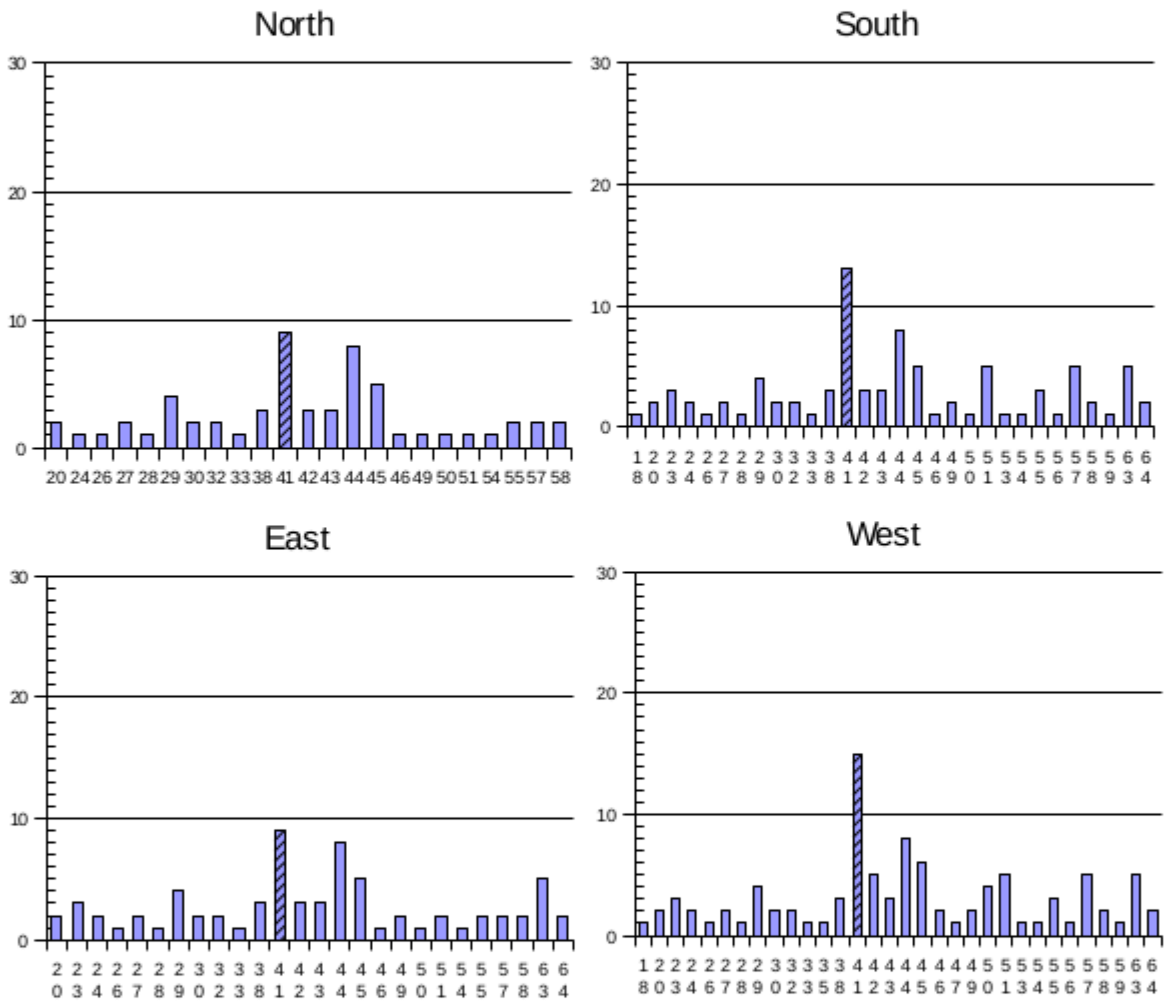
Granularity



Number of samples



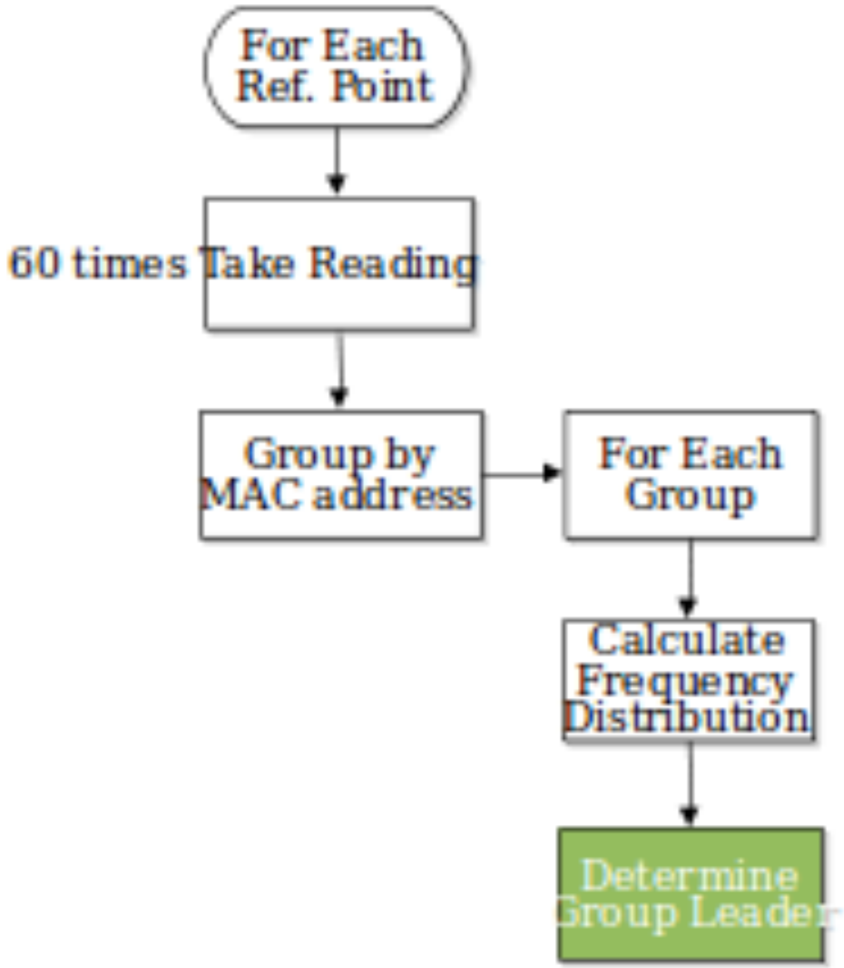
User orientation



Algorithms

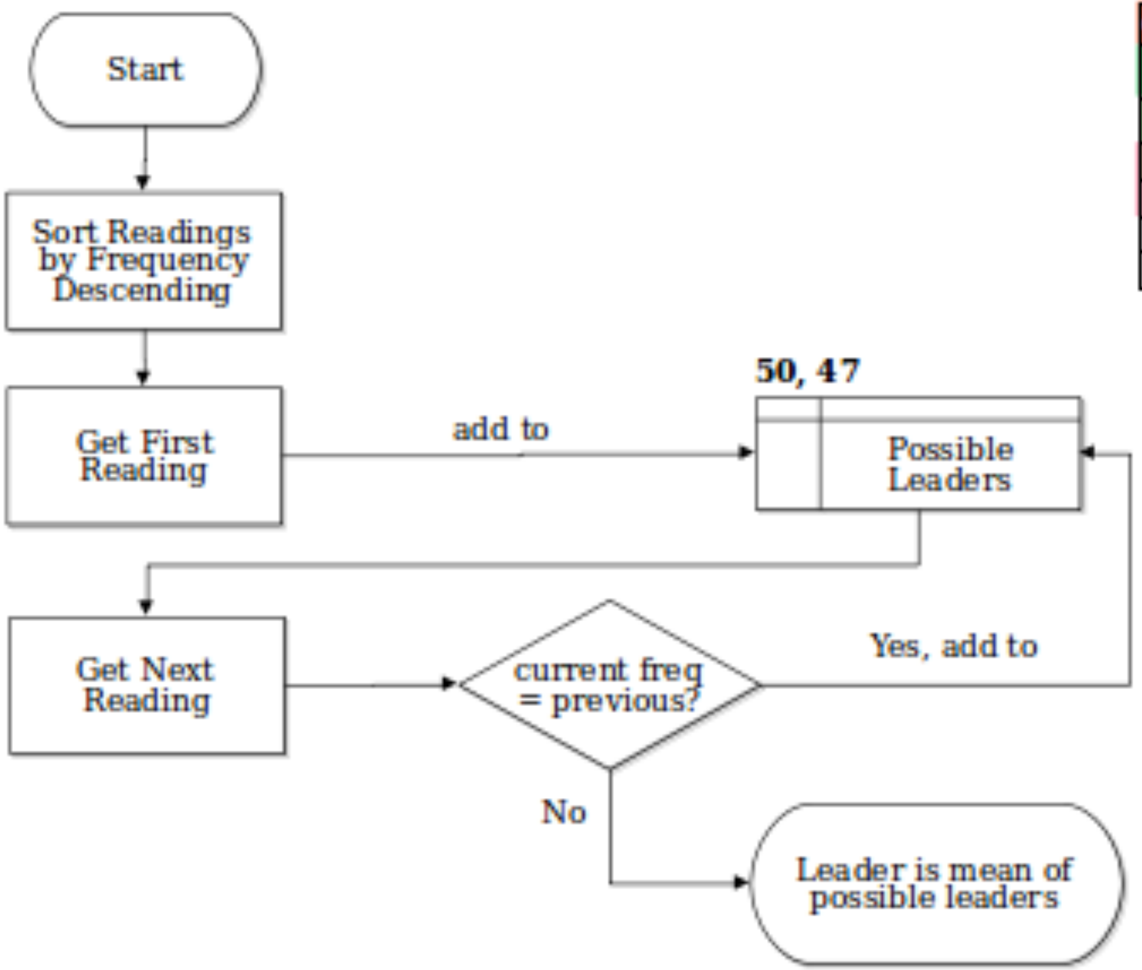
- **Training**
  - Fingerprinting algorithm
- **Matching**
  - Range based algorithm
  - Euclidean distance based algorithm

