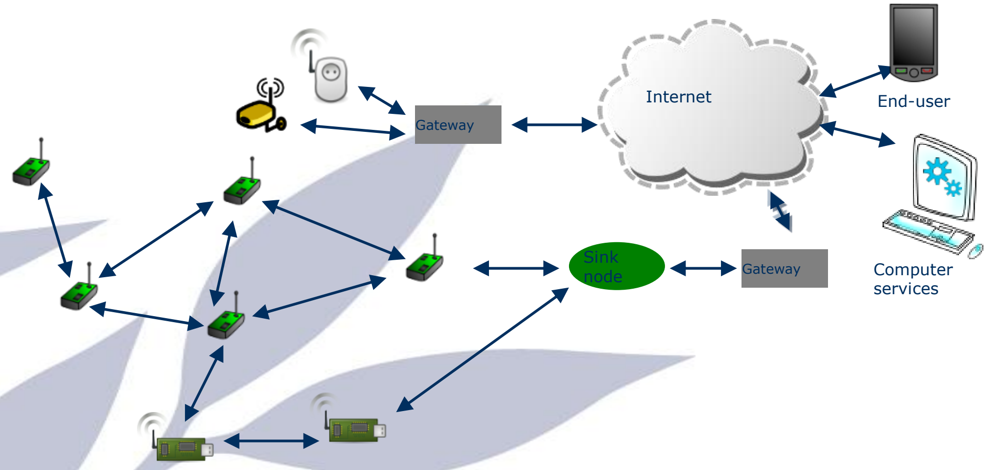


Continuous All k Nearest Neighbor Queries in Smartphone Networks- Proximity Sensing, Part I

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6341377

Stockholms universitet

Outline

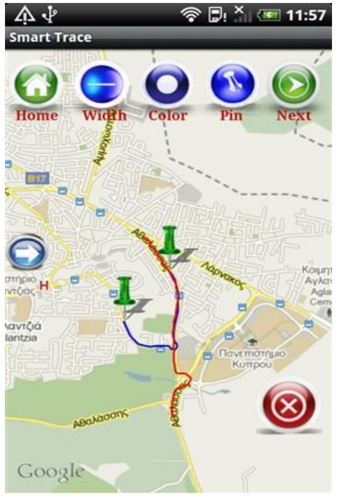
- **Motivation**
- System Model and Problem Formulation
- Proximity Algorithm
- Experimental Evaluation

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6341377

Stockholms universitet

Smartphones

- **Smartphone:** A powerful sensing device!
  - **Processing:** 2x2.73 GHz Octa-core Samsung Galaxy S10)
  - **RAM & Flash Storage:** 6GB & 128GB, respectively
  - **Networking:** WiFi, 3G (Mbps) / 4G (100Mbps–1Gbps)
  - **Sensing:** Proximity, Ambient Light, Accelerometer, Microphone, Geographic Coordinates based on AGPS (fine), WiFi or Cellular Towers (coarse) & etc.
- ***In-House Applications!***



SmartTrace
(ICDE'09,MDM'09,TKDE'12)

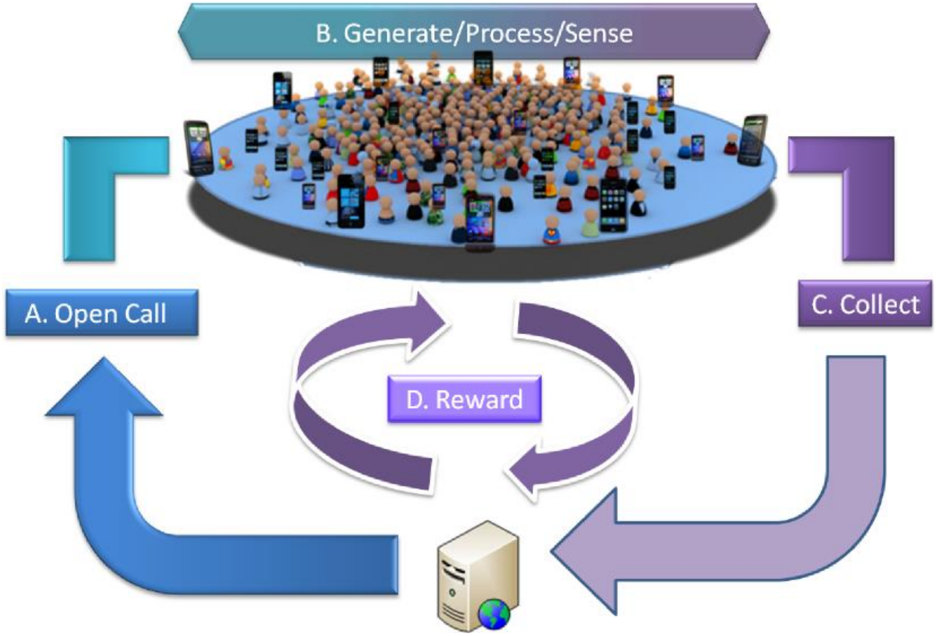
SmartP2P
(MDM'11, MDM'12)

Airplace
(MobiSys'12, MDM'12)

Stockholms universitet

Motivation

- Crowdsourcing with Smartphones
  - A smartphone crowd is constantly moving and sensing providing large amounts of opportunistic data that enables new services and applications

