



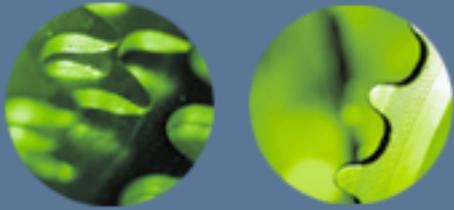
Distributed Moving K-Nearest-Neighbours Query Implementation

Marcin Kwietniewski



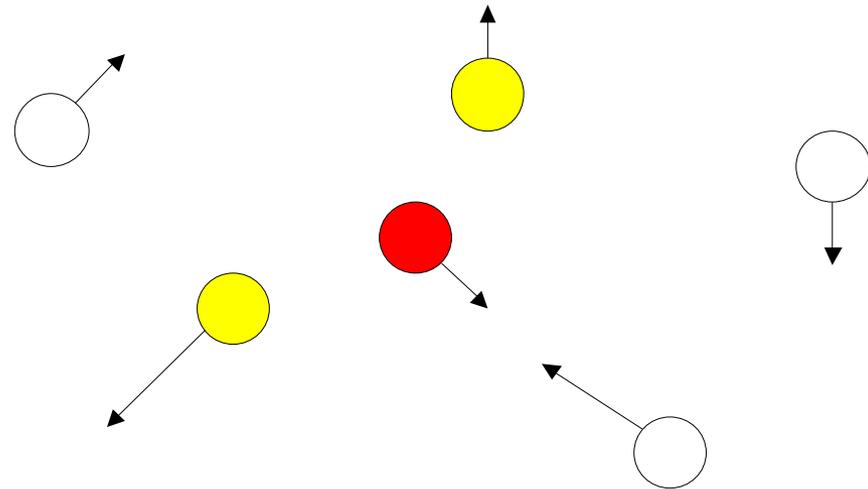
Agenda

- Problem revision
- System overview
- Classes/Threads implementation
- Algorithm & Messaging
- Testing problems
- Q&A



Problem Revision

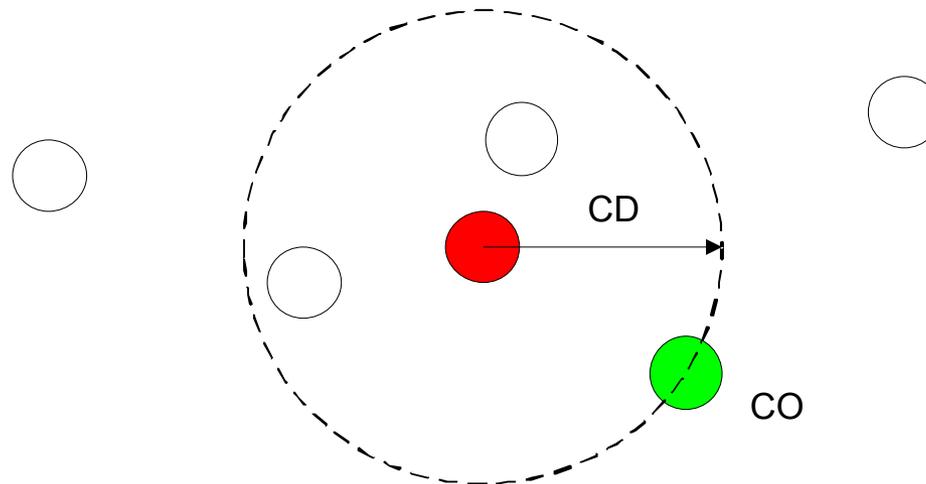
- A mobile network: server, base stations, mobiles
- We are interested in k nearest neighbours of some mobile
- Mobiles are constantly moving





