WSN localization with Senseless

Peter De Cauwer
Tim Van Overtveldt

artesis
hogeschool antwerpen

Team

- Students:
  - Peter De Cauwer
  - Tim Van Overtveldt
- Promotors:
  - Jeroen Doggen
  - Jerry Bracke
  - Maarten Weyn

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Contributions

- Expand Senseless framework to incorporate localization with RSSI
  - Compare different algorithms
  - Test the influence of the orientation of a node
- Interface this framework to Scala

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Wireless Sensor Network

- A **wireless sensor network** (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants.

Motivation

- What?
  - To determine the physical coordinates of a group of sensor nodes in a wireless sensor network (WSN)
  - Due to application context and massive scale, use of GPS is unrealistic, therefore, sensors need to self-organize a coordinate system
- Why?
  - To report data that is geographically meaningful
  - Services such as routing rely on location information; geographic routing protocols; context-based routing protocols, location-aware services

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Applications

- Environmental monitoring (air, water, soil chemistry, surveillance)
  - REDWOOD
- Home automation (smart home)
- Inventory tracking (in warehouses, laboratories)

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

RTLS - Definitions

- Anchor Nodes:
  - Nodes that know their coordinates a priori
  - By use of GPS or manual placement
  - For 2D three and 3D four anchor nodes are needed
- Goal: to position a blind node by using pair-wise measurements with the anchor nodes.
  - Anchor-based

RTLS - 2 phases

1. Determine the distances between blind nodes and anchor nodes.

2. Derive the position of each node from its anchor distances.

RTLS - Phase 1

- Range-less
  - Connectivity
  - Hop Count
    - Sum-Dist
    - Dv-Hop
    - Euclidean
- Range-based
  - Ranging methods

RTLS - Phase 1 - Range-based

- TOA
- TDOA
- RTT
- AOA
- RSS

RTLS - Phase 1 - Range-based

- TOA
- TDOA
- RTT
- AOA
- RSS

Phase 1 – Range-based (RSS)

- Radio signals attenuate with distance
- Available in most radios
  - No extra cost
- Poor accuracy
  - Difficult to model

RSS - Errors

- Environmental errors
  - Multipath
  - Shading
  - Interference
    - Gaussian noise

RSS - Errors

- Device errors
  - Transmitter variability
  - Receiver variability
  - Antenna orientation

RSS - Model

- Different models
  - log-distance path loss model
- $RSS(d) = P_T - P(d0) – 10\ n \log(d / d0) + Xo$
  - P_T Transmitted power [dBm]
  - RSS Received Signal Strength[dBm]
  - P(d0) Path loss in dBm at a distance of d0
  - n Path loss exponent
  - d Distance between two nodes[m]
  - d(0) Reference distance[m]: 1m
  - Xo Gaussian random variable

RTLS - Phase 2

- Range-based algorithms
  - Trilateration
  - MinMax
- Range-less algorithms
  - CL
  - WCL

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Framework

- Product of the thematic ICT week:
  - WSN Middleware
  - Software framework:
    - WSN (Telos rev. B & Sun Spot)
    - Controller + database
    - GUI
  - Distributed

Framework

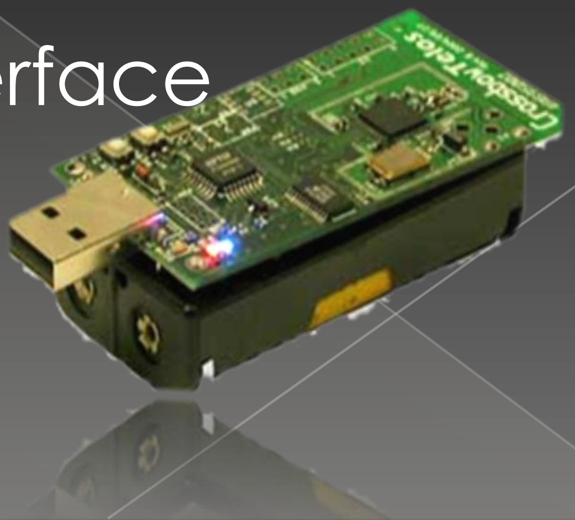
- Data interface to the WSNs and GUIs
  - XML
- Database
  - Stored Procedures
- Localization algorithms
  - Centralized

Framework - Technologies

- WSN
  - TinyOS
  - TelosB
  - Xubuntos
- WSN XML Parser
  - Java
- Controller, GUI
  - C#
  - .NET 3.5
- Interfaces
  - XML over TCP
  - WCF (http)

WSN - Telos rev. B

- TI MSP430 microcontroller with 10kB RAM
  - Ultra low-power
- IEEE 802.15.4 compliant radio
- Integrated temperature, light, humidity and voltage sensor
- Programmable via USB interface
- TinyOS 2.X compatible
- Integrated antenna



WSN - TinyOS

- Most popular OS for Wireless Sensor Networks
- Open source
- Energy efficient – low power
  - Hurry up and go to sleep!
  - Split phase commands
- Multi-platform

WSN - TinyOS

- Primary functions:
  - Sensing
  - Actuating
  - Communication
    - Collection
    - Dissemination

TinyOS - nesC

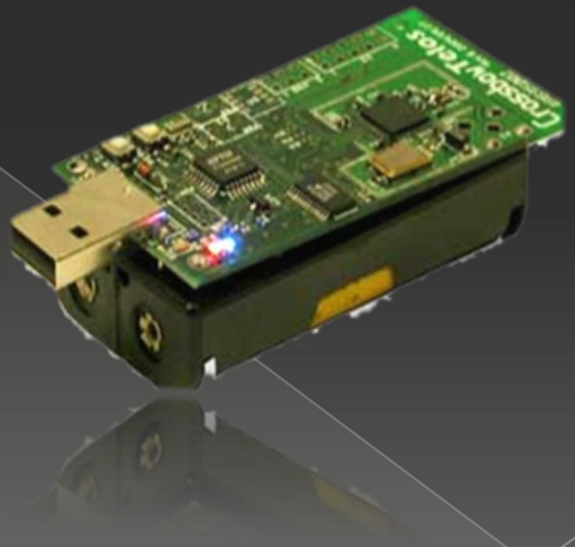
- TinyOS is competely programmed in nesC
  - Interfaces
  - Tasks
    - atomic
- nesC is a C dialect
- .nc
- Source code passes through a preproccessor
  - C-code
- Gcc

TinyOS

- Still very experimental & academic
- Limited support
- No development environment
  - No debugger
  - Printf library

WSN

- Three different roles:
  - Root Node
  - Anchor Node
  - Blind Node



WSN

- Three different messages:
  - Sensor
  - Location
  - Status

WSN - Sensor message

- Battery (voltage)
- Light
- Humidity
- Temperature
- Button pressed
- Mote ID

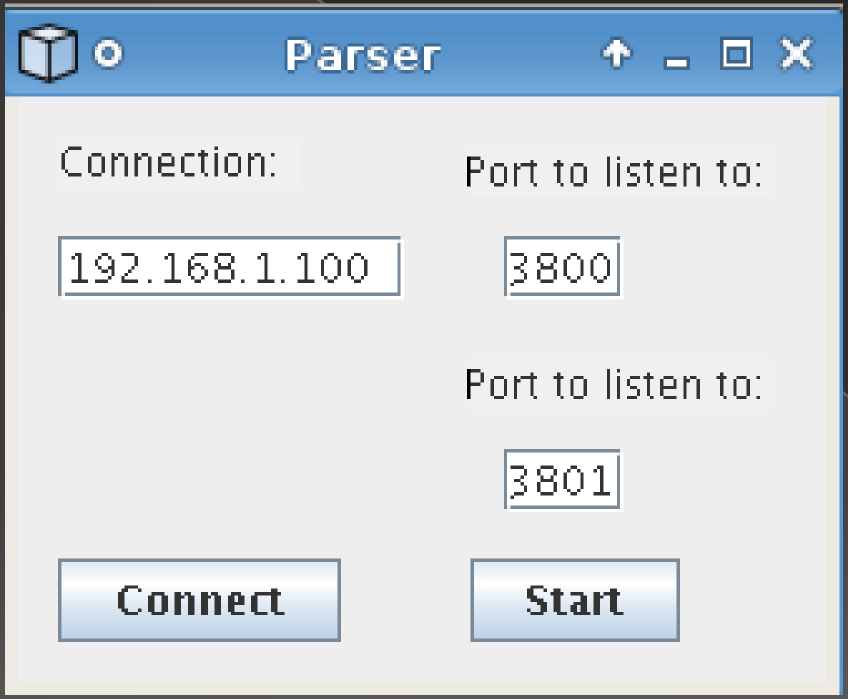
WSN - Location message

- Mote id
- Anmoteid
- VANs
- VANr
- Hop count
- RSSI

WSN - Status message

- Mote id
- Active
- AN
- Posx
- Posy
- Samplerate
- locRate
- leds
- power
- frequency

