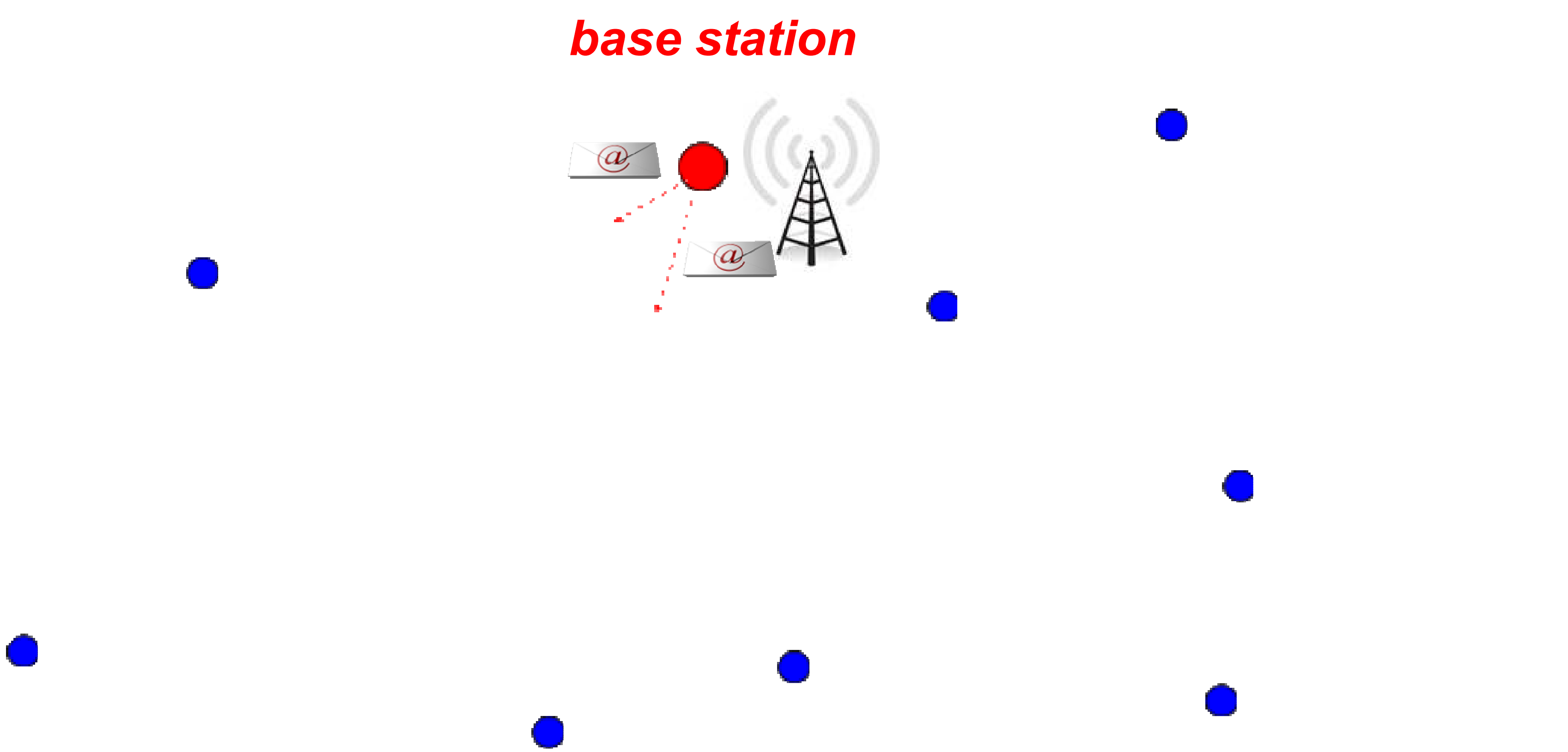
Probabilistic Time Consistent Queries over Moving Objects

Xiang Lian and Lei Chen
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Hong Kong, China
{*xlian*, *leichen*}@*cse.ust.hk*