Fingerprinting algorithm



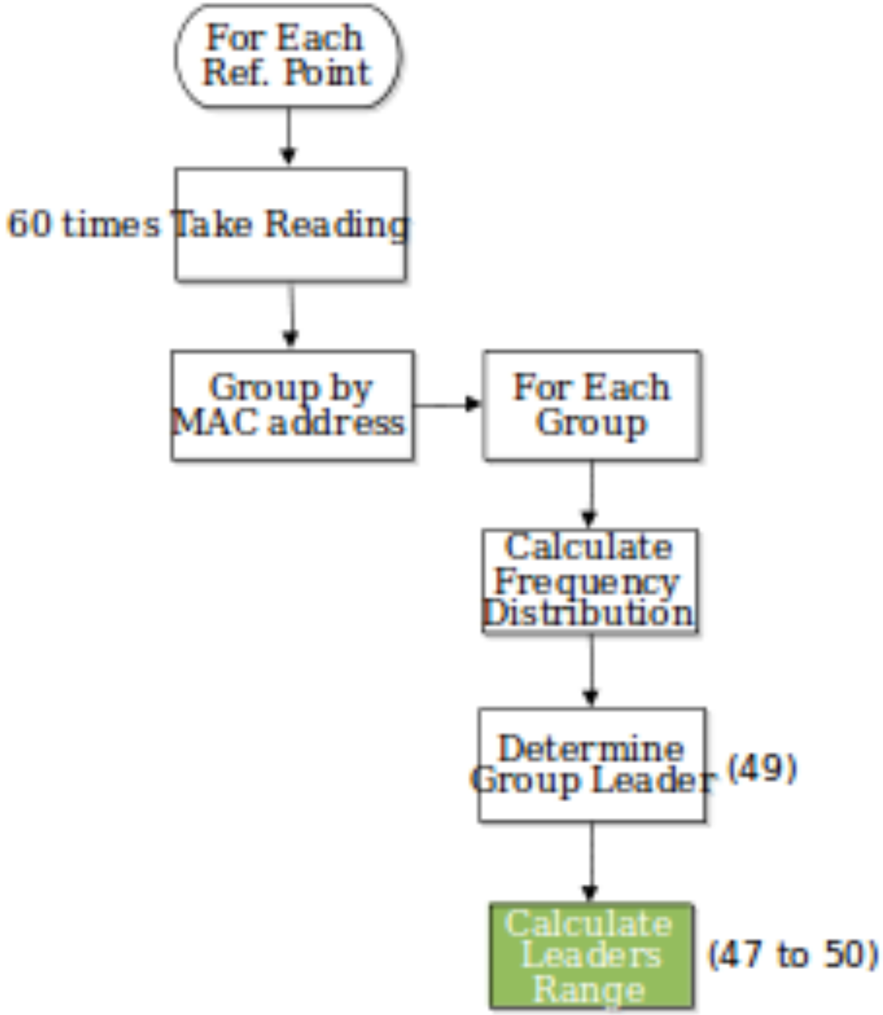
Determine Group Leader

Group A7C8	
RSSI	Frequency
50	20
47	20
45	11
36	6
25	3

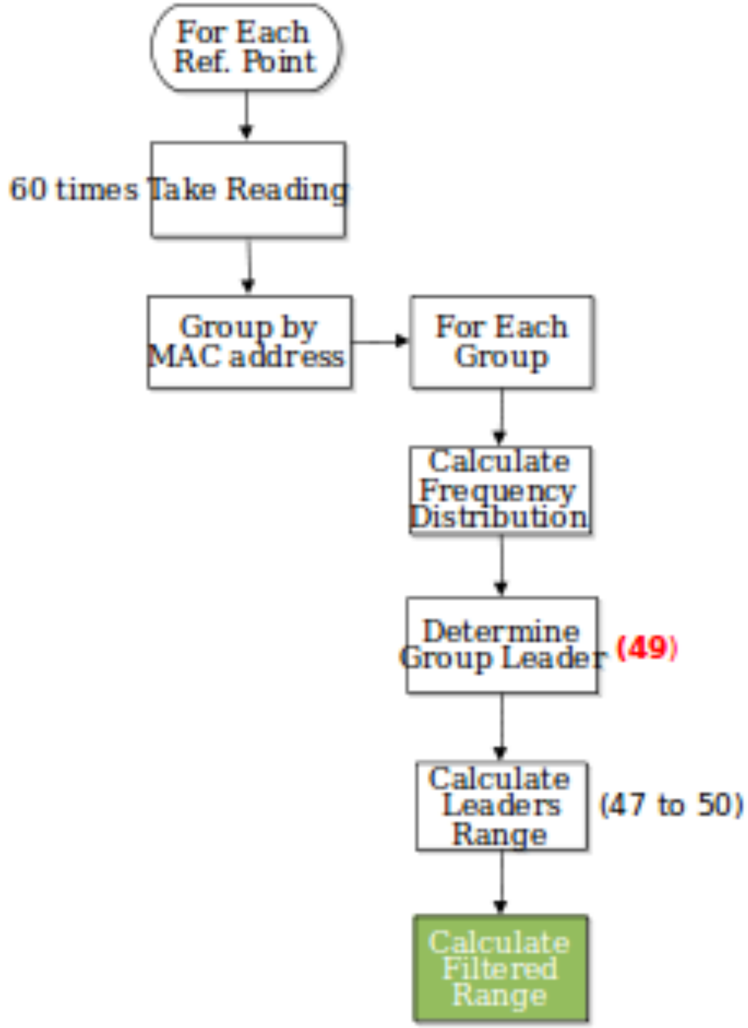


$$\frac{(50+47)}{2} = 48.5 \approx 49$$

Fingerprinting algorithm



Fingerprinting algorithm



Calculate filtered range

Group	
RSSI	Frequency
50	20
47	20
45	11
36	6
25	3

Threshold
If we take it as 50%

Minimum frequency
Freq(Leader) x Threshold
= Freq(49) x 50% = 20 x 50% = 10 readings

∴Readings with frequency < 10 will not be considered.

Calculate filtered range

Group	
RSSI	Frequency
50	20
47	20
45	11
36	6
25	3

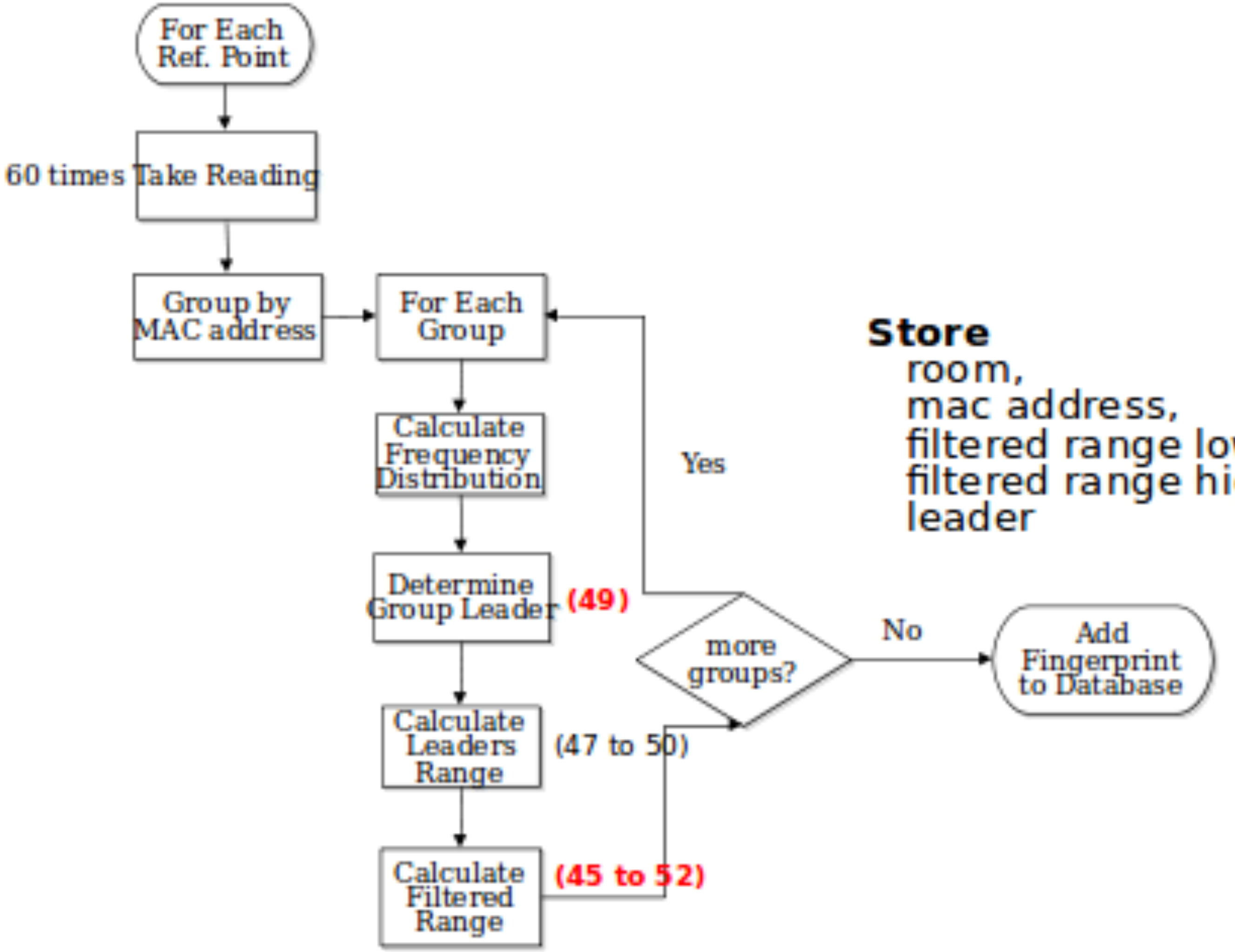
Filtered Standard Deviation
The SD of all readings above the minimum frequency (10).

Filtered SD = 1.98 instead of 6.18

Filtered Range
The leaders range ± filtered SD.

Why filtered standard deviation?

Fingerprinting algorithm



Store
room,
mac address,
filtered range low,
filtered range high,
leader

Matching algorithm

Two algorithms

- Range based algorithm
- Euclidean distance based algorithm

Range based