WSN - Parser


Parser
Connection:
192.168.1.100
Port to listen to:
3800
Port to listen to:
3801
Connect
Start

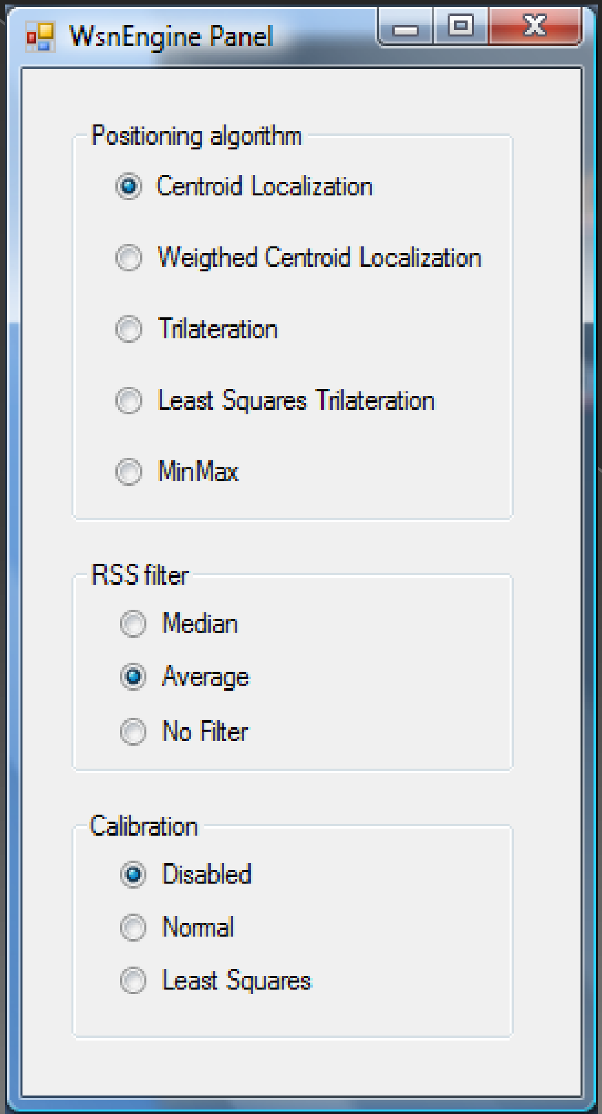
Database

- MySQL 5.0 database
  - ODBC
  - Stored Procedures

Controller

- Core of the system
- Gatekeeper to the database
- Central gathering point
- Localization support
- Interface to SCALA

Controller - WSN Engine panel

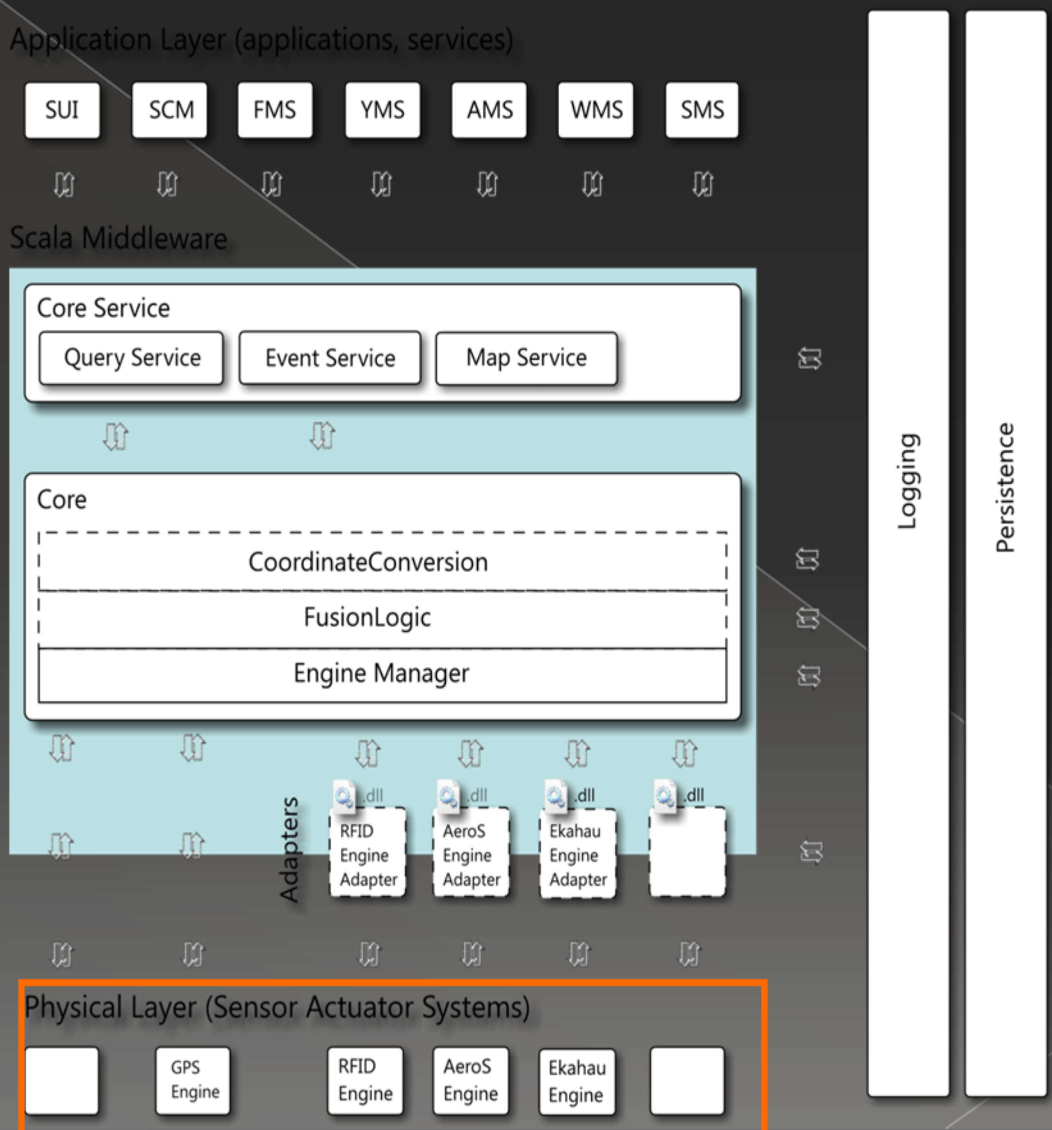


Scala

- RTLS Middleware
  - Next presentation
- Seamless integration of different locating systems

- Engine: our system
- Middleware: Scala.Core
- GUI: SUI

Scala - Engine



Scala

- Communication happens via a WCF service
  - http
  - Several interfaces
    - Tag Information
    - Event
    - Query
    - Map
  - Roughly based on the ANSI RTLS API