SSDBM 2011 @ Portland, USA

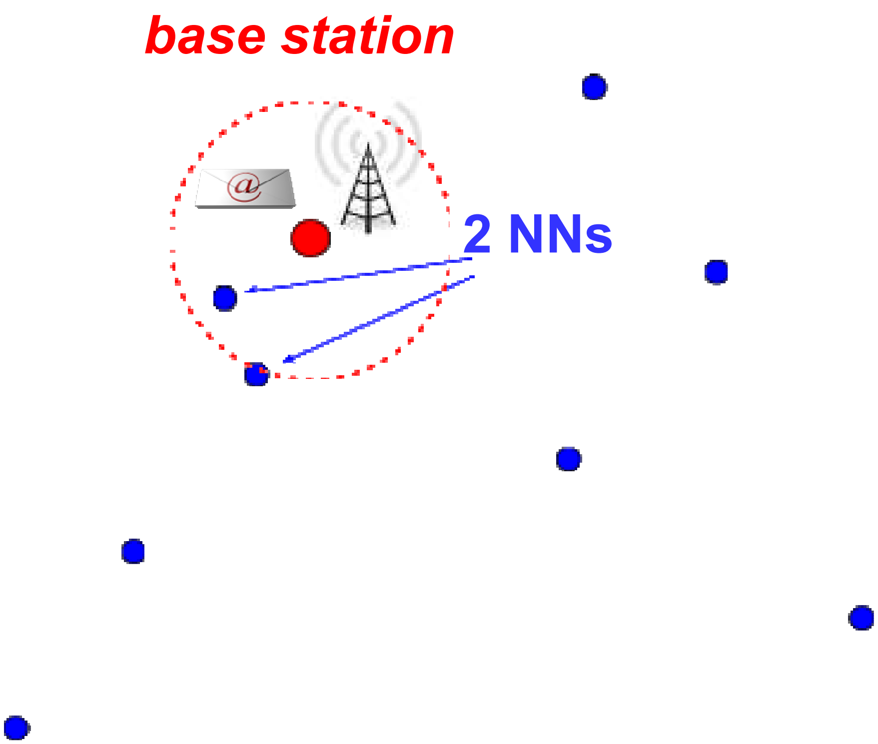
Motivation Example

- Delay Tolerance Network (DTN)



Motivation Example (cont'd)

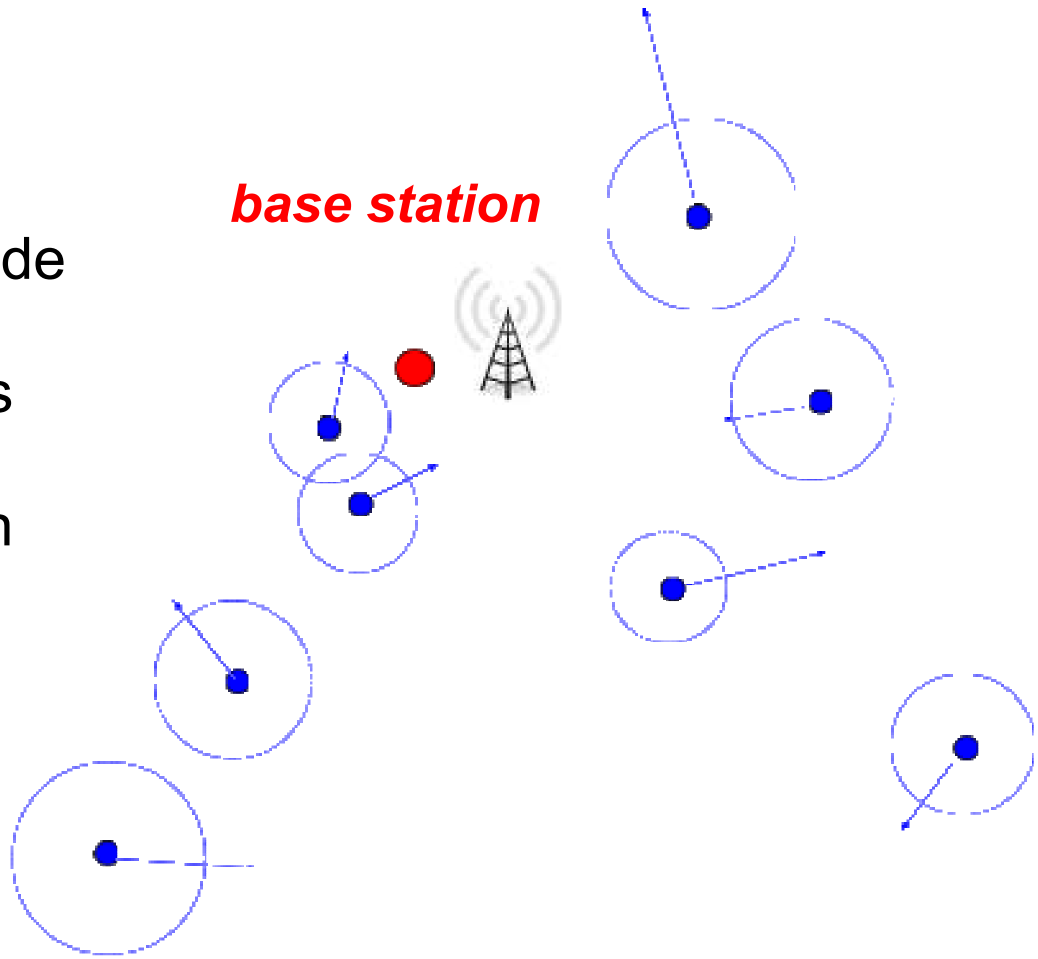
- Delay Tolerance Network (DTN)
  - A base station transmits a message to passing mobile nodes
  - Due to the limited bandwidth, the base station only sends message to its closest mobile nodes
  - It takes some time to send the message (e.g. audios/videos)



- Find those mobile nodes that are *consistently* being *k*-nearest neighbors of a base station for at least *T consecutive* timestamps in a future period [0, *ted*]

Motivation Example (cont'd)