Room ID	Ref Point	MAC	Low Range	High Range	Stored
1	1	A7C8	35	42	38
1	2	A7C8	35	41	37
2	1	A7C8	34	38	35
2	2	A936	20	22	21

Given the observed signal vector:

MAC: A7C8, Signal: 37

Room 1 is more likely to be the correct location

Euclidean distance based

$$distance = \sqrt{\sum_{i=1}^{n} (o_i - s_i)^2}$$

n represents the number of access points recorded for each reference point, **o** is the observed signal value and
s is a stored signal value in the database.

For simplicity of the example, let **n = 1**, which means that only 1 access point is recorded for each reference point.

$$simplified\ distance = \sqrt{(o - s)^2}$$

Euclidean distance based

Room ID	Ref Point	MAC	Low Range	High Range	Stored
1	1	A7C8	35	42	38
1	2	A7C8	35	41	37
2	1	A7C8	34	38	35
2	2	A936	20	22	21

$$\sqrt{(o-s)^2}$$

$$\sqrt{(37-38)^2}=1$$

$$\sqrt{(37-37)^2}=0$$

$$\sqrt{(37-35)^2}=2$$

Given the observed signal vector:

MAC: A7C8, Signal: 37

Room 1 with reference point 2 is more likely to be the correct location, because it has the minimal Euclidean distance from the observed signal vector.