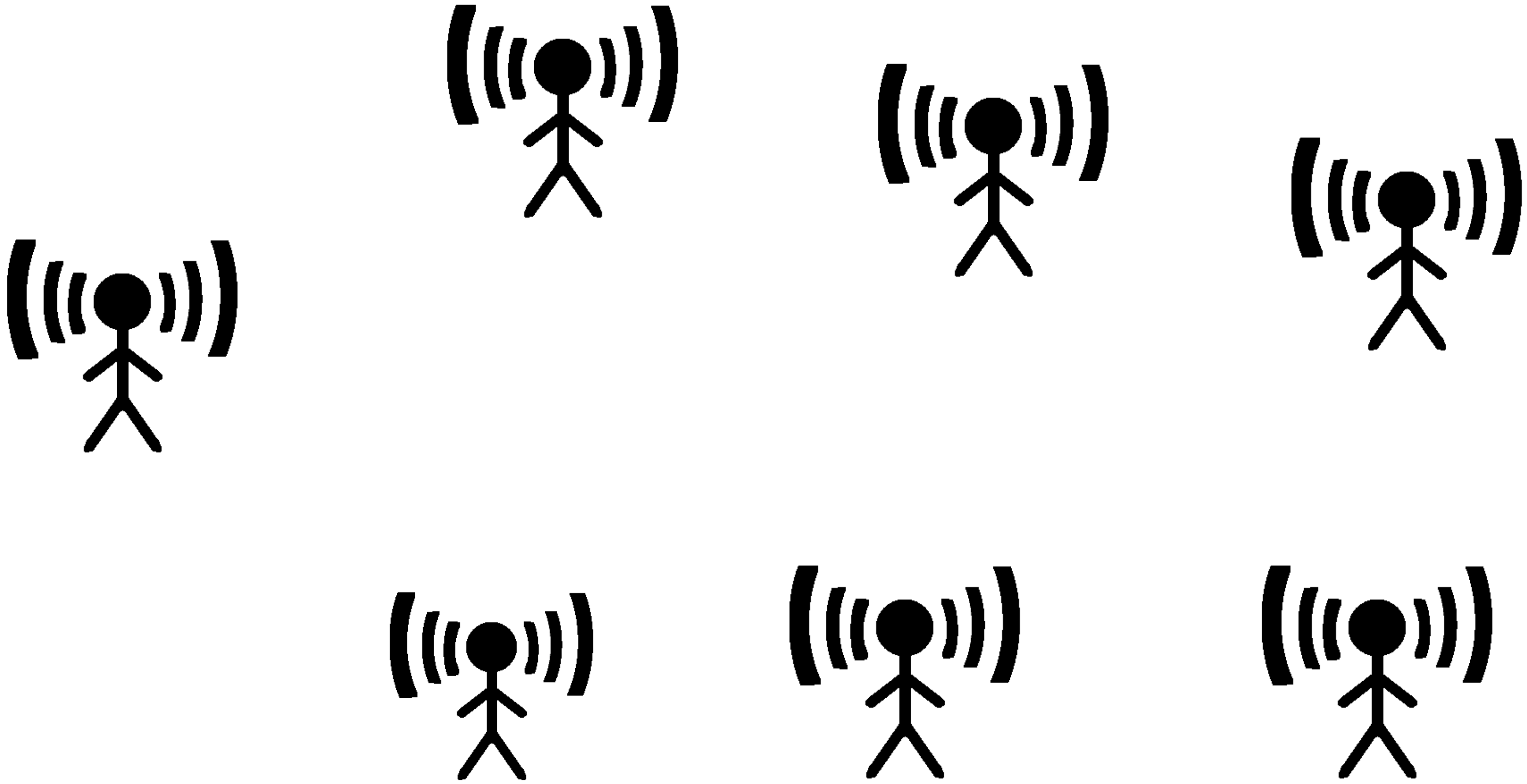


Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6341377

Stockholms universitet

Continuous k Nearest Neighbor Queries

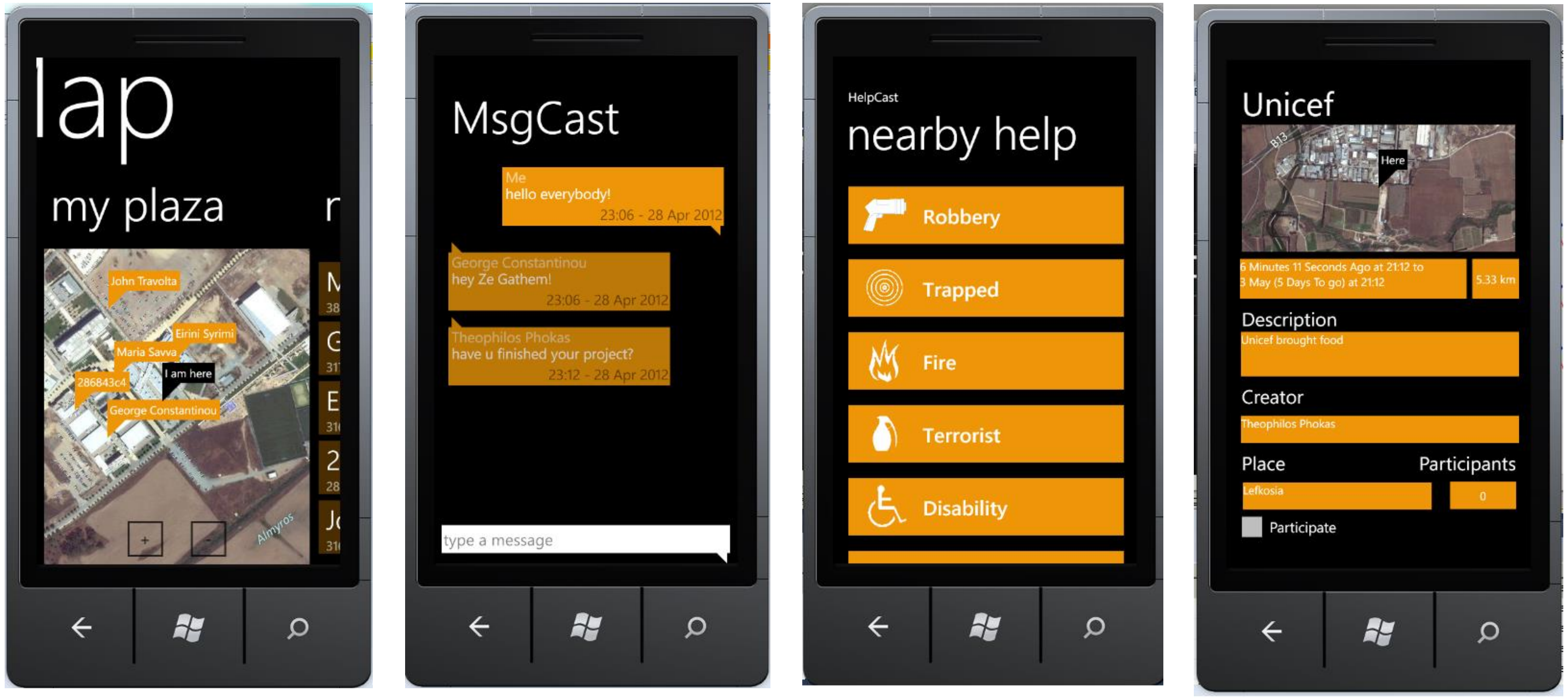
Find 2 Closest Neighbors for 1 User



Stockholms universitet

Motivation

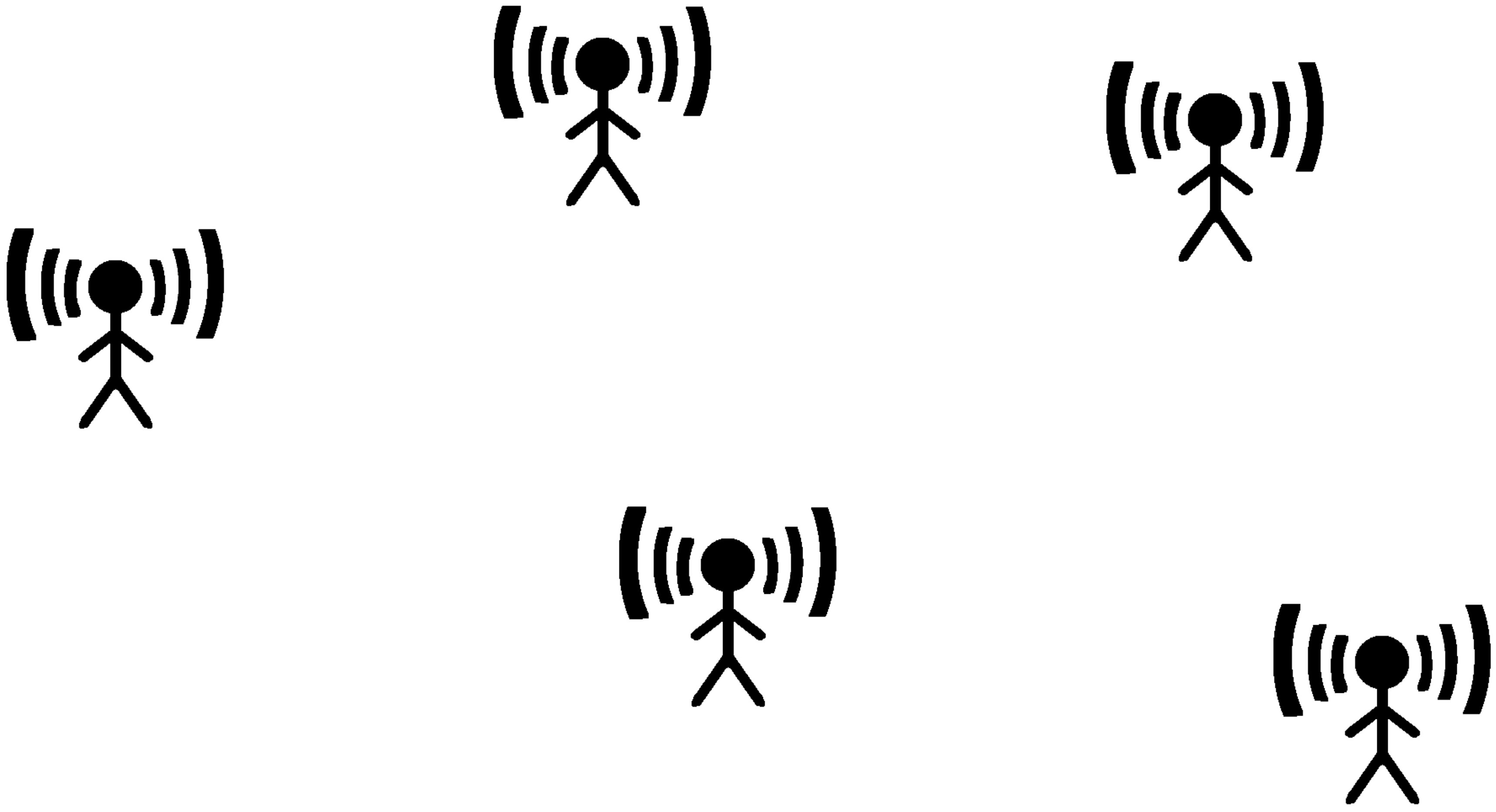
"Create a Framework for Efficient
***Proximity** Interactions "*