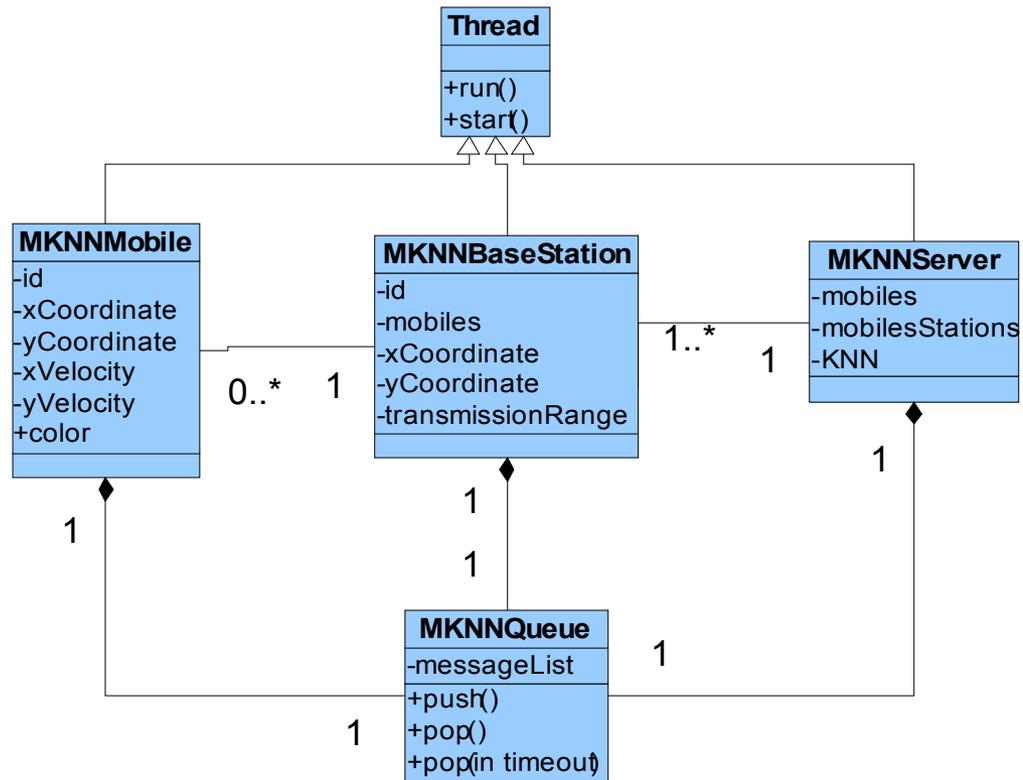
The Algorithm

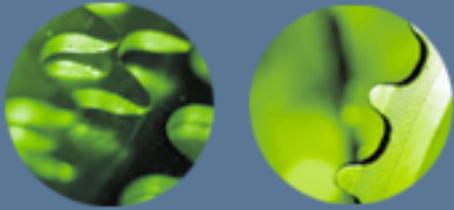
- First phase: we compute the initial result by asking the mobiles about their positions
- Continuous processing: we keep track of positions of the query owner and the critical object





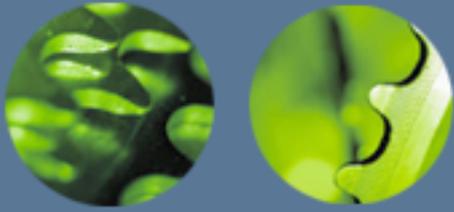
Implementation overview





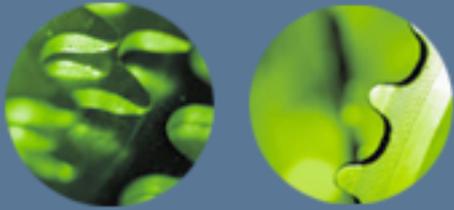
Class implementation

- MKNNBaseStation
 - Checks for messages and forwards them
 - Broadcast option
- MKNNMobile
 - Moves randomly
 - Reports events
 - Does other things (sleeps)
 - Pops messages with timeout
- MKNNServer
 - Performs its tasks only by responding to messages
 - Stores current query state



Message passing

- Each thread with a single incoming message queue
- One reader, multiple writers
- **synchronized** push/pop methods
- Server holds references to base stations' queues
- Base stations – mobiles communication through the MKNNEther



Initial processing

- NEWQUERY message
- Position request broadcast
- Each reply saved – last reply triggers the continuous phase
- Server computes the result and sends it to the query owner



Continuous phase

- Both query owner and the critical object report their positions
- Positions broadcasted to all mobiles
- A mobile checks if it changes the result of the query or becomes the critical object and reports it