Results

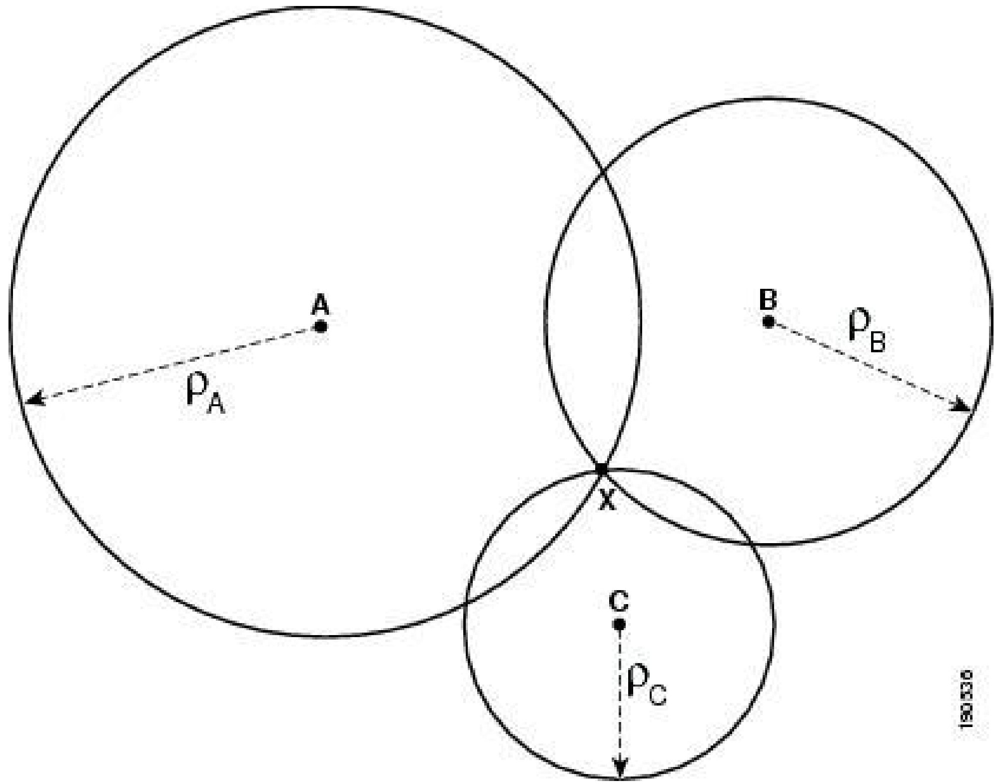
The system is able to detect user's location correctly within 3-4 meters accuracy

- 70% of the time, using Euclidean based
- 60% of the time, using range based

Basic notions of trilateration

Lateration is the most common method for deriving the location of a wireless device.

- For e.g., GPS use trilateration
- Two steps are involved:
  - The distance (range) to anchors is calculated
  - The location of the nodes is estimated based on the distance



Main algorithms for trilateration positioning

- The most used approaches for trilateration positioning:
  - Non-Linear Least Square (NLS)
  - Bayes Filters as Extended Kalman Filters (EKF) methods.
- There are three unknowns:
  - Coordinate of the receiver
- NLS is largely used to calculate these unknowns
- EKF for tracking and sensor fusion

Positioning: the problem of data fitting

The problem of positioning is one of data fitting.

- **Given a set of input measurements {ρi }, what parameters {x̂, ŷ, ẑ } provide a best fit?**

- Regression analysis estimates the parameters.
- The best fit must provide a solution which is close (in some sense) to the true location p = {x, y, z }

Non-linear regression

- Current position estimate **p = {x^, y^, z^ }**
- Associated static source s_i = {x_i , y_i , z_i}

ρ_i = || s_i - p || + v = √ { (x^-x_i)^2 + (y^-y_i)^2 + (z^-z_i)^2 +v }

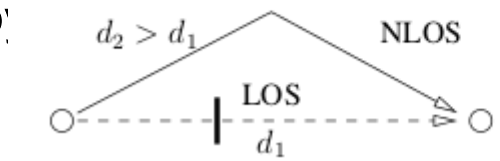
- This equation is a non-linear function of the parameters
  - Non Linear regression
- No closed-form expression for the best-fitting parameters
  - **Numerical optimization**

$$\hat{p} = \arg\min_{\hat{p}} \sigma^2 = \arg\min_{\hat{p}} \frac{1}{N} \sum_{i=1}^{N} e_i^2 .$$

When least-square (LS) works

It gives a position that agrees with most of the data.

- Least-square methods are appropriate when:
  - The noise is Gaussian with equal variance (v).
  - The input data are reliable.
  - Only one or two datums are multipathed.
    - Covered b



Linearization: LLS algorithm

- The Linear Least Square (LLS) approach linearizes the NLS problem
  - by introducing a constraint in the formulation
- It obtains a closed form expression of the estimated location.

Example

Start with **N > 1** equations to estimate position $(\hat{x}, \hat{y})$:

$$\begin{cases} (x_1 - \hat{x})^2 + (y_1 - \hat{y})^2 = d_1^2 \\ (x_2 - \hat{x})^2 + (y_2 - \hat{y})^2 = d_2^2 \\ \vdots \\ (x_N - \hat{x})^2 + (y_N - \hat{y})^2 = d_N^2 \end{cases}$$

and subtracting the constraint :

$$\frac{1}{N}\sum_{i=1}^{N}[(x_i - \hat{x})^2 + (y_i - \hat{y})^2] = \frac{1}{N}\sum_{i=1}^{N} d_i^2$$

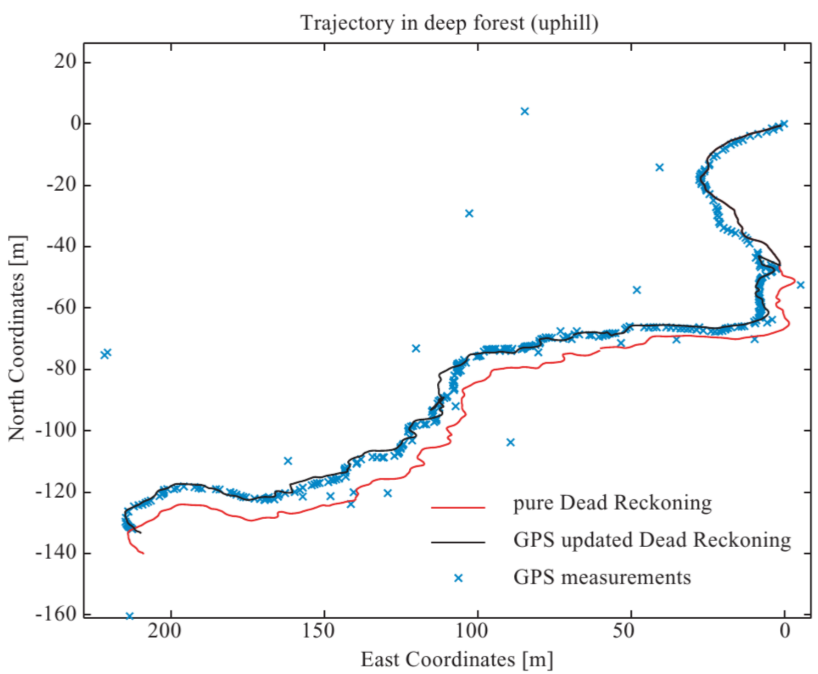
from both sides of each equation, the above can be rewritten as **A p = b**.