Screenshots from a prototype system for Windows Phone
http://www.zegathem.com/

Continuous All K Nearest Neighbor (CAkNN) Queries

Find 2 Closest Neighbors for ALL User



Stockholms universitet

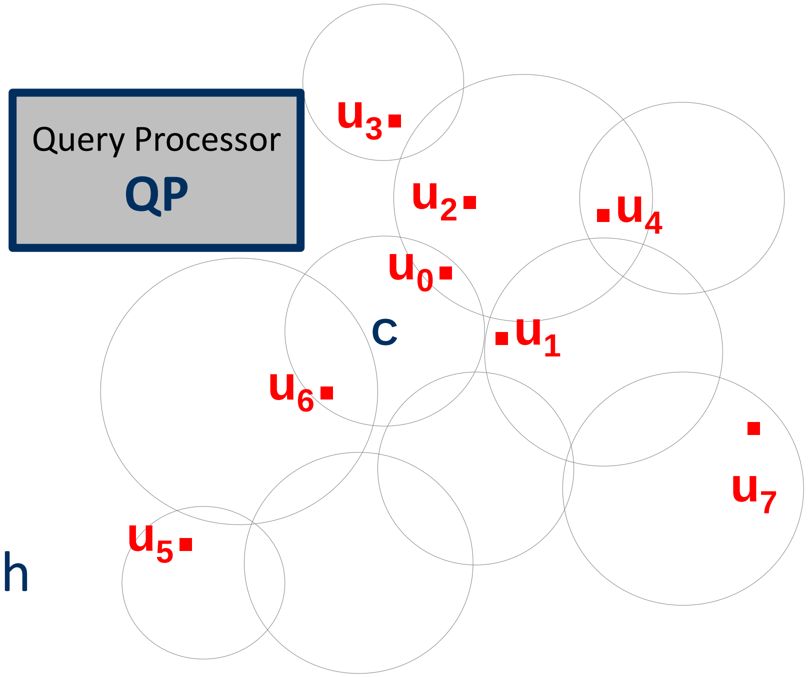
Outline

- Motivation
- **System Model and Problem Formulation**
- Proximity Algorithm
- Experimental Evaluation

Stockholms universitet

System Model

- A set of users moving in the plane of a region (u_1, u_2, … u_n)
- Area covered by a set of ***Network Connectivity Points*** **(*NCP*)**
  - Each *NCP* creates the notion of a *cell*
  - W.l.o.g., let the cell be represented by a circular area with an arbitrary radius
- A mobile user **u** is **serviced** at any given time point by one *NCP.*
- There is some centralized service, denoted as **QP (Query Processor),** which is aware of the coverage of each NCP.
- Each user u reports its **positional information** to **QP** regularly

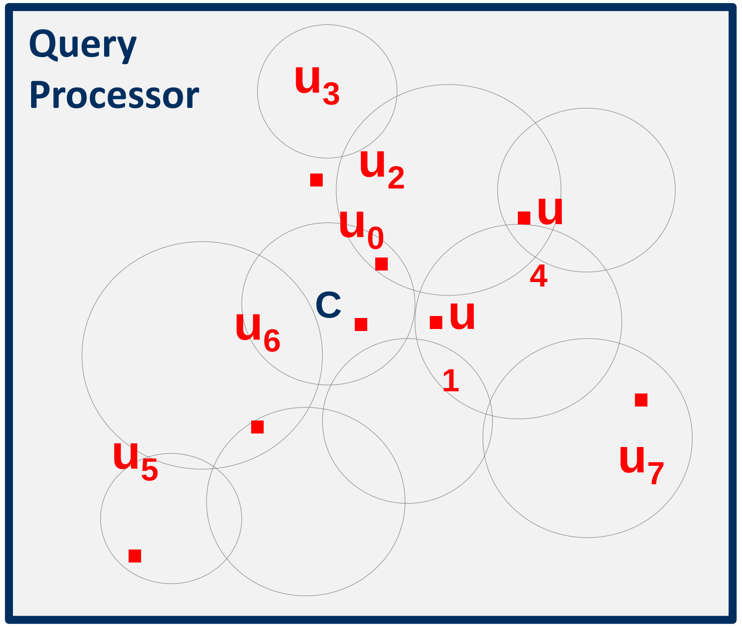


Stockholms universitet

Problem Definition

- ***Definition* of the CAkNN *problem:***

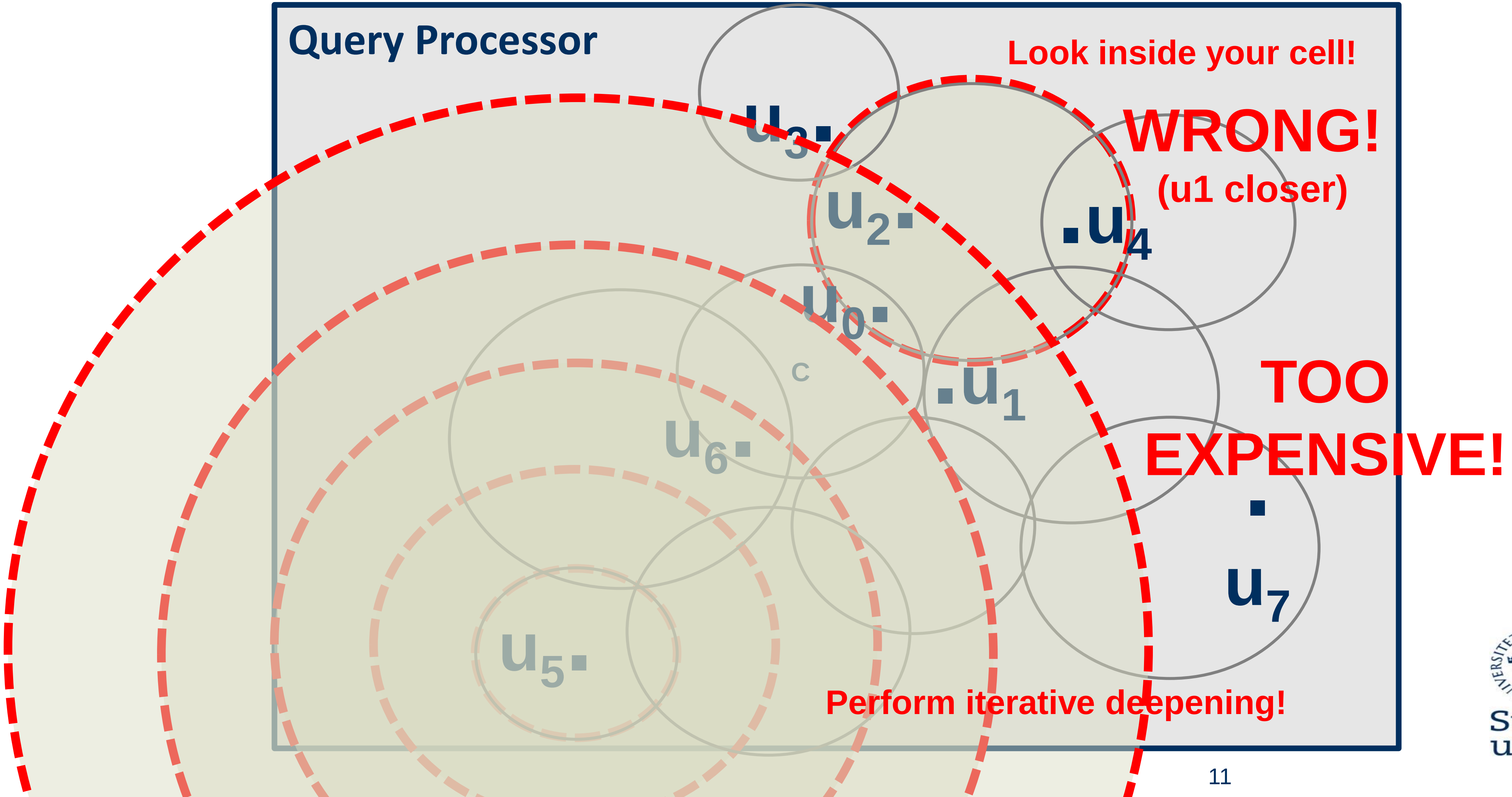
  Given a set **U** of **n** users and their location reports $\mathbf{r_{i,t} \in R}$ at *timestep* $\mathbf{t \in T}$, then the *CAkNN* problem is to find in each *timestep* $\mathbf{t \in T}$ and for each user $\mathbf{u_i \in U}$ the **k** objects $\mathbf{U_{sol} \subseteq U - u_i}$ such that for all other objects $\mathbf{u_o \in U - U_{sol} - u_i}$, $\mathbf{dist(u_k, u_i) \leq dist(u_o, u_i)}$ holds