- Delay Tolerance Network (DTN)
  - The location of each mobile node can be imprecise due to the accuracy of positioning devices
  - Conduct *probabilistic time consistent queries* on uncertain moving objects *efficiently* and *effectively*



Outline

- Introduction
- Data Model
- Problem Definition
- The Proposed Approaches
- Experimental Results
- Conclusions

Introduction

- Data uncertainty is ubiquitous in many real-world applications
  - Sensor networks
  - Location-based services
  - Moving object search
  - Privacy preserving
- Reasons
  - Imperfect nature of sensing devices
  - Noises added to data during the data transmission
  - Intentional perturbation for privacy preserving

Uncertainty Model for Uncertain Moving Objects

Definition of Probabilistic Time Consistent Queries

- Probabilistic Time Consistent Queries (PTCQ) on Uncertain Moving Objects
  - An uncertain moving object database, *DU*
  - A query point *q*
  - A time threshold *T*
  - A future period [0, *ted*] (*ed* + 1 □ *T*)
  - A probabilistic threshold *a*
  - A PTCQ query retrieves uncertain objects *o*□ *DU* such that *o* are in the *k*NN set of *q*, P*k*NN(*ti*), for *all* timestamps *ti*□[*tj*, *tj*+*T*-1] □ [0, *ted*], with *k*NN probability greater than or equal to *a* at each timestamp

$$PTCQ(t_i) = \bigcup_{t_j=0}^{t_{ed}-T+1} \left(\bigcap_{t_i=t_j}^{t_j+T-1} PkNN(t_i) \right)$$

Con-*k*NN Definition (cont'd)

- A probabilistic *k*-nearest neighbor (P*k*NN) query retrieves uncertain objects of being *k*-nearest neighbor of a query point *q* with *k*NN probability *PrkNN*(*q*, *o*, *ti*) *a*

$$
\begin{aligned}
& Pr_{kNN}(q, o(t_i)) \\
= \quad & \int_{v_o^-}^{v_o^+} pdf_v(v_o) \cdot \int_{o'(t_i) \in UR(o(t_i))} Pr\{dist(q, o'(t_i)) = r\} \\
& \cdot \sum_{\forall S = \{p_1(t_i), \ldots, p_s(t_i)\} \in \mathcal{D}^U \wedge s < k} \left(\left(\prod_{m=1}^{s} Pr\{dist(q, p_m(t_i)) < r\} \right) \right. \\
& \left. \cdot \left(\prod_{p_n \in \mathcal{D}^U \setminus (S \cup \{o\})} Pr\{dist(q, p_n(t_i)) \geq r\} \right) \right) do'(t_i) dv_o
\end{aligned}
$$

Pruning Heuristics

- Assume we can obtain *k* objects *o*1, *o*2, …, *ok*, close to query point *q*
- In future period [0, *ted*], the maximum distance, *Rmax*, from these *k* objects to *q*, is given by:
  - *Rmax* = *maxj*{*dist*(*q*, *oj*(0)), *dist*(*q*, *oj*(*ted*)) | 1 □ *j* □ *k*}
- Any object intersecting with is a

PTCQ candidate

Pruning Techniques

- Basic idea
  - If an uncertain object *o* of being P*k*NN only last for less than *T* consecutive timestamps in future period [0, *ted*], then object *o* can be safely pruned
- Pruning methods
  - *T*-pruning
  - Period pruning
  - Segment pruning

T-Pruning

- The trajectory of *o* should intersect with circle centered at *q* with radius *Rmax* having length at least *vo* □ *T*

T-pruning condition:

$$||q|| \cdot cos(\eta_q - \gamma) < C_o[x] \cdot cos\gamma + C_o[y] \cdot sin\gamma, \ or$$

$$||q|| \cdot sin(\gamma - \eta_q) - (C_o[x] \cdot sin\gamma - C_o[y] \cdot cos\gamma) > (R_{max} + r_o)^2 - (v_o^- \cdot T)^2/4.$$

Period Pruning – Case 1

- Object *o* is initially outside

period pruning condition:

$$||q|| \cdot cos(\eta_q - \gamma) - (C_o[x] \cdot cos\gamma + C_o[y] \cdot sin\gamma) < \frac{v_o^{+2} \cdot (t_{ed} - T)^2 - (R_{max} + r_o)^2 + dist^2(q, C_o)}{2 \cdot v_o^{+2} \cdot (t_{ed} - T)}$$

Period Pruning – Case 2

- Object *o* is initially intersecting with

period pruning condition:

$$||q|| \cdot cos(\eta_q - \gamma) - (C_o[x] \cdot cos\gamma + C_o[y] \cdot sin\gamma) < \frac{v_o^{-2} \cdot T^2 - (R_{max} + r_o)^2 + dist^2(q, C_o)}{2 \cdot v_o^{-2} \cdot T}$$

Segment Pruning

- Observation: *Rmax* might vary over time

Utilize Velocity and Object Distributions

- Filtering with velocity distributions
  - Replace *vo*- and *vo*+ with *vomin* and