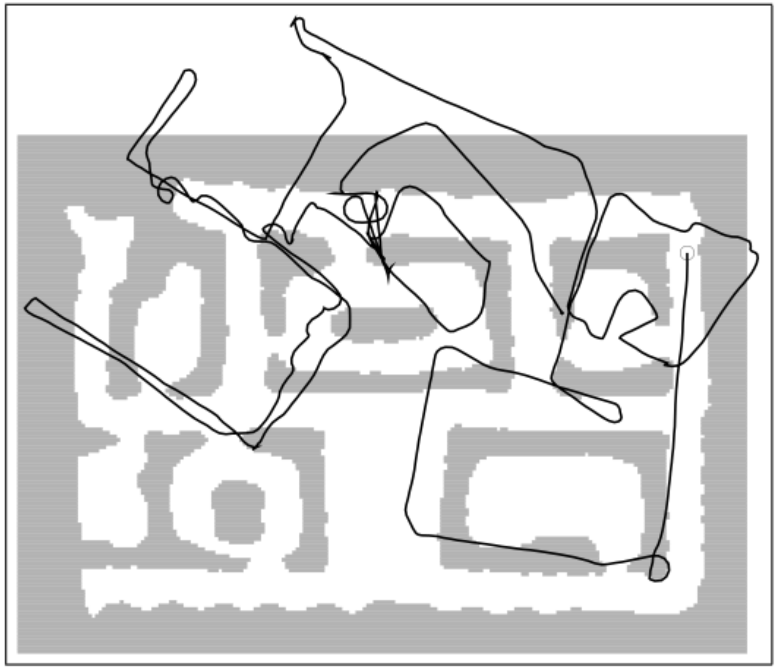
Solution is : $\mathbf{p = (A^T A)^{-1} A^T b}$

Inertial Tracking

- Use electronic accelerometers, compasses.
- Sense movement and direction in 2D and 3D
- Dead-reckoning → relative positions
  - (w.r.t. given fixed point)
- Position error grows with time and distance in the absence of:
  - Position fixes
  - Fusion with other sensors (EKF)
  - Context information (e.g., map)

Good and bad results with DR





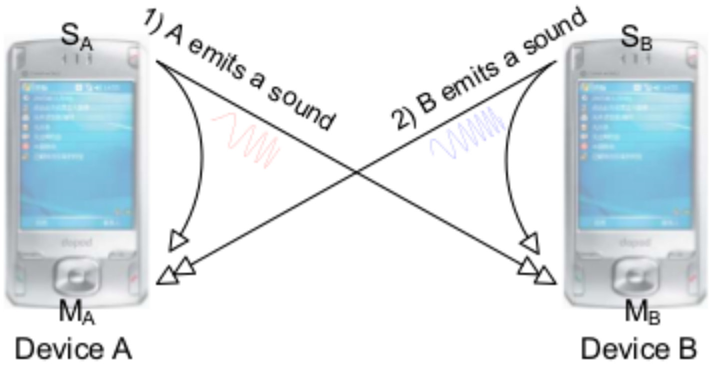
- http://roodin.crs4.it/
  - Implements dead-reckoning in android using pedometer
  - Code Libraries are available
  - Good starting point (2 publications + MS thesis reference (MS is in italian))

Acoustic Localization

- Sound propagation speed is much slower than radio.
- ≈ 343.2 m/s (air) versus ≈ 300 m/μs, 10^6 times slower than light!!
- No need for high clock rates, but :
  - Clock drift 1 ms error in TOA estimation will translate to more than 30 centimeters error in the ranging result
  - Android system inaccurate to guarantee exact time of received packet and transmitted packet

BeepBeep

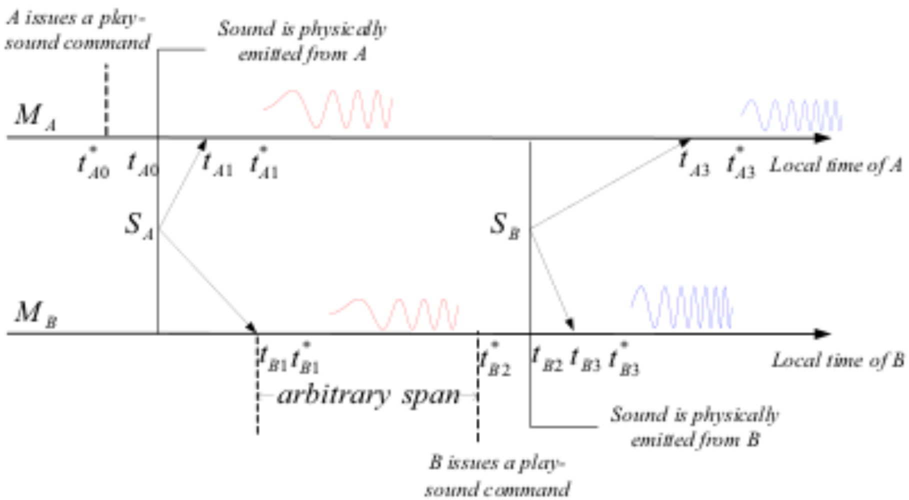
BeepBeep: A High Accuracy Acoustic Ranging System using COTS Mobile Devices



$$
\begin{aligned}
d_{A,A} &= c \cdot (t_{A1} - t_{A0}) \\
d_{A,B} &= c \cdot (t_{B1} - t_{A0}) \\
d_{B,A} &= c \cdot (t_{A3} - t_{B2}) \\
d_{B,B} &= c \cdot (t_{B3} - t_{B2})
\end{aligned}
$$

$$
\begin{aligned}
D &= \frac{1}{2} \cdot (d_{A,B} + d_{B,A}) \\
&= \frac{c}{2} \cdot ((t_{B1} - t_{A0}) + (t_{A3} - t_{B2})) \\
&= \frac{c}{2} \cdot (t_{B1} - t_{B2} + t_{B3} - t_{B3} + t_{A3} - t_{A0} + t_{A1} - t_{A1}) \\
&= \frac{c}{2} \cdot ((t_{A3} - t_{A1}) - (t_{B3} - t_{B1}) + \\
&\quad (t_{B3} - t_{B2}) + (t_{A1} - t_{A0})) \\
&= \frac{c}{2} \cdot ((t_{A3} - t_{A1}) - (t_{B3} - t_{B1})) + d_{B,B} + d_{A,A}
\end{aligned}
$$