Stockholms
universitet

Naïve Solution

***Find 2-NN** for u_0 at timestep t.* ***For** u_5?*



Stockholms universitet

Stateless & Stateful

- Existing algorithms for **CkNN (not CAkNN)**

  Yu et al. (YPK) [11] and Mouratidis et al. (CPM) [12]

- **Stateless Version**: *Iteratively expand* the search space for each user into *neighboring cells* to find the *kNN (like previous slide)*
- **Stateful Version**: Improve the stateless version by *utilizing previous state*, under the *following assumptions*:

  Not Appropriate for CAkNN

  - i) Static Querying User (i.e., designated for point queries)
  - ii) Target users move slowly (i.e., state does not decay)
  - iii) Few Target users

11. Yu, Pu, Koudas. "Monitoring k-nearest neighbor queries over moving objects," ICDE '05
12. Mouratidis, Papadias, Hadjieleftheriou, "Conceptual partitioning: an efficient method for continuous nearest neighbor monitoring," SIGMOD '05.

Stockholms universitet

Outline

- Motivation
- System Model and Problem Formulation
- **Proximity Algorithm**
- Experimental Evaluation

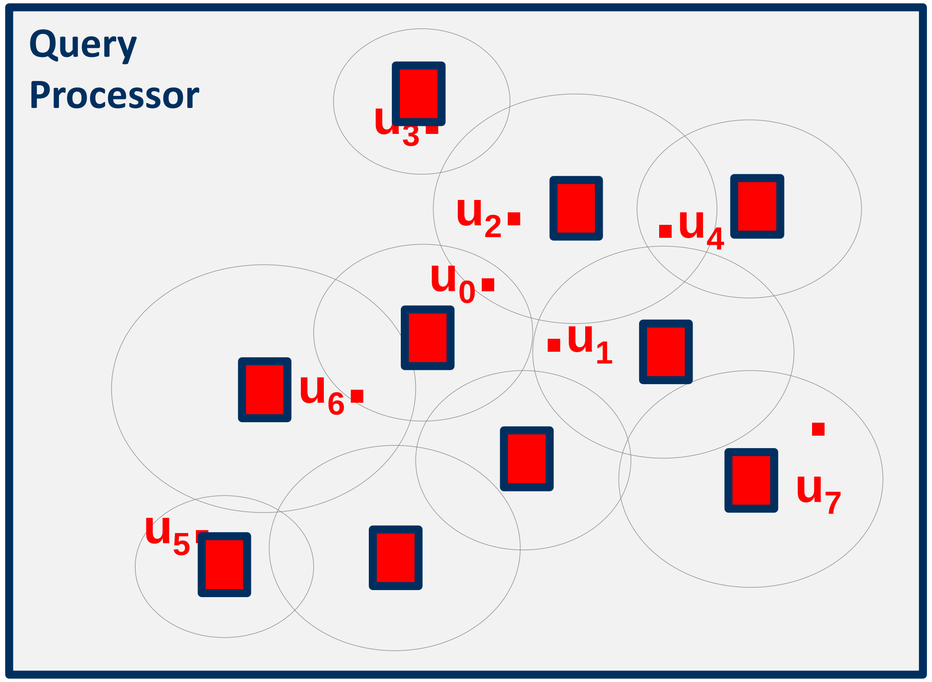
Proximity Overview

- The first specialized algorithm for ***Continuous All k-NN (CAkNN)*** queries
- Important characteristics:
  - **Stateless** (i.e., optimized for **high mobility**)
  - **Batch processing** (i.e., with **search space searching**)
  - **Parameter-free** (i.e., no tuning parameters)
- Generic operator for proximity-based queries
  - See Crowdcast application presented later.

Proximity Outline

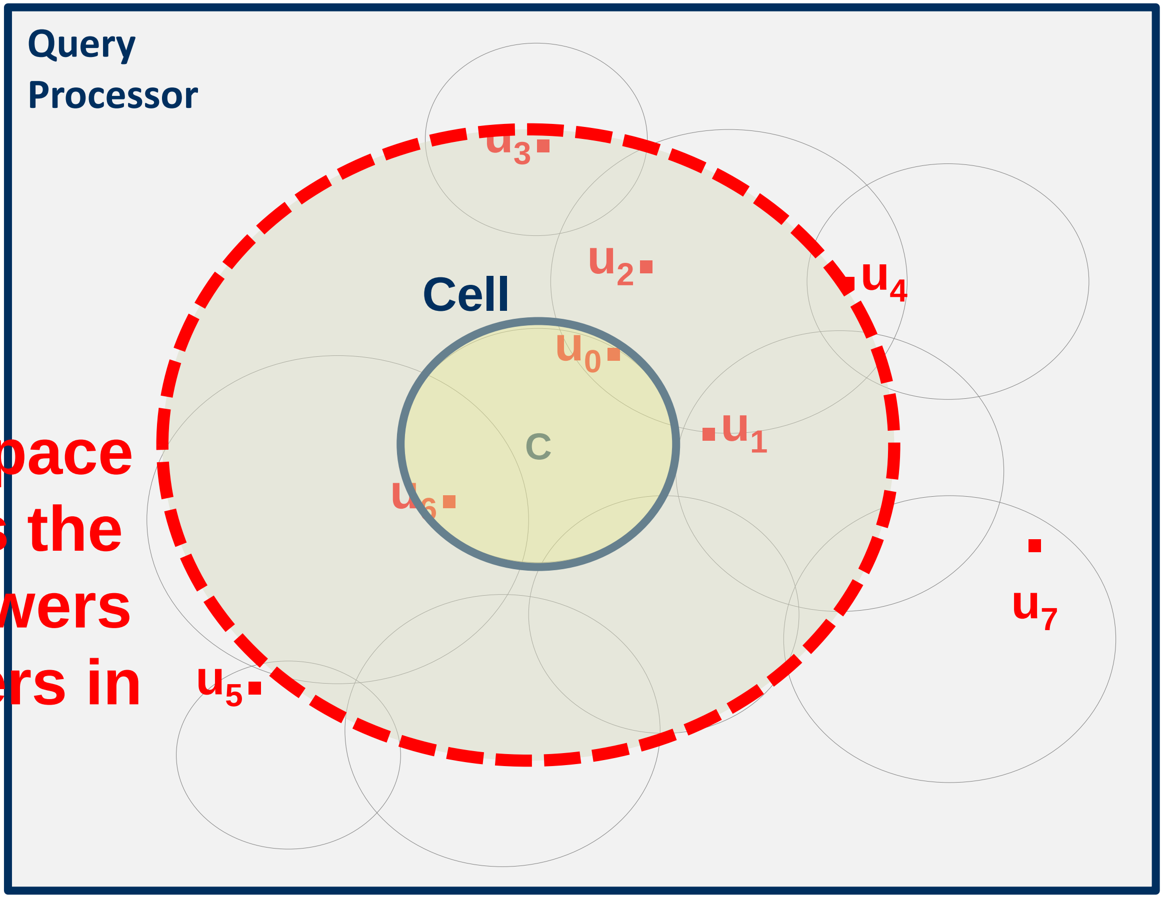
- For every timestep:

1. Initialize a ***k*⁺-heap** for every cell
2. Insert **every** user's location report to every ***k*⁺-heap**
   - Notice that k^+-heap is a heap-based structure and most location reports will be dropped as a result of an insert operation
3. For every user scan the k^+-heap of his cell to find his k-NN



Intuition behind Proximity

- Users in a cell will share the **same search space** (search space sharing)
- Compute 1 search space per cell only!