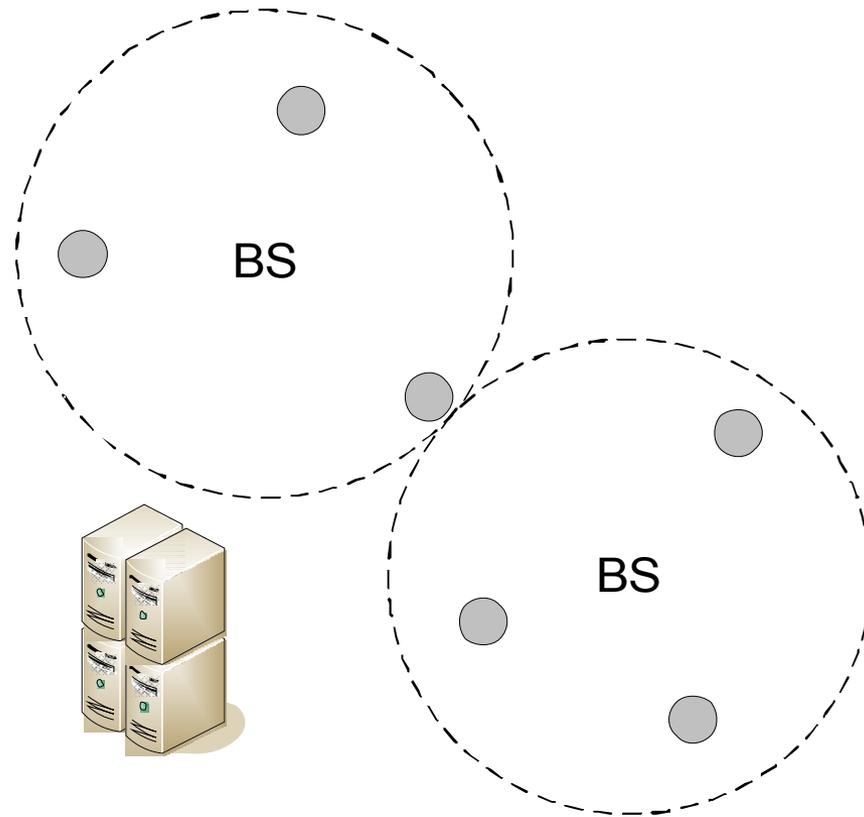
Testing

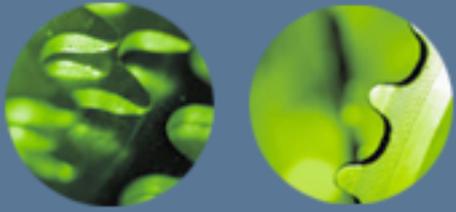
- Problems with defining correct behaviour
- Delays in information flow
- Amount of messages too big for text output analysis
- Animation...
 - Volatile coordinates & colours



Some issues discovered

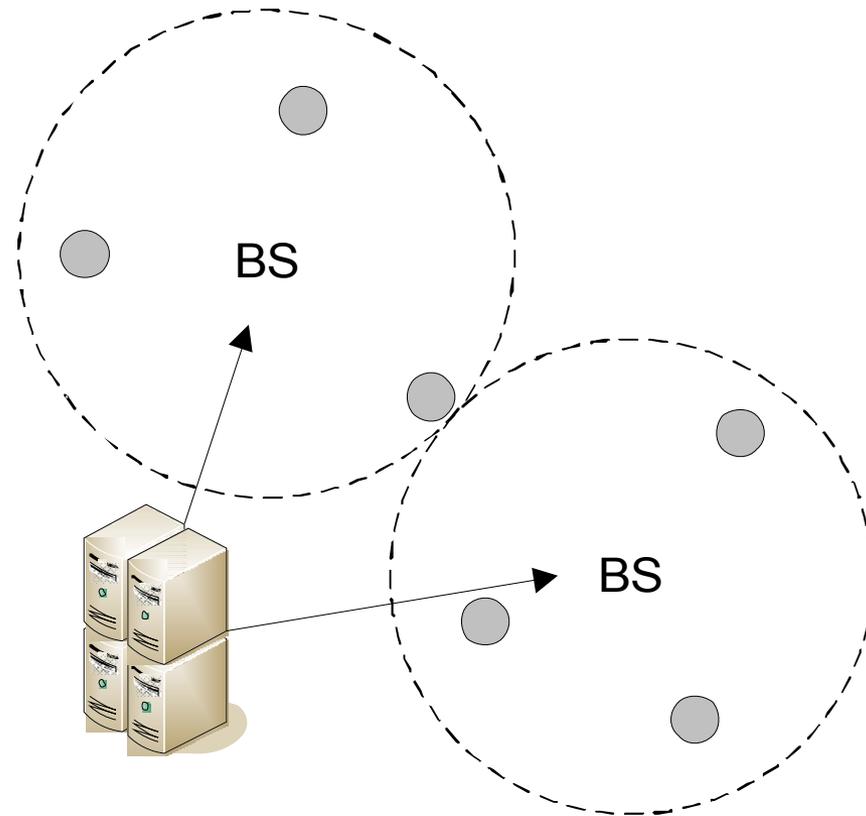
- Broadcast problem:





Some issues discovered

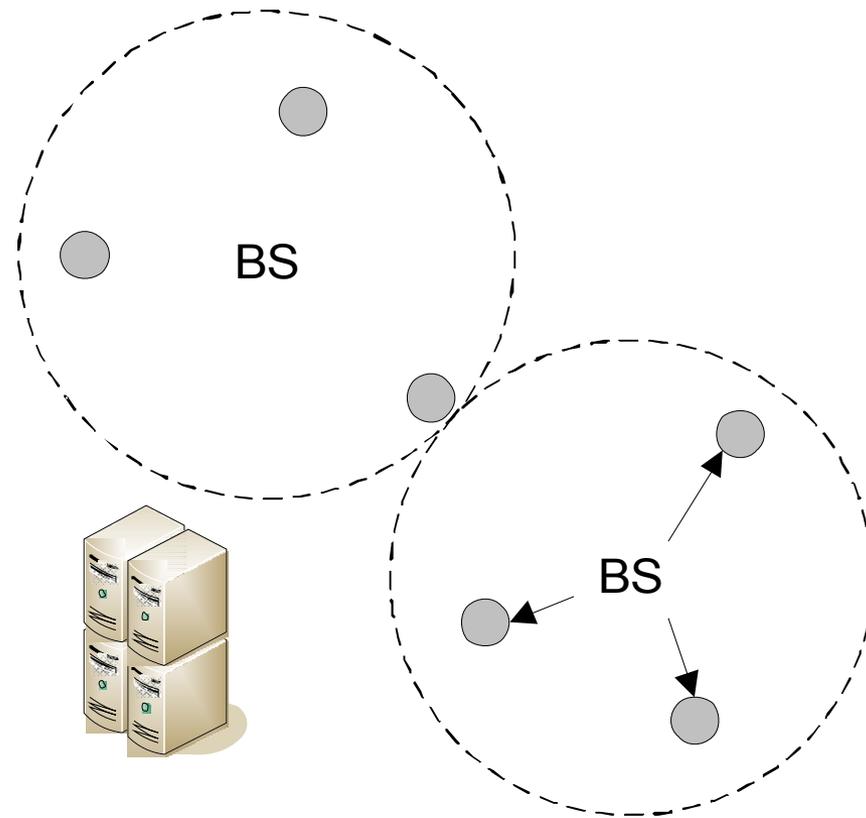
- Broadcast problem:





Some issues discovered

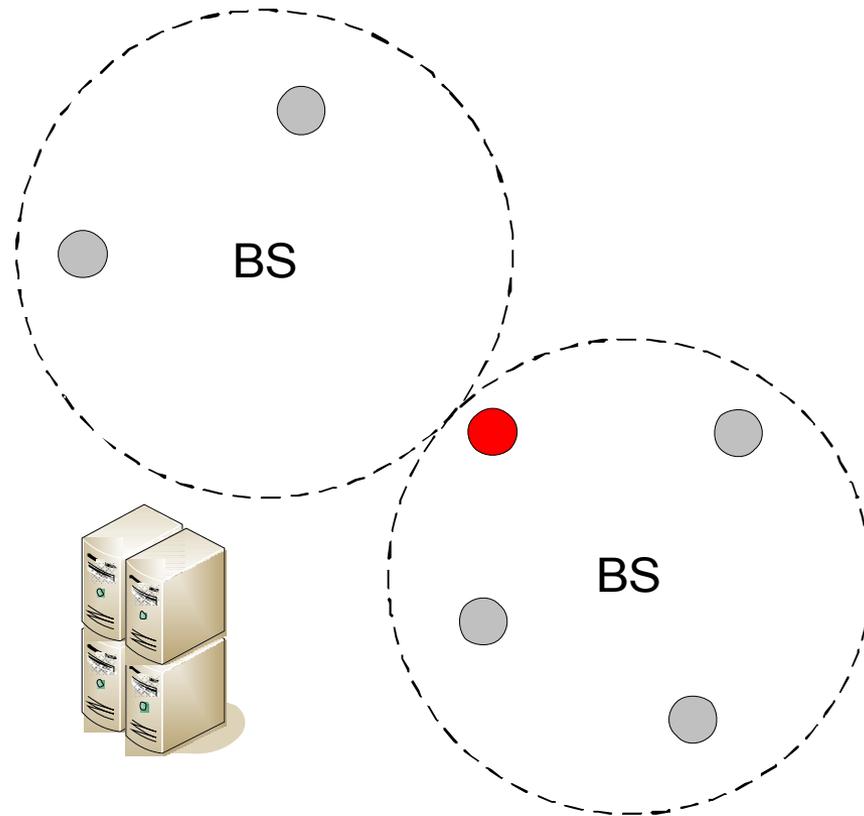
- Broadcast problem:





Some issues discovered

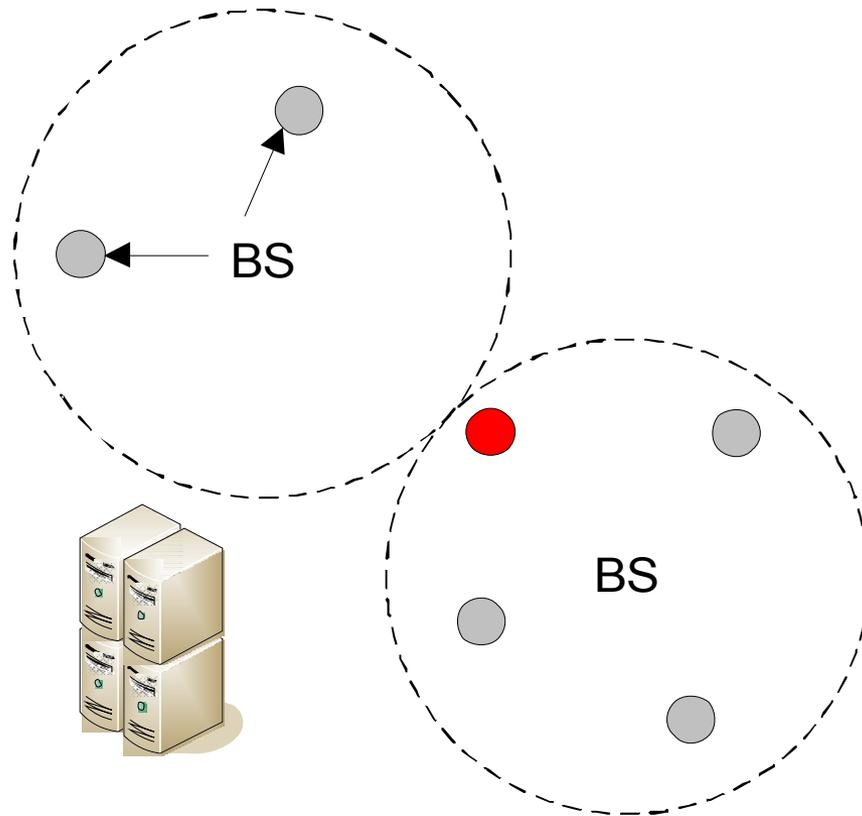
- Broadcast problem:





Some issues discovered

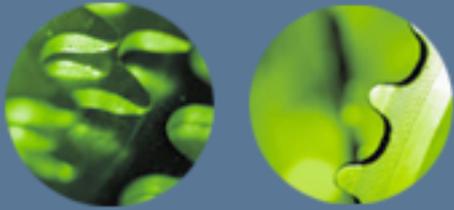
- Broadcast problem:





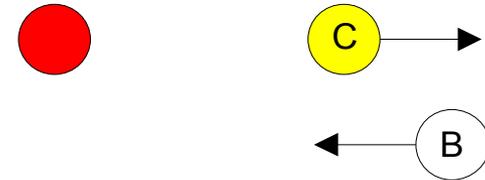
Solution

- Server knows which mobiles are assigned to which stations
- Stations send a summary of a broadcast
- Re-send
- Too many replies are OK.



Problems pt. 2

- State of the computation does not propagate immediately
- A becomes the new critical object
- A replaces C
- B „replaces” C



- Need to check if mobile's info is up to date



Conclusion

- This kind of testing is not enough
- We need to put bounds on information flow delays
- Some automatic verification is needed



Thank you!

Questions?