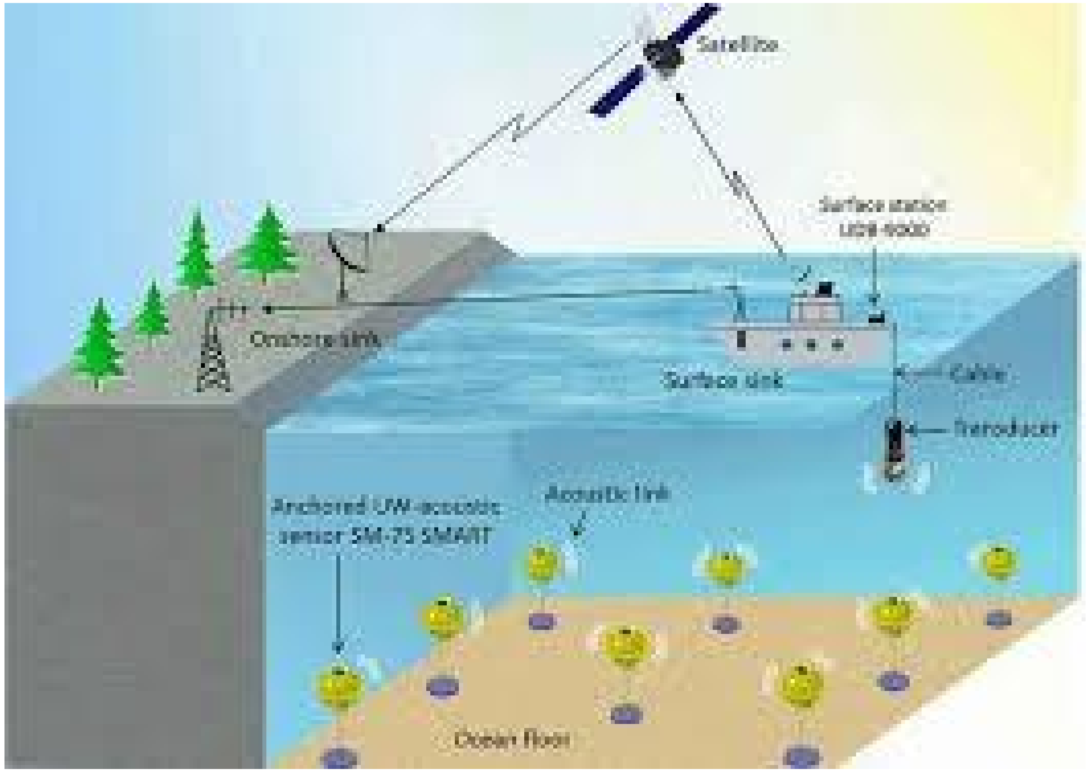
Details



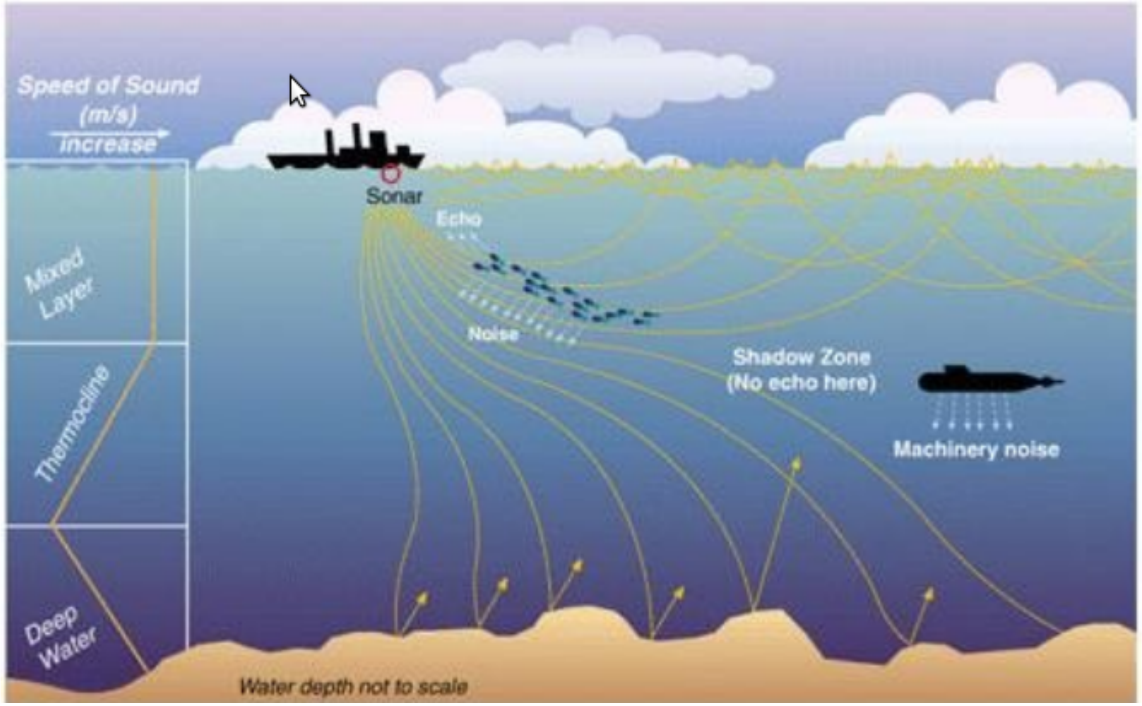
- Off the shield implementation (only software)
- No time-synchronization
- Works directly on received signal,offline computation
- Accuracy on several cm
- Easy to be implemented on Android :)

From land to water ...

- Underwater Acoustic Wireless Sensor Networks (UAWSN)



UAWSNs Communication Issues



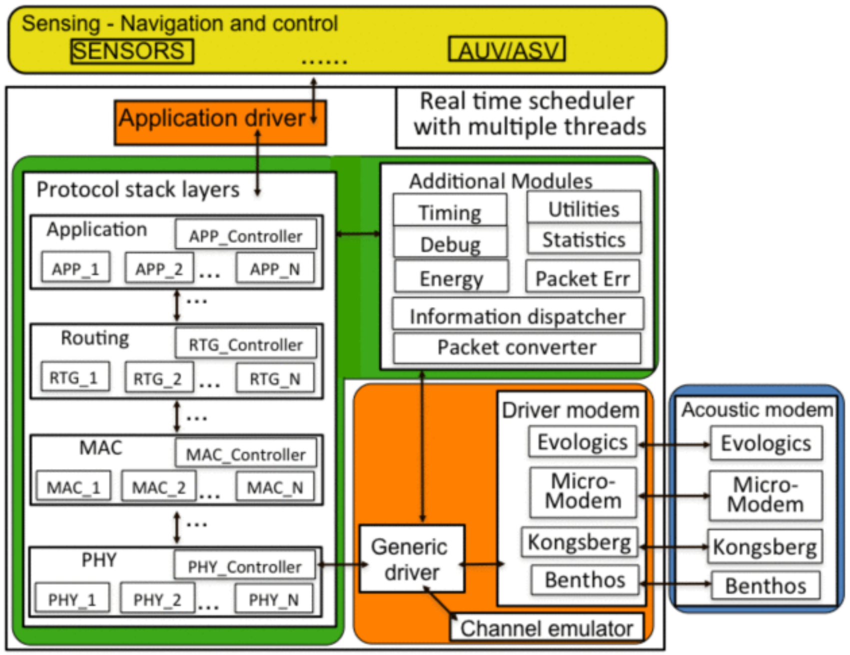
Propagation Delay (1500 m/s)
Attenuation (depends on frequency)
High Variablity (asymmetric links)
Low Data Rate (1 kb/s)
Noise

No mathematical UW channel mode

Protocols designed for terrestrial sensor networks are unsuitable.