Scala - Data

- Location
  - X
  - Y
  - Map
  - Accuracy

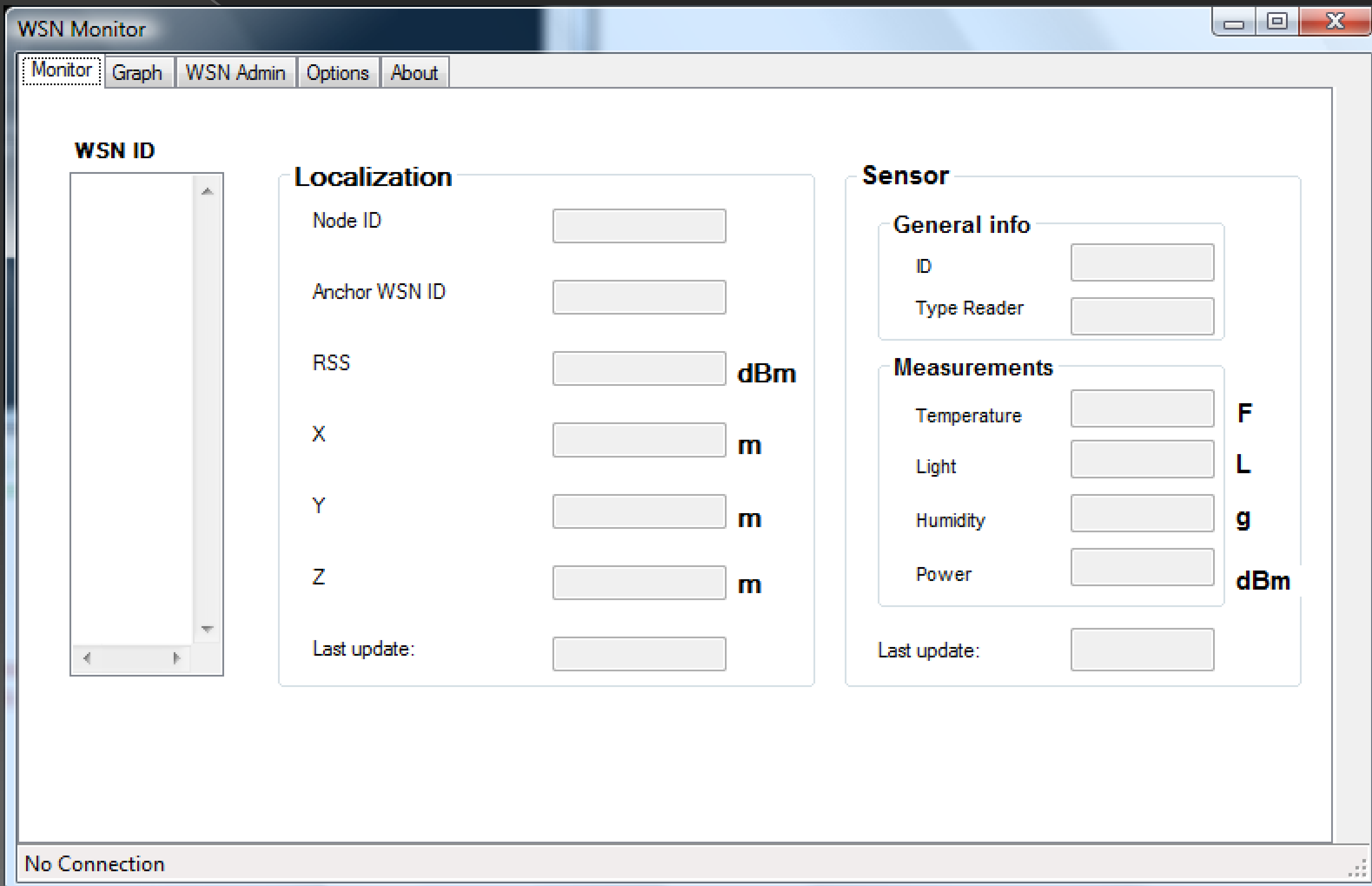
Scala - Data

- Temperature
- Humidity
- Light
- Button state

GUI

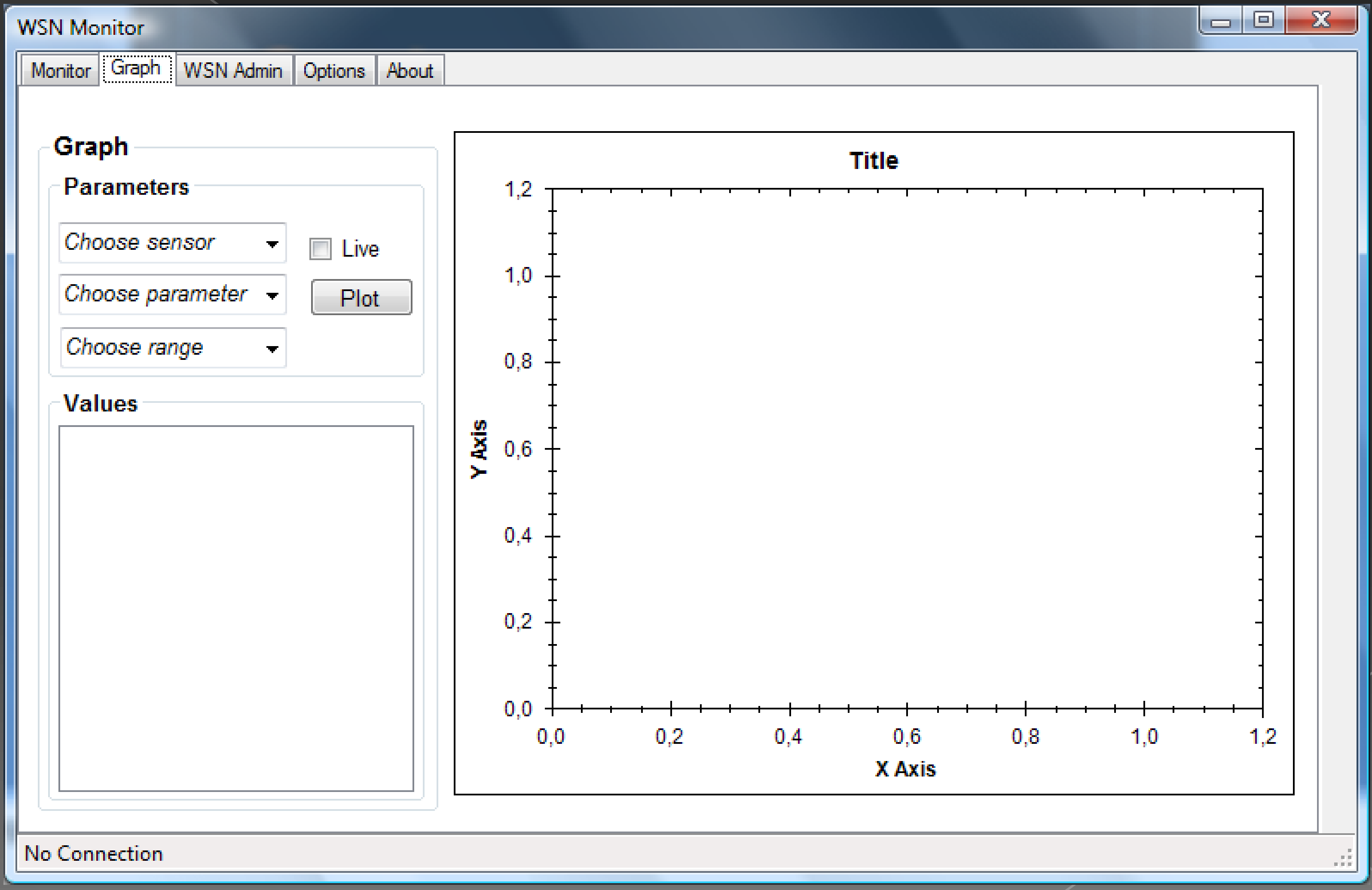
- Monitoring
- Controlling the WSN:
  - Active
  - Anchor node
  - Coordinates
  - Sample rate of location and sensor message
  - Leds

GUI - Monitor

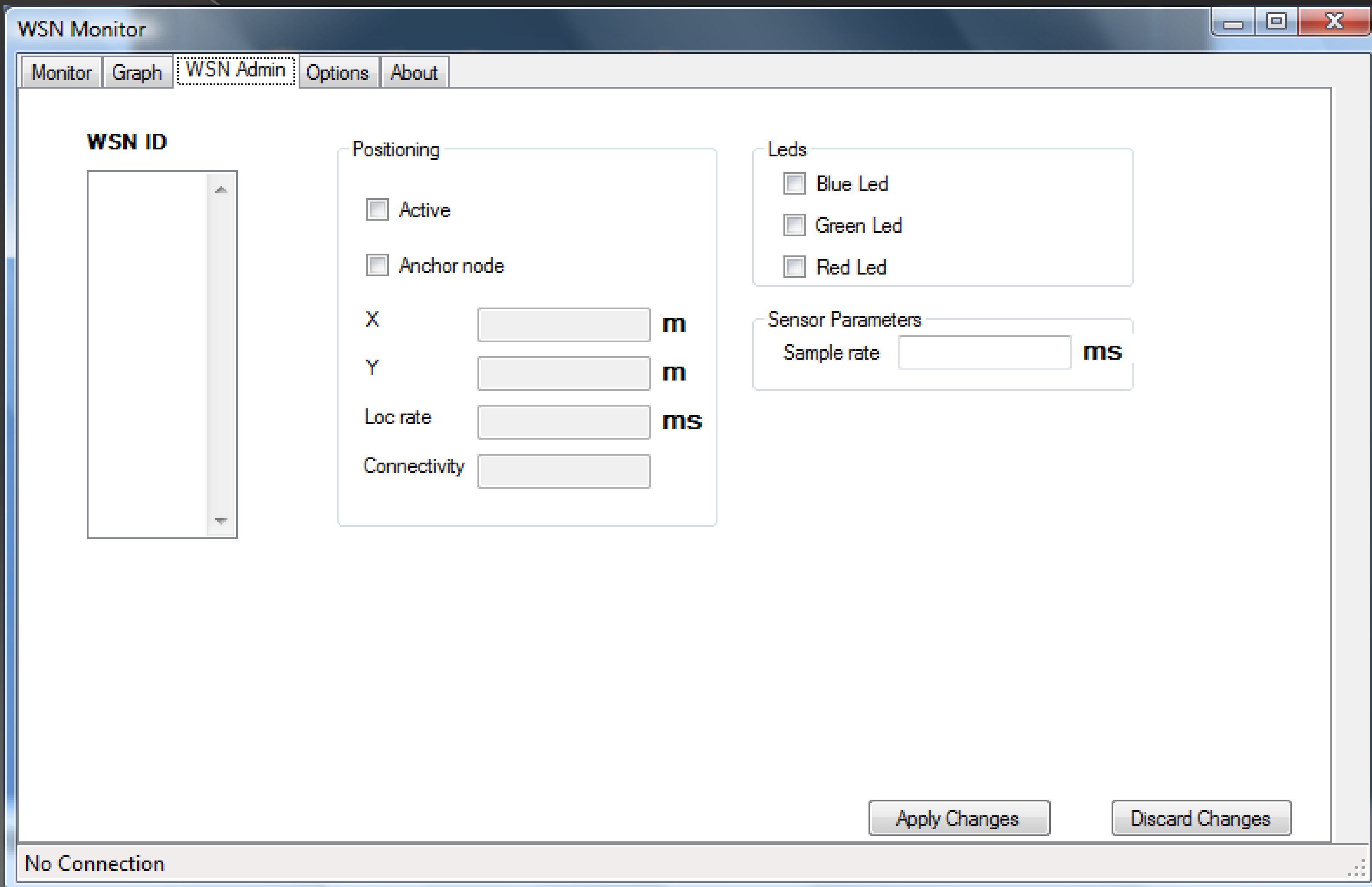


WSN Monitor
Monitor
Graph
WSN Admin
Options
About
WSN ID
Localization
Node ID
Anchor WSN ID
RSS
dBm
X
m
Y
m
Z
m
Last update:
Sensor
General info
ID
Type Reader
Measurements
Temperature
F
Light
L
Humidity
g
Power
dBm
Last update:
No Connection

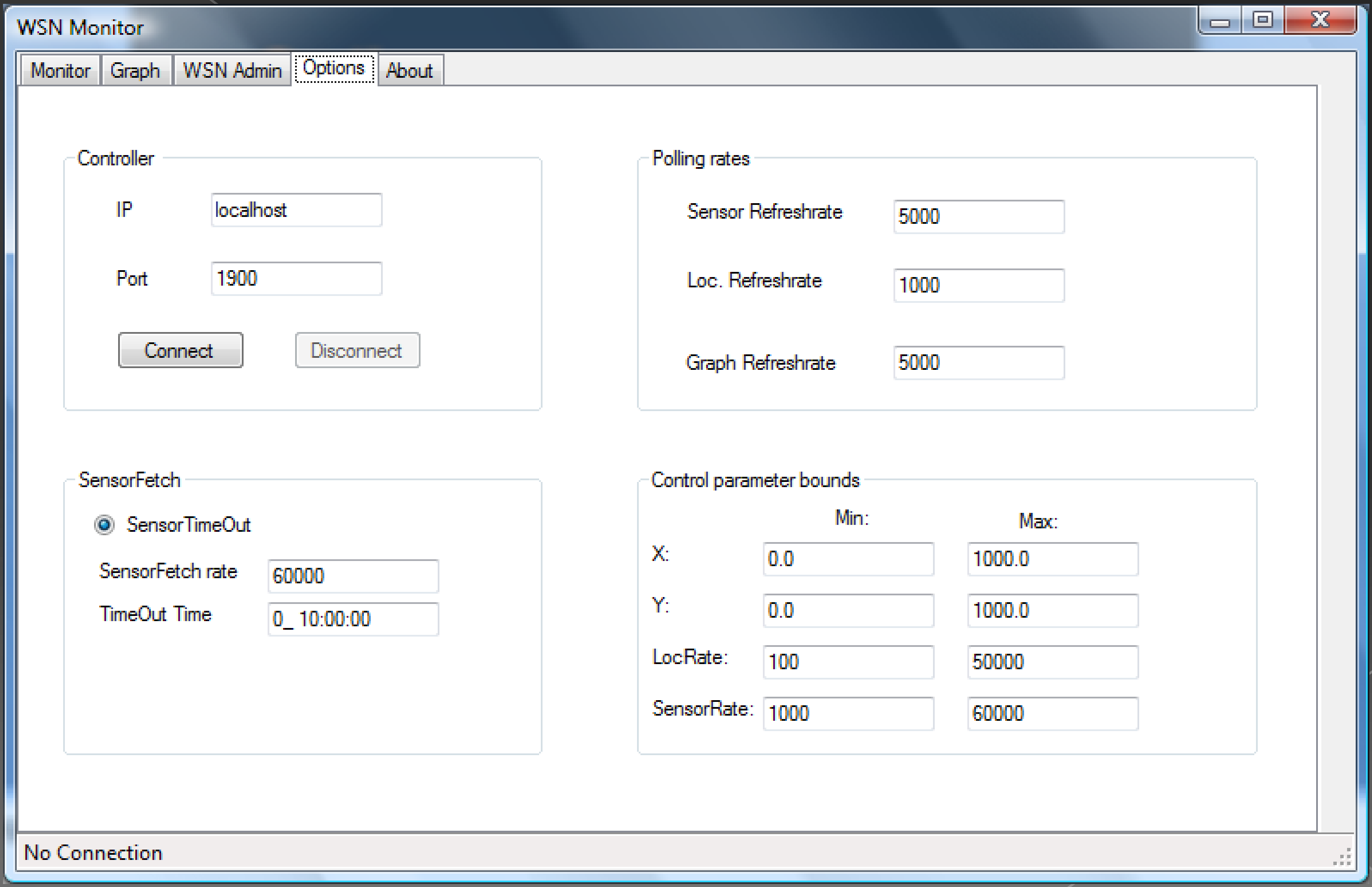
GUI - Graphs


WSN Monitor
Monitor
Graph
WSN Admin
Options
About
Graph
Parameters
Choose sensor
Live
Choose parameter
Plot
Choose range
Values
Title
1,2
1,0
0,8
0,6
0,4
0,2
0,0
Y Axis
0,0
0,2
0,4
0,6
0,8
1,0
1,2
X Axis
No Connection

GUI - Control panel



GUI - Options


WSN Monitor
Monitor
Graph
WSN Admin
Options
About
Controller
IP
localhost
Port
1900
Connect
Disconnect
Polling rates
Sensor Refreshrate
5000
Loc. Refreshrate
1000
Graph Refreshrate
5000
SensorFetch
SensorTimeOut
SensorFetch rate
60000
TimeOut Time
0_ 10:00:00
Control parameter bounds
Min:
Max:
X:
0.0
1000.0
Y:
0.0
1000.0
LocRate:
100
50000
SensorRate:
1000
60000
No Connection

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Localization

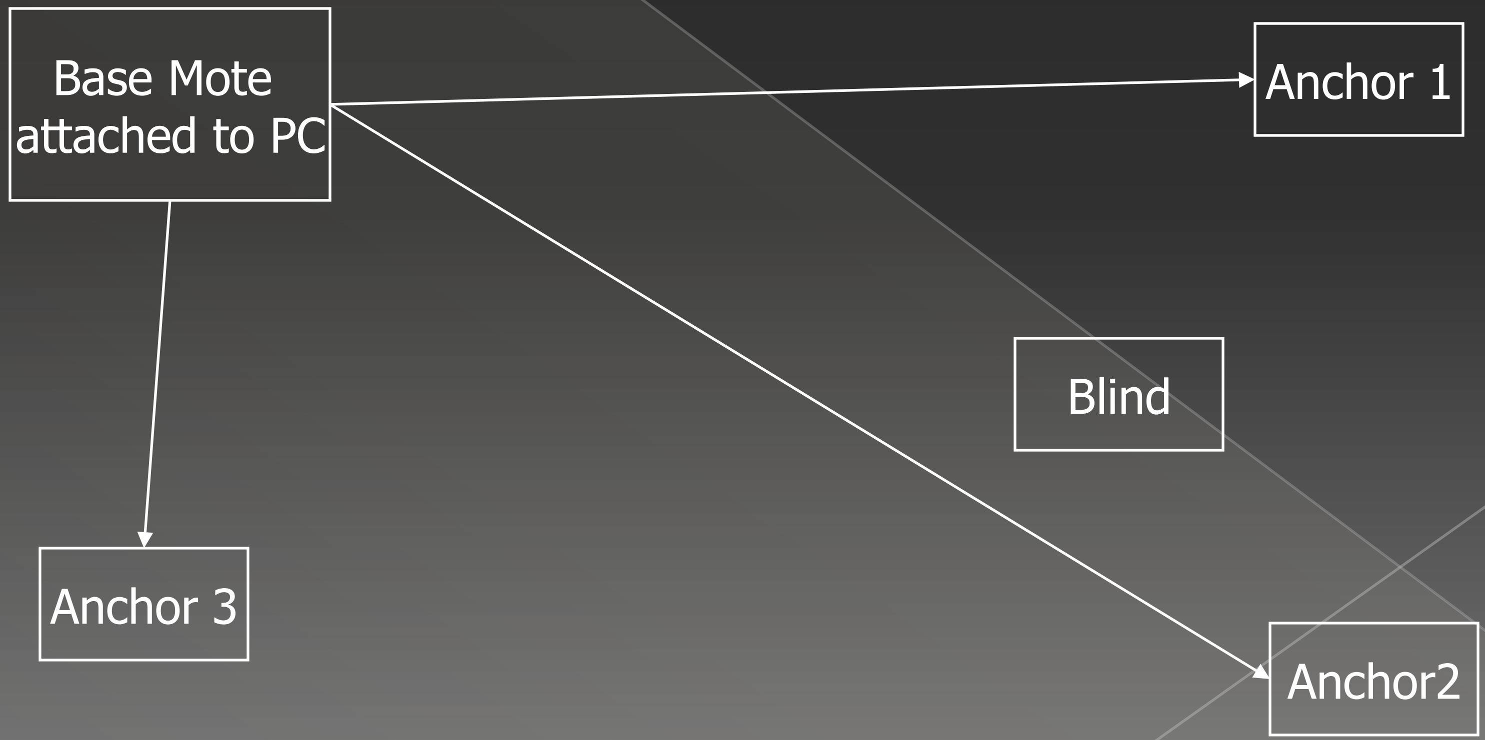
- 2 phases:
  - Ranging + calibration
  - Algorithms

Localization - Ranging

- RSS(d) = - P(d0) – 10 n log(d / d0)
  - RSS Received Signal Strength[dBm]
  - P(d0) Path loss in dBm at a distance of d0
  - n Path loss exponent
  - d Distance between two nodes[m]
  - d(0) Reference distance[m]: 1m

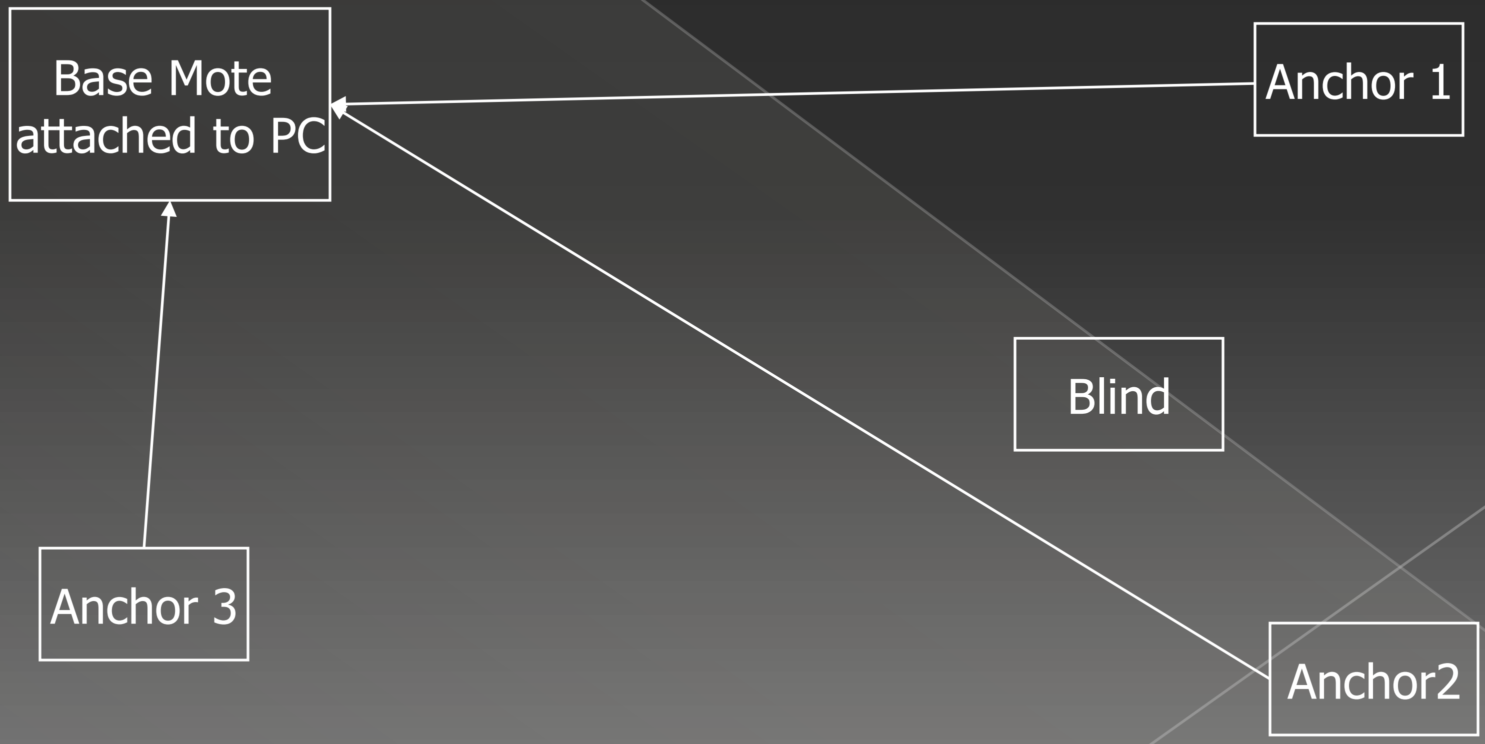
Localization - calibration

- Configure anchor nodes with dissemination protocol



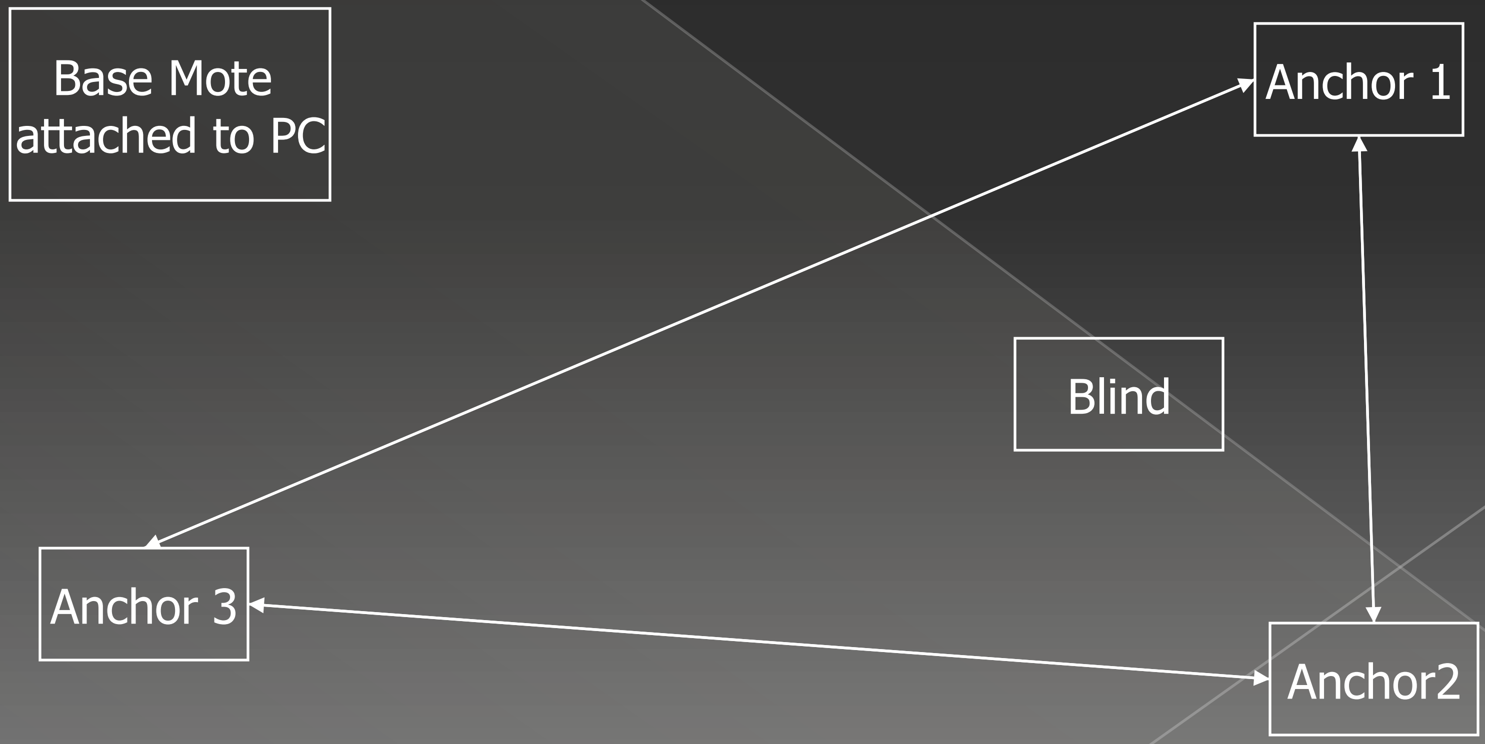
Localization - calibration

- Confirmation with a status message



Localization - calibration

- Broadcast in order to measure RSSI



Localization - calibration

- Send back RSSI with the collection protocol

Localization – calibration (LS)

- RSS(d) = - P(d0) – 10 n log(d / d0)
  - RSS Received Signal Strength[dBm]

$$\begin{bmatrix} RSS1 \\ \cdot \\ \cdot \\ \cdot \\ RSSi \end{bmatrix} = \begin{bmatrix} -1 & -10log\,\frac{d1}{d0} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ -1 & -10log\,\frac{di}{d0} \end{bmatrix} \times \begin{bmatrix} P(d0 \\ n \end{bmatrix}$$

$$\alpha \qquad\qquad \beta \qquad\qquad \Omega$$

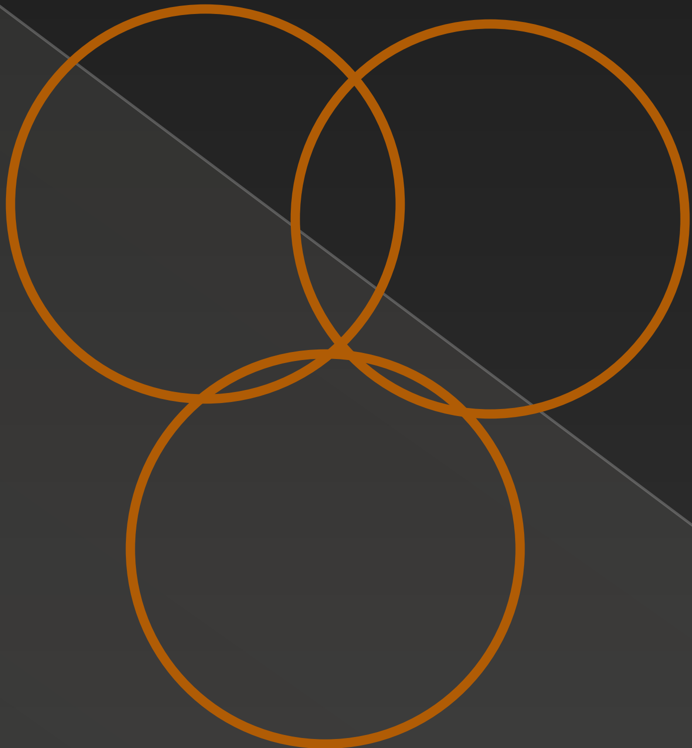
$$\Omega = (\beta^T \times \beta)^{-1} \times \beta^T \times \alpha$$

Localization - Algorithms

- Trilateration
- Min-Max
- CL
- WCL

Trilateration

- Lateration needs (in theory) distance measurements from:
  - 3 non-collinear references to compute a 2D position



- Circle:

$$(x-x1)^2 + (y-y1)^2 = r1^2$$

$$\vdots$$

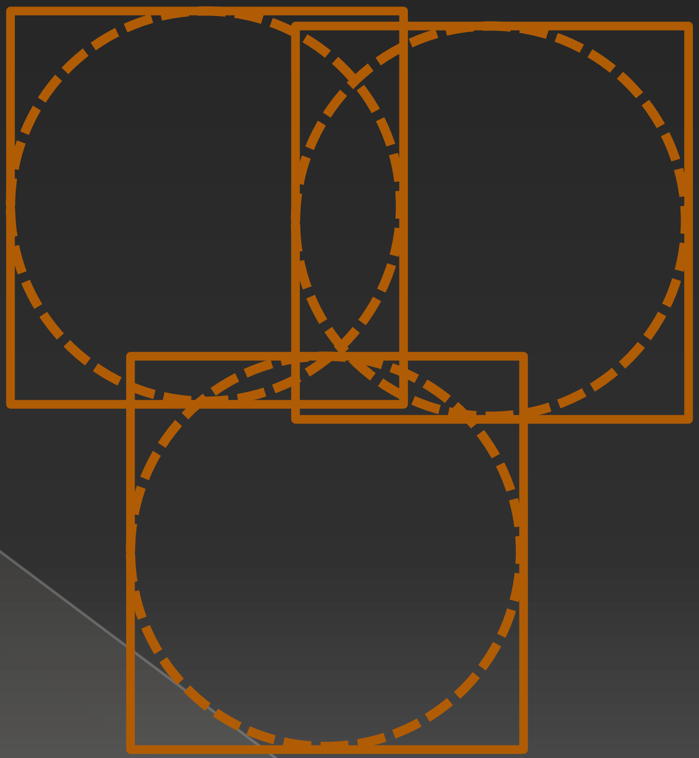
$$(x-xk)^2 + (y-yk)^2 = rk^2$$

$$\underbrace{2 \times \begin{bmatrix} x2 - x1 & y2 - y1 \\ \vdots & \vdots \\ xk - x1 & yk - y2 \end{bmatrix}}_{\alpha} \times \underbrace{\begin{bmatrix} x \\ y \end{bmatrix}}_{\beta} = \underbrace{\begin{bmatrix} x2^2 - x1^2 + y2^2 - y1^2 + r1^2 - r2^2 \\ \vdots \\ xk^2 - x1^2 + yk^2 - y1^2 + r1^2 - rk^2 \end{bmatrix}}_{\Omega}$$

$$\Omega = \frac{1}{2}(\beta^T \times \beta)^{-1} \times \beta^T \times \alpha$$

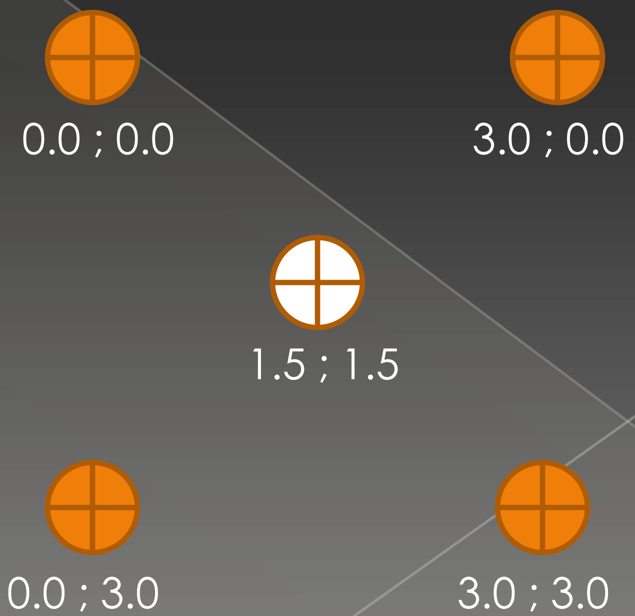
Min-Max

- Lateration is computation-heavy; a good simplification models around each anchor node a bounding box and estimates position at the intersection of boxes



Centroid localization

- Coarse grained localization
- calculate the unknown position as the centroid of the anchor nodes within their communication range



Weighted CL

- A weight is coupled to the position of each anchor node by its RSS.

$$Weight = \frac{1}{RSS^{g}}$$

Localization - methods

- Antenna orientation
  - Onboard - External
  - 20°
  - Outdoor
  - 1 & 5 meter
- Algorithms (outdoor)

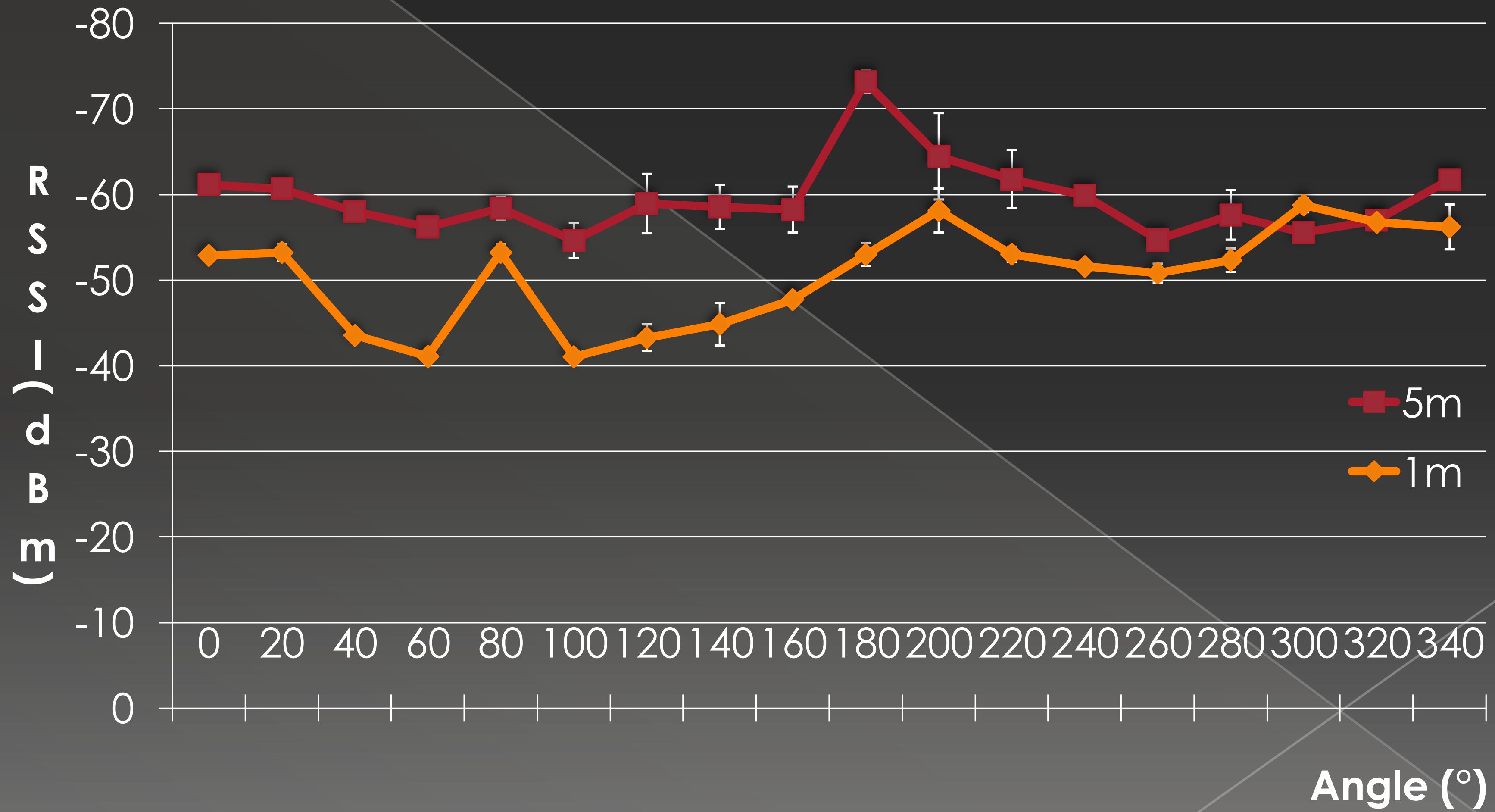
Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

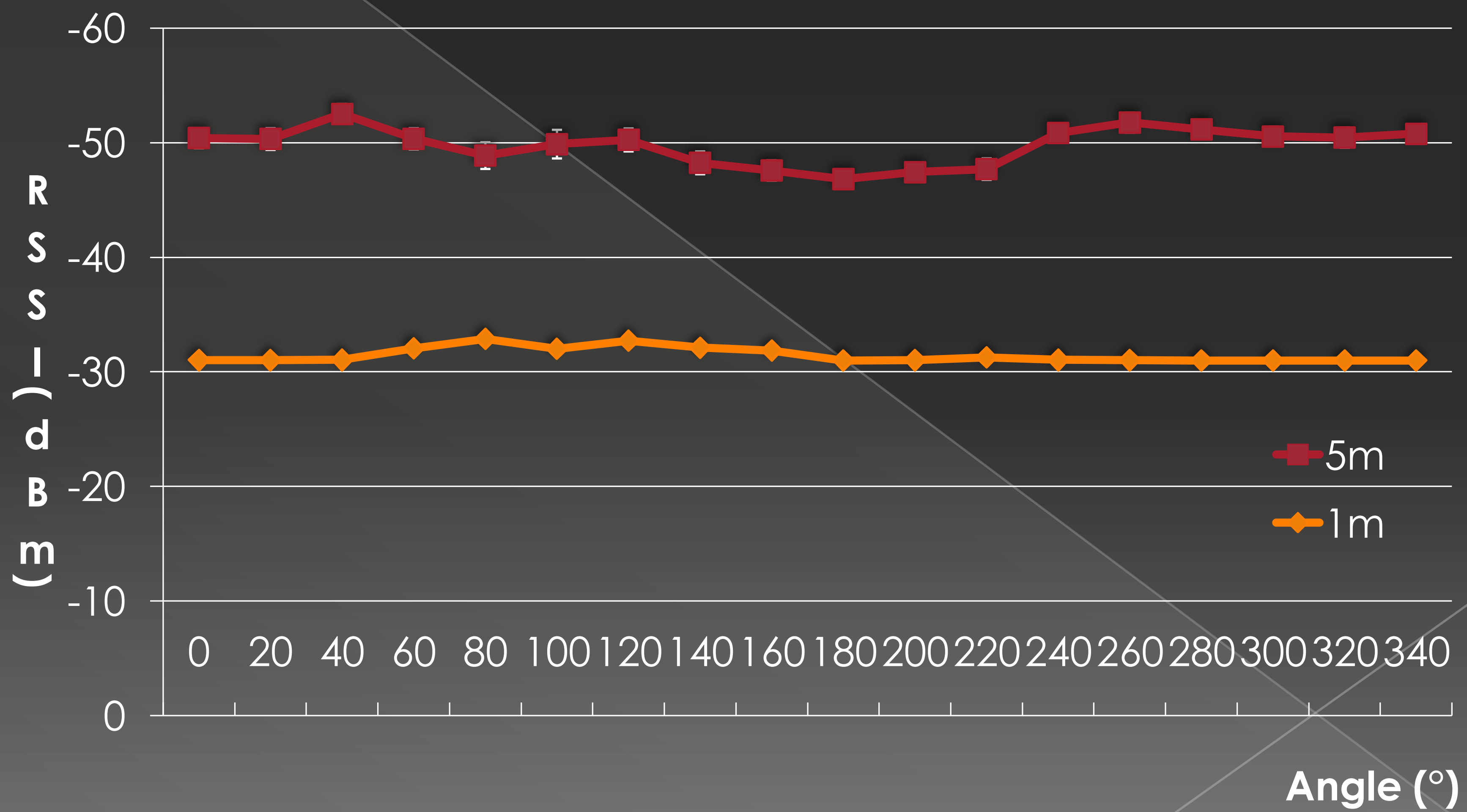
Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

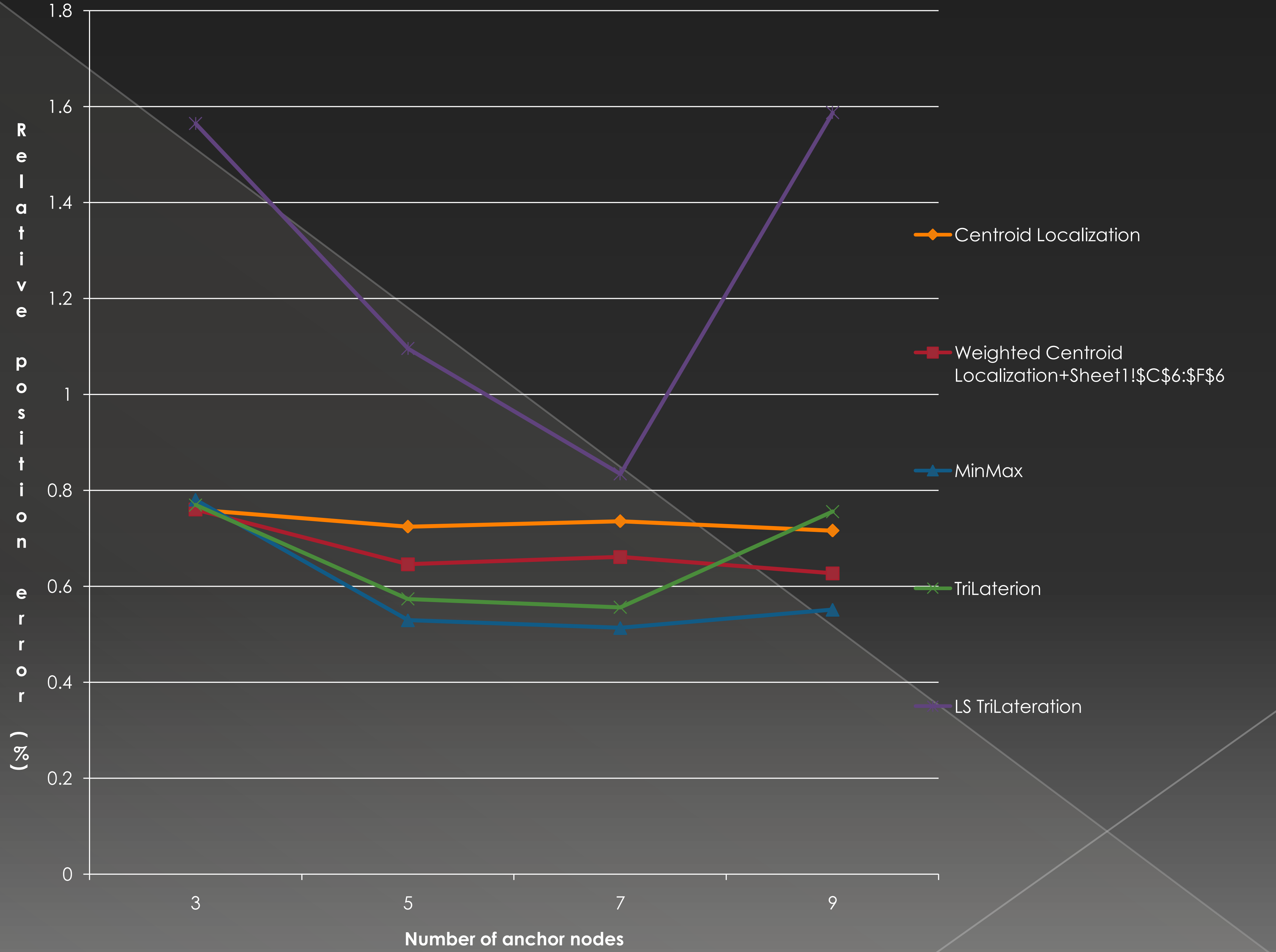
Results – Orientation



Results - Orientation



Outdoor positioning


1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Relative position error (%)
3
5
7
9
Number of anchor nodes
Centroid Localization
Weighted Centroid Localization+Sheet1!$C$6:$F$6
MinMax
TriLaterion
LS TriLateration

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Conclusion

- Successfully enhanced the framework and implemented different localization algorithms
- Made a working interface to Scala
- Made a WSN Configuration Tool

- Spent too much time on the framework, too few on the algorithms

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Future work

- Simplify the framework
- Distributed?
- Database and object integrity / ORM
- Implement interfaces with WCF
- Event-based
- C-based serial forwarder under Windows
- More algorithms!
- Implement algorithms distributed
- Find / help develop tool to make developing WSN applications more simple and less time-consuming

Live Demo!

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A

Overview

- Contributions
- Motivation
- Applications
- WSN as a RTLS
- Framework
- Localization
- Results
- Conclusion
- Future work
- Q&A