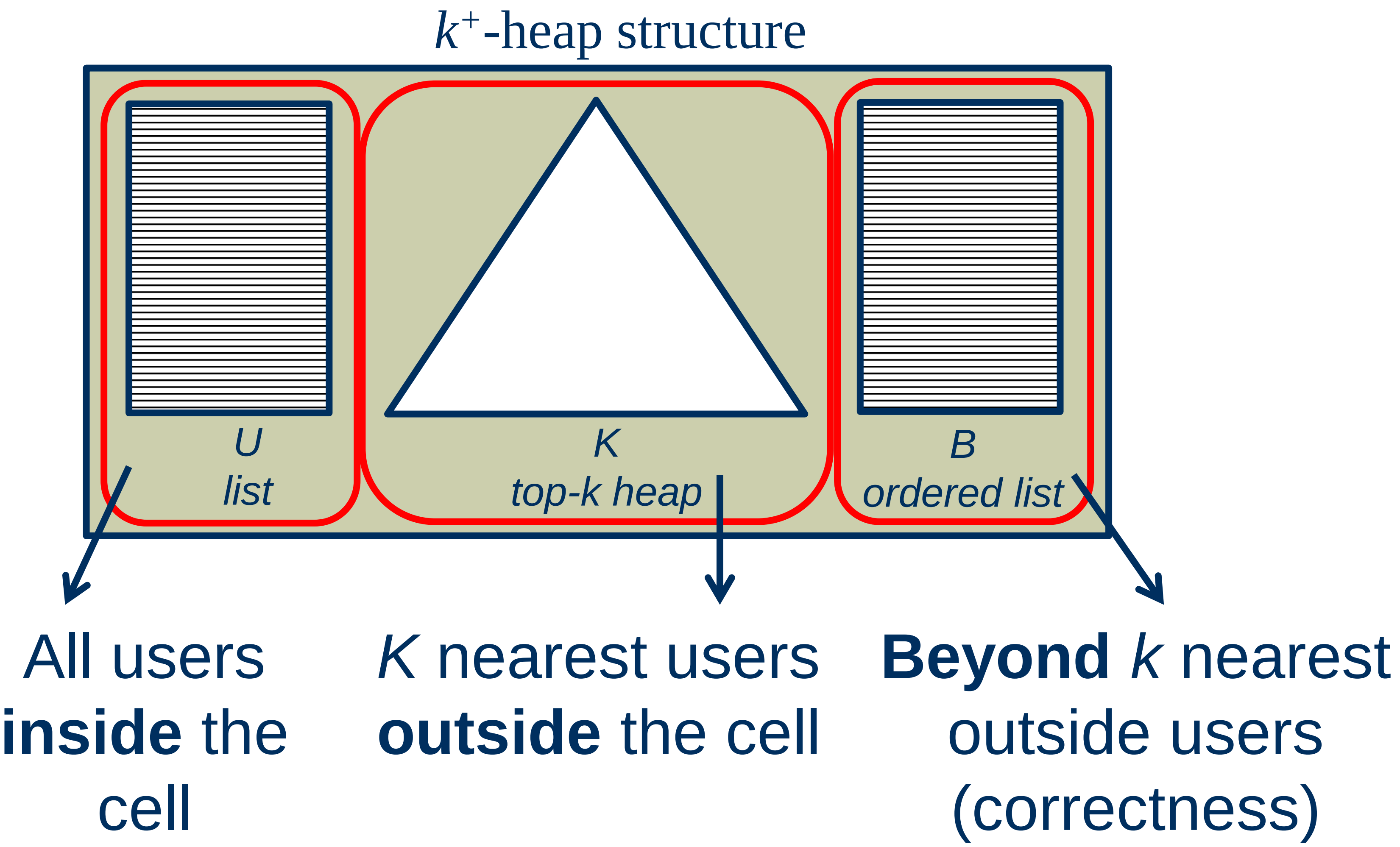


Search Space (contains the right answers for all users in C)

Stockholms universitet

Proximity k^+-heap

- ***The k⁺*-heap structure for a cell**

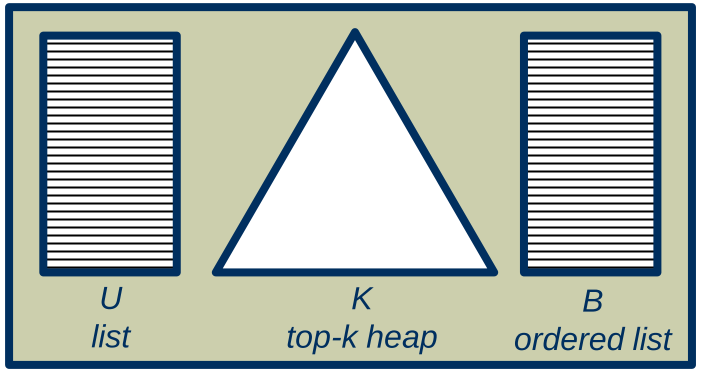


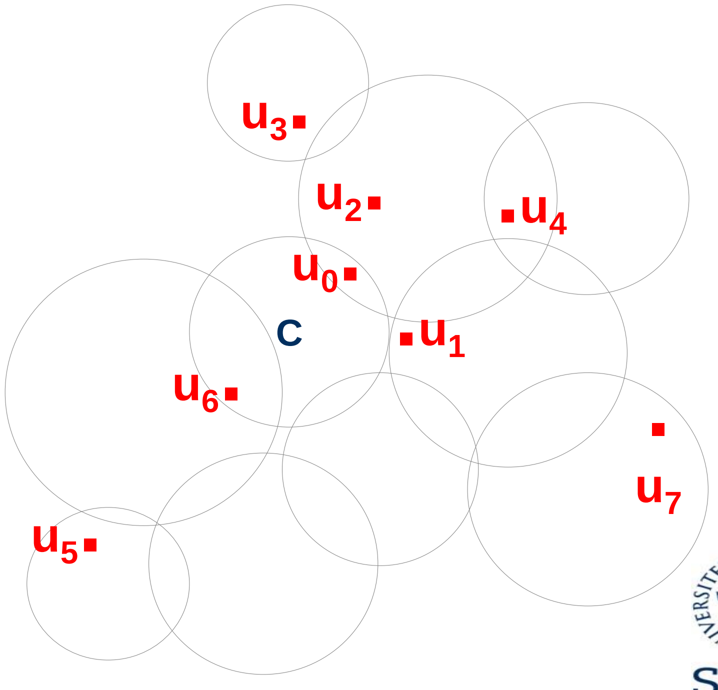
k^+-heap Construction (for Cell C)

Assume k=2.

Arriving Reports	Structure U	Structure K	Structure B
$\mathbf{u_6}$	u_6		
$\mathbf{u_4}$	u_6	u_4	
$\mathbf{u_7}$	u_6	u_7, u_4	
$\mathbf{u_2}$	u_6	u_4, u_2	u_7
$\mathbf{u_3}$	u_6	u_3, u_2	u_4, u_7
$\mathbf{u_1}$	u_6	u_2, u_1	u_3, u_4
$\mathbf{u_5}$	u_6	u_2, u_1	u_3, u_4, u_5
$\mathbf{u_0}$	u_6, u_0	u_2, u_1	u_3, u_4, u_5

k^+-heap structure



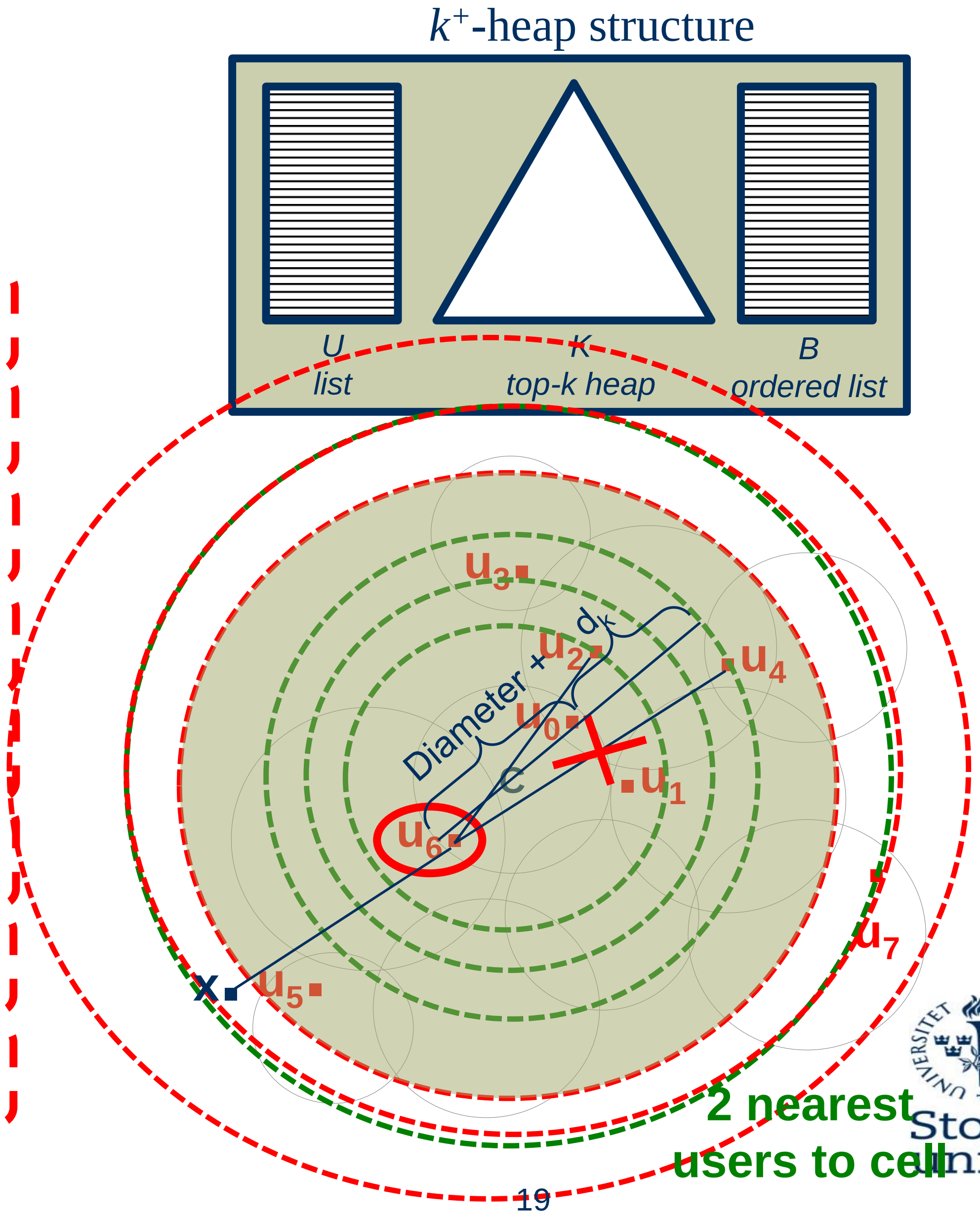


Stockholms universitet

k^+-heap Construction (for Cell C)

Assume k=2.

Arriving Reports	Structure U	Structure K	Structure B
u_6	u_6		
u_4	u_6	u_4	
u_7	u_6	u_7, u_4	
u_2	u_6	u_4, u_2	u_7
u_3	u_6	u_3, u_2	u_4, u_7
u_1	u_6	u_2, u_1	u_3, u_4
u_5	u_6	u_2, u_1	u_3, u_4, u_5
u_0	u_6, u_0	u_2, u_1	u_3, u_4, u_5



Outline

- Motivation
- System Model and Problem Formulation
- Proximity Algorithm
- **Experimental Evaluation**

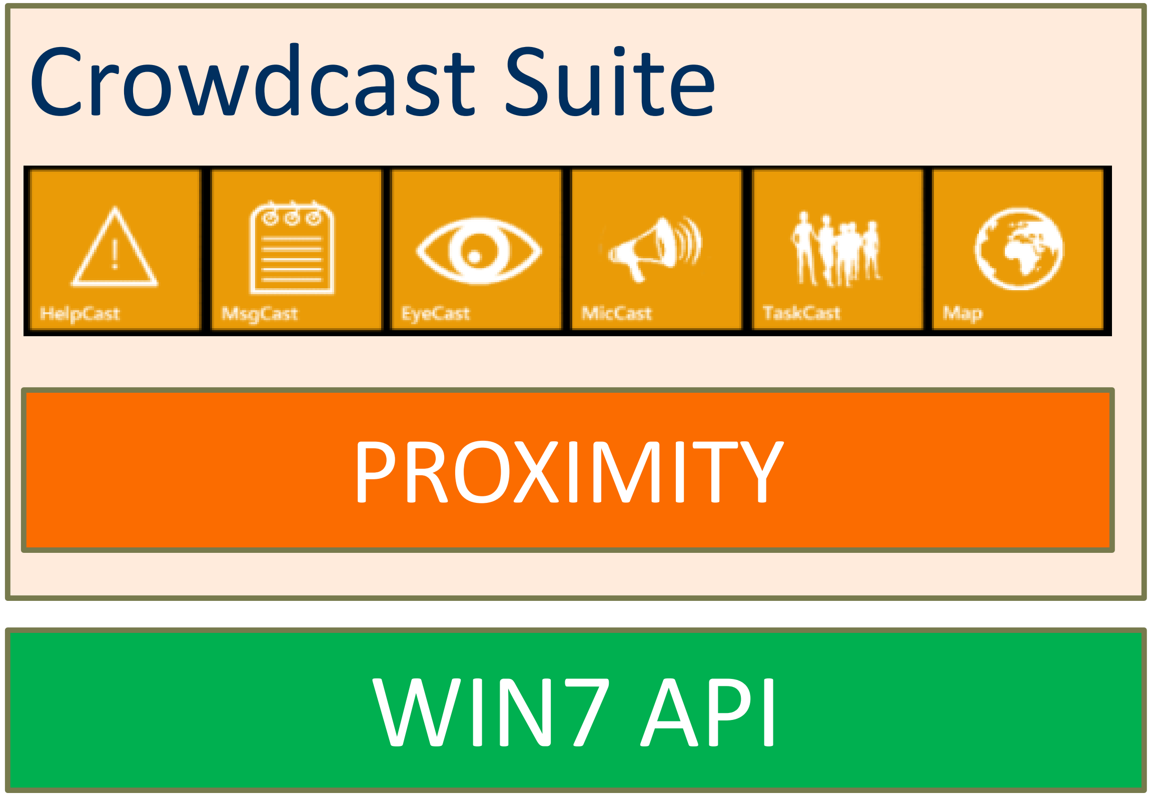
Stockholms universitet

Experimental Methodology

- **Dataset**:
  - Oldenburg Dataset:
    - Spatiotemporal generator with roadmap of Oldenburg as input (25km x 25km)
    - 5000 maximum users (Oldenburg population: 160.000)
  - Manhattan Dataset:
    - Vehicular mobility generator with roadmap of Manhattan as input (3km x 3km)
    - 500 users
- Network connectivity points (*NCP*s) uniformly in space
  - Communication ranges 1km, 4km and 16km for the *Oldenburg* dataset and ranges 1km and 4km for the *Manhattan* dataset
- **Query:** *CAkNN*
- **Comparison:** Proximity vs YPK vs CPM (using the optimal parameter value for YPK and CPM)

Stockholms universitet

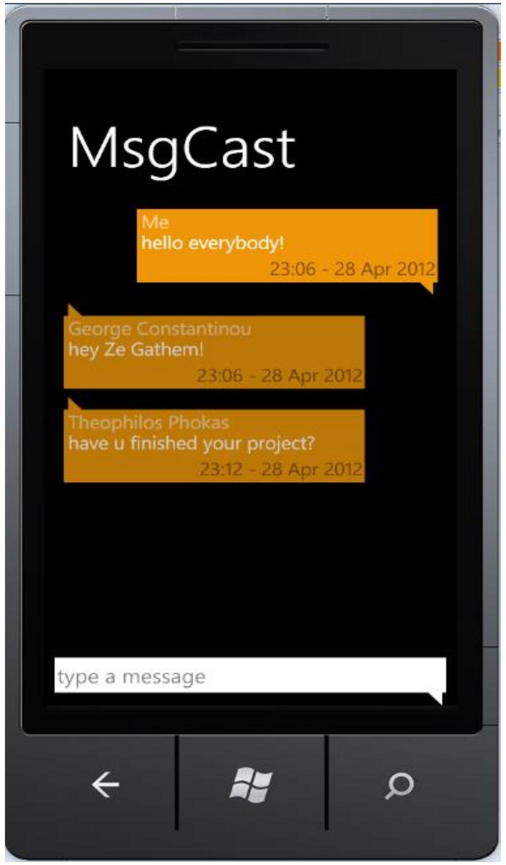
The Crowdcast Application Suite



Ranked 3rd in Microsoft's *ImagineCup* contest local competition (to appear in the next demo session!)

Stockholms universitet

Crowdcast: MsgCast



- Location-based Chat Channel (Post / Follow)
- Provide Guidelines

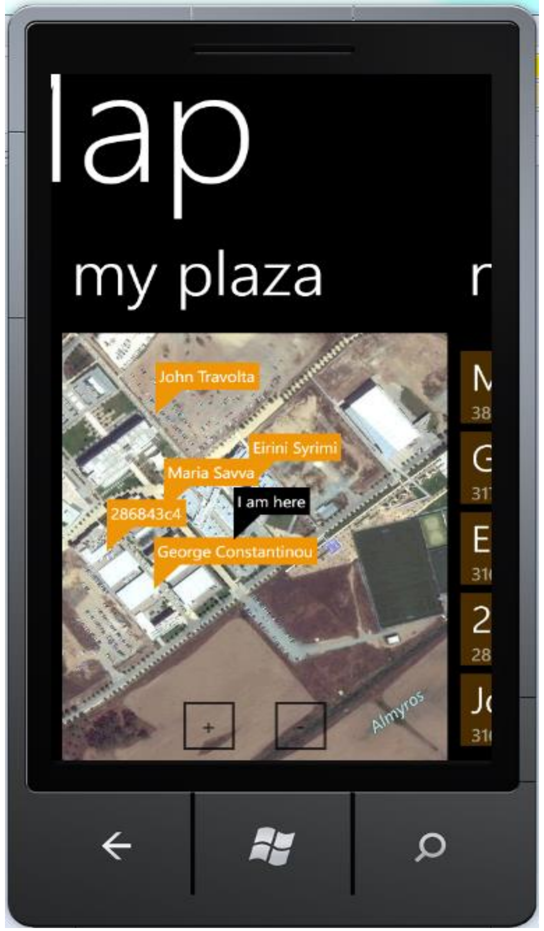
"Get your location-based questions answered!"

Stockholms universitet

Crowdcast: HelpCast

HelpCast

"The Ubiquitous Help Platform for Anyone in Need"



Stockholms universitet

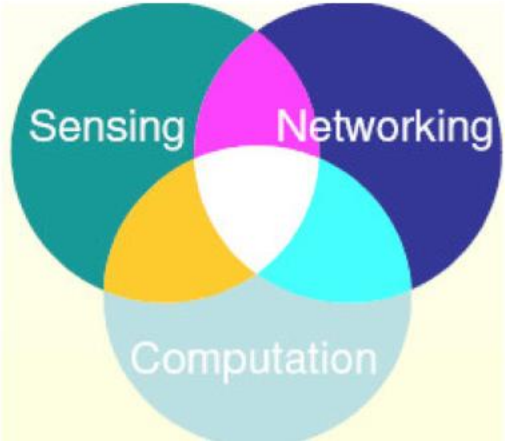
Crowdcast: EyeCast

- Disaster Recovery Operations
- Indoor Video Conferencing Network

"Extend your Vision beyond your 2 eyes!"

Stockholms universitet

IOT, Proximity Sensing Localization in Sensor Network Part II



Part II Outline

- GPS is not always good
- Localization Problem
- Many Ways to approach this problem
  - Trilateration and Triangulation
  - Ad-hoc approaches
  - Optimization
  - Multidimensional Scaling (MDS)
  - Fingerprinting, classification and scene analysis
- Multilateration: Use plane geometry
- Collaborative Multilateration: Use joint optimization
- Mass-spring Localization

Find Where the Sensor is ….

- Location information is important.
  1. Devices need to know where they are.
     - Sensor tasking: turn on the sensor near the window…
  2. We want to know where the data is about.
     - A sensor reading is too hot – where?
  3. It helps infrastructure establishment, such as geographical routing and sensor coverage.
     - Intruder detection;
     - Localized geographical routing.

GPS is not always good

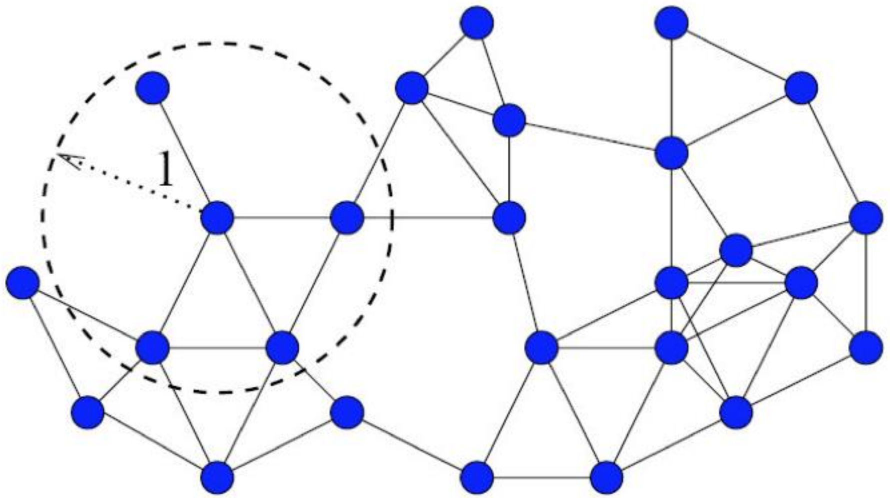
- Requires clear sky, doesn't work indoor.
- Too expensive.
  - A $1 sensor attached with a $100 GPS?

Localization:

- (optional) Some nodes (anchors or beacons) have GPS or know their locations.
- Nodes make local measurements;
  - Distances between two sensors, angles between two neighbors, etc.
- Communicate between each other;
- Infer location information from these measurements.

Model of a sensor network

- Sensor networks with omni-directional antennas are usually modeled by unit disk graphs.
  - Two nodes have a link if and only if their distance is within 1.



- Use the graph property (connectivity, local measurements) to deduct the locations.

Localization Problem

- Output: nodes' location.
  - Global location, e.g., what GPS gives.
  - Relative location.

- Input:
  - Connectivity, hop count.
    - Nodes with k hops away are within Euclidean distance k.
    - Nodes without a link must be at least distance 1 away.
  - Distance measurement of an incoming link.
  - Angle measurement of an incoming link.
  - Combinations of the above.

Measurements

Distance estimation:

- Received Signal Strength Indicator (RSSI)
  - The further away, the weaker the received signal.
  - Mainly used for RF signals.
- Time of Arrival (ToA) or Time Difference of Arrival (TDoA)
  - Signal propagation time translates to distance.
  - RF, acoustic, infrared and ultrasound.

Angle estimation:

- Angle of Arrival (AoA)
  - Determining the direction of propagation of a radio-frequency wave incident on an antenna array.
- Directional Antenna
- Special hardware, e.g., laser transmitter and receivers.

Localization

- Given distances or angle measurements, find the locations of the sensors.

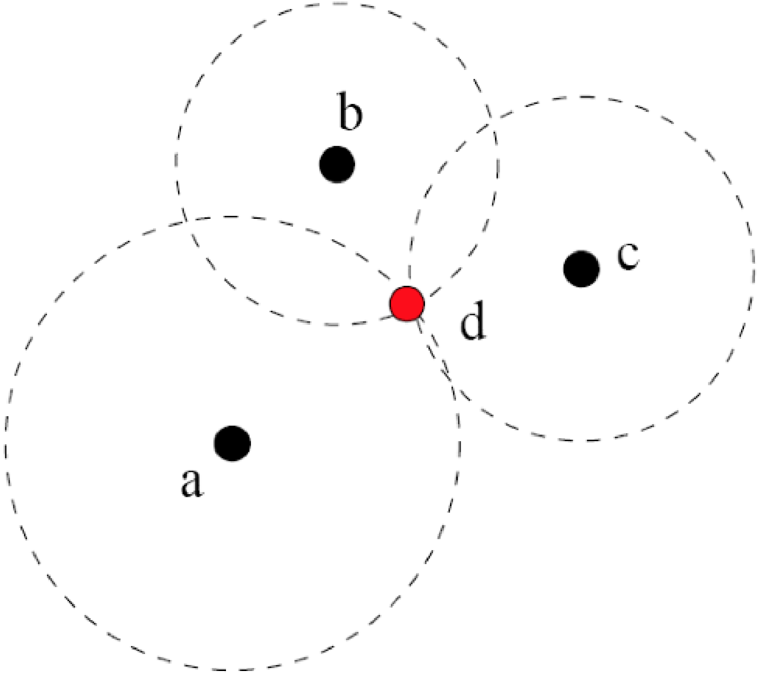
- **Anchor-based**
  - Some nodes know their locations, either by a GPS or as pre-specified.
- **Anchor-free**
  - Relative location only.
  - A harder problem, need to solve the global structure. Nowhere to start.
- **Range-based**
  - Use range information (distance estimation).
- **Range-free**
  - No distance estimation, use connectivity information such as hop count.

Many Ways to approach this problem

- Trilateration and triangulation
- Fingerprinting and classification
- Ad-hoc and range/free
- Graph rigidity
- Identifying codes
- Bayesian Networks
- Optimization
- Multi-dimensional scaling

Trilateration and Triangulation

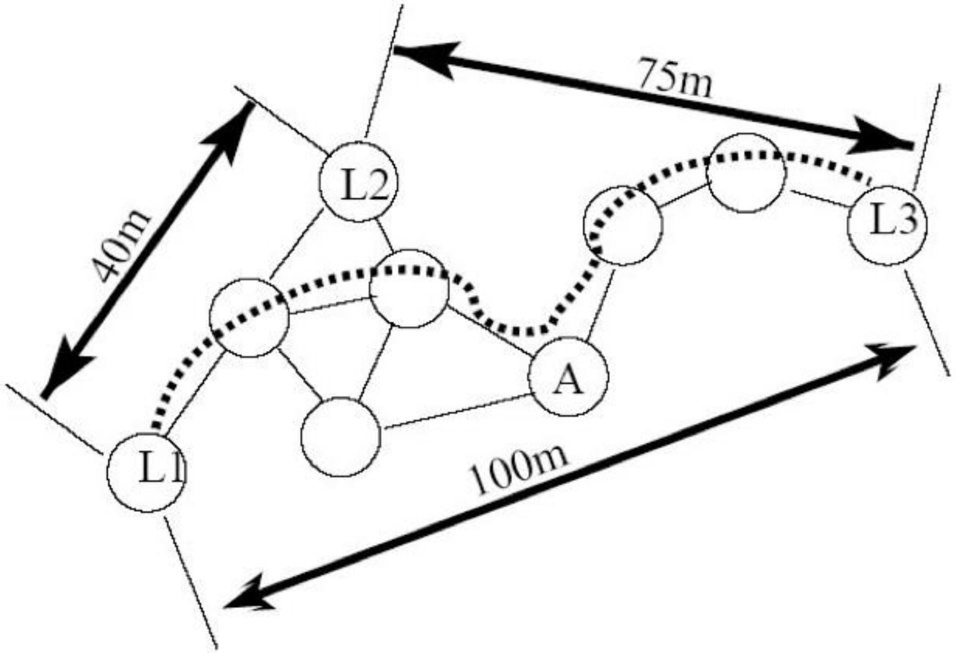
- Use geometry, measure the distances/angles to three anchors.
- Trilateration: use distances
  - Global Positioning System (GPS)
- Triangulation: use angles
  - Cell phone systems.
- How to deal with inaccurate measurements?
- How to solve for more than 3 (inaccurate) measurements?



Stockholms universitet

Ad-hoc approaches

- Ad-hoc positioning (APS)
  - Estimate range to landmarks using hop count or distance summaries
- APS:
  - Count hops between landmarks
  - Find average distance per hop
  - Use multi-lateration to compute location



Optimization

- View system of nodes, distances and angles as a system of equation with unknowns.
- Add inequalities
  - E.g. radio range is at most 1.
- Solve resulting system of inequalities as an optimization problem.

Multidimensional Scaling (MDS)

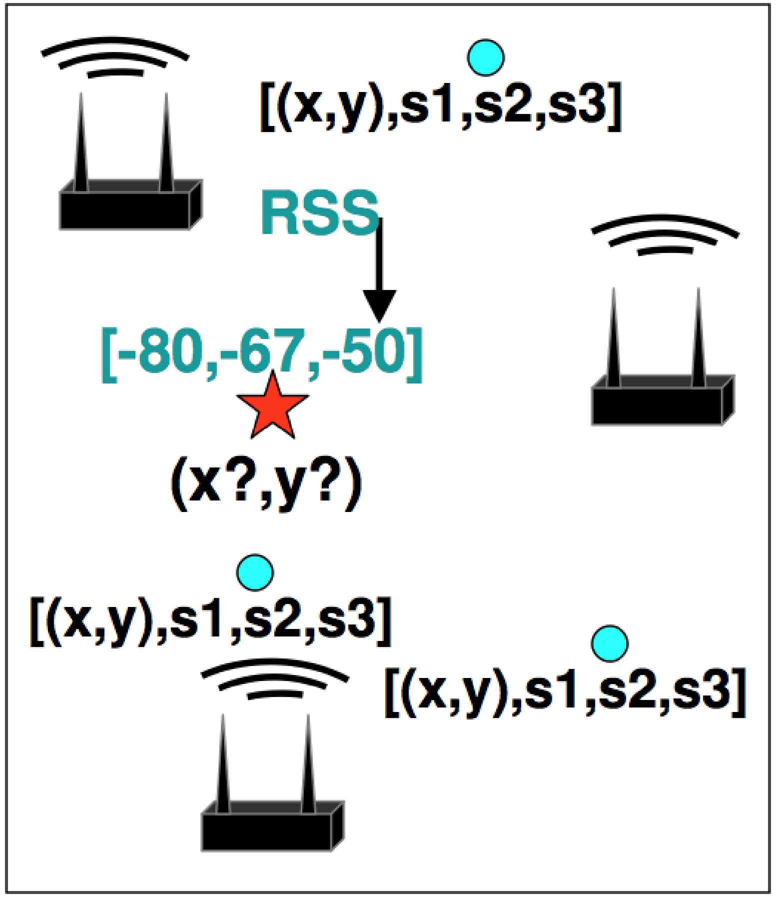
- MDS is a general method of finding points in a geometric space that preserves the pair-wise distances as much as possible.
  - It works also for non-metric data.

- Given the pairwise distances P, find a set of points X in m-dimensional space.

- Take the largest 2 eigenvalues and eigenvectors of X as the best 2D approximations.

Fingerprinting, classification and scene analysis

- Offline phase: collect training data (fingerprints): [(x, y), SS].
  - E.g., the mean Signal Strength to N landmarks.
- Online phase: Match RSS to existing fingerprints probabilistically or by using a distance metric.
- Cons:
  - How to build the map?
    - Someone walks around and samples?
    - Automatic?
  - Sampling rate?
  - Changes in the scene (people moving around in a building) affect the signal strengths.



Stockholms universitet

Bayesian Networks

- View positions as random variables
- Build network to describe likely values of these variables given observations
- Pros:
  - Captures any set of observations and priors
- Cons:
  - Computationally expensive
  - Accuracy

Reference Papers

- **<u>Multi-lateration:</u>**
- **[Savvides01]** A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. Proc. MobiCom 2001.
- **[Savvides03]** A. Savvides, H. Park, and M. B. Strivastava. The *n*-hop multilateration primitive for node localization problems, Mobile Networks and Applications, Volume 8, Issue 4, 443-451, 2003.
- **<u>Mass-spring model:</u>**
- **[Howard01]** A. Howard, M. J. Mataric, and G. Sukhatme, Relaxation on a Mesh: a Formalism for Generalized Localization, IEEE/RSJ Internaltionsl Conference on Intelligent Robots and Systems, October, 2001.