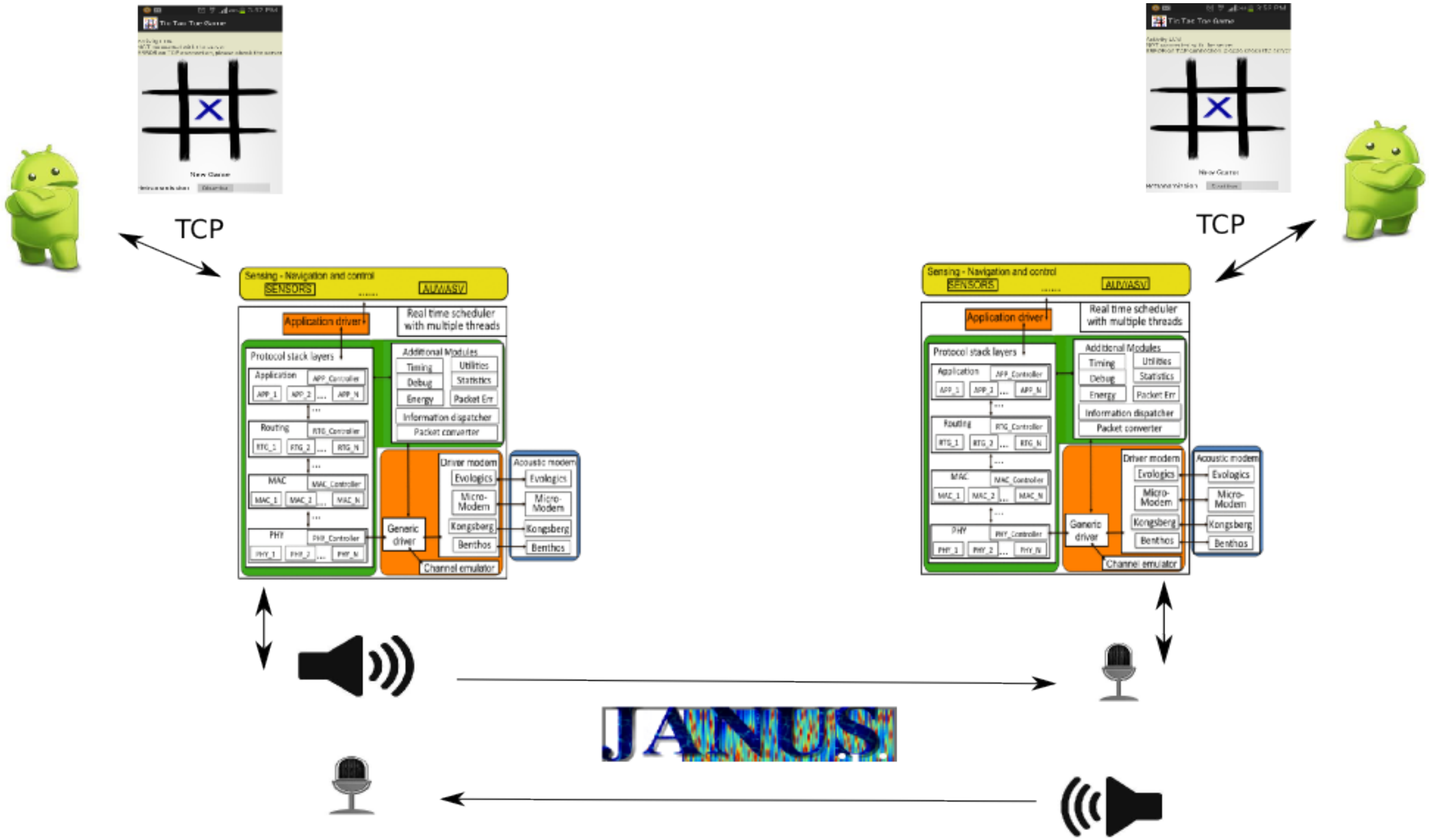
http://reti.dsi.uniroma1.it/UWSN_Group/

SUNSET Framework

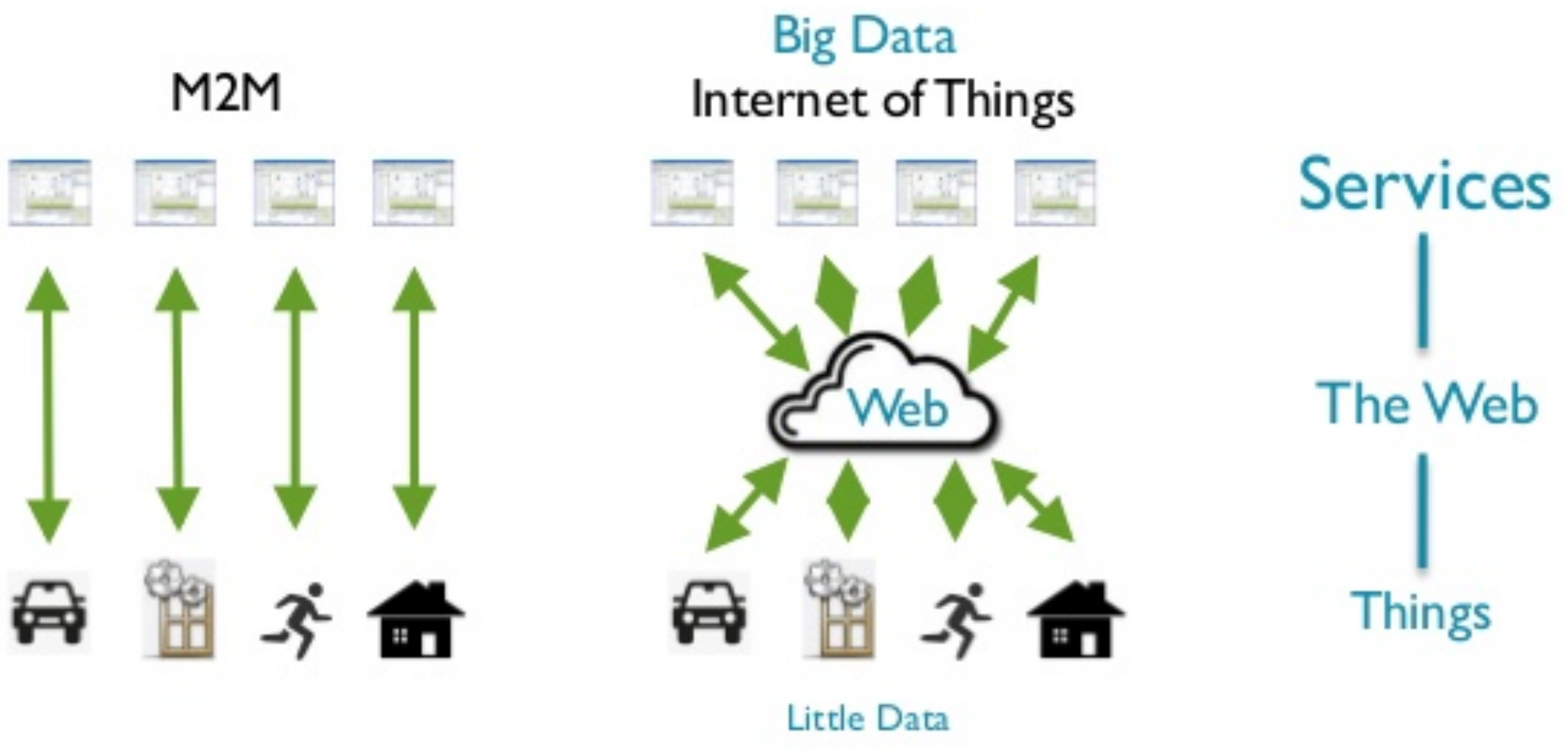


- MAC & Routing protocol design, simulate and validate at sea
- Path planning algorithms for AUV
- Mathematical modelling of UWSN
- Acoustic transmission & Signal Processing
- Cooperation and communication Modems + Sensors + AUV
- Underwater IoT

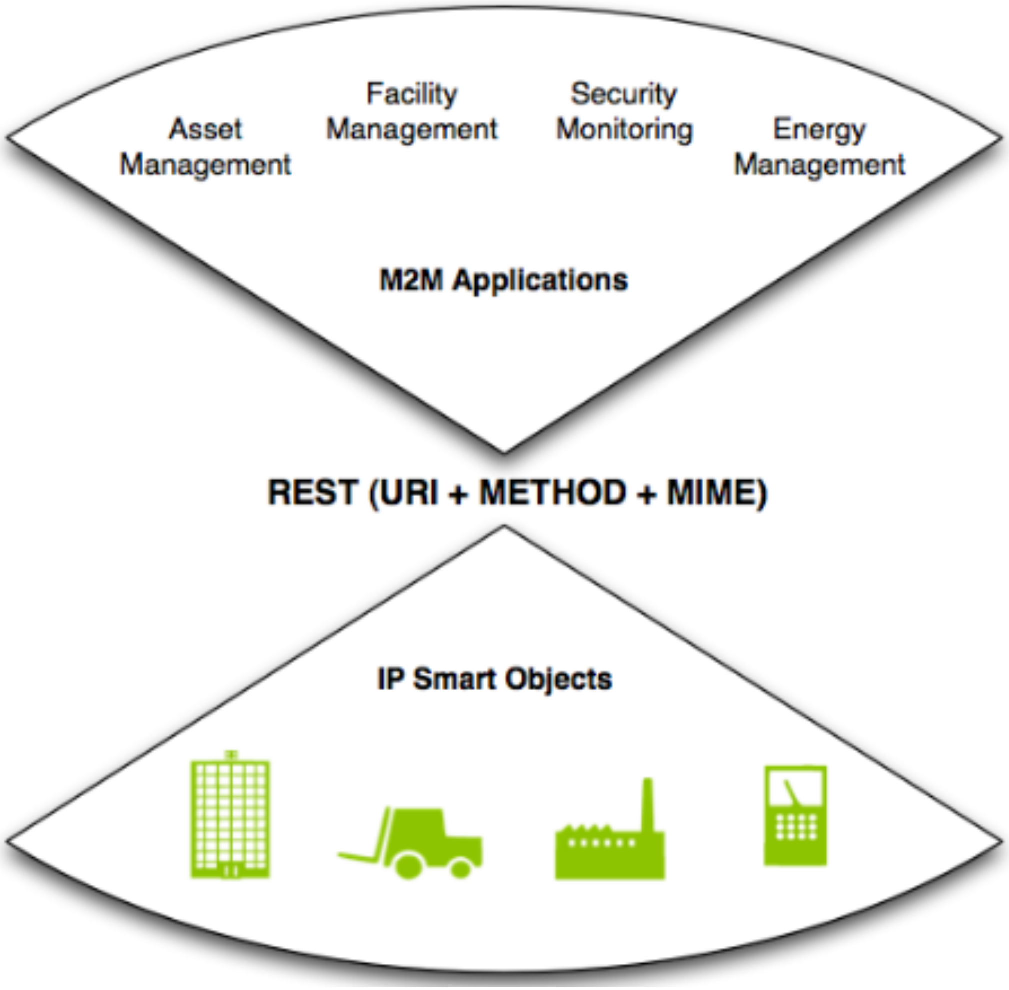
Underwater Tic Tac Toe :)



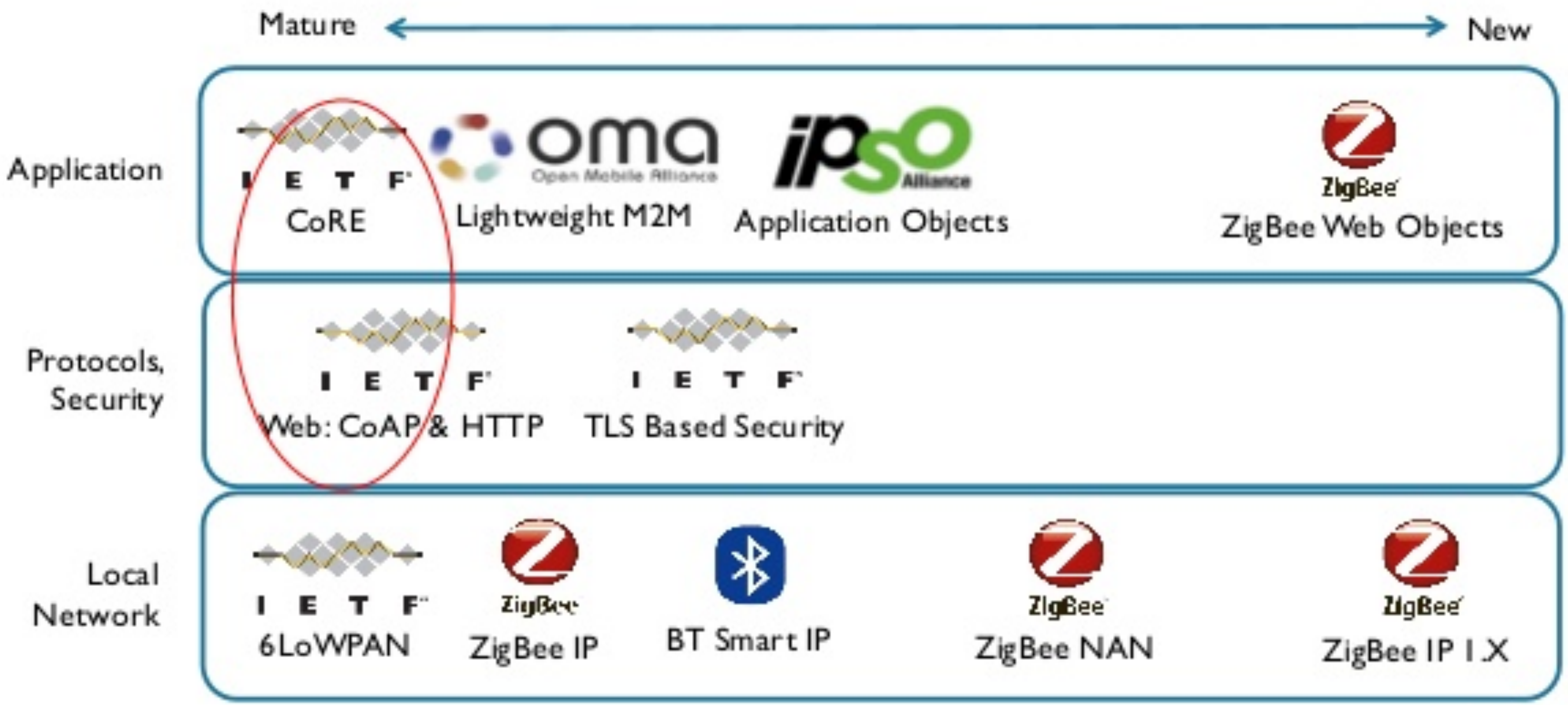
Evolution of M2M



The Web of Things



IoT Standards



CoRE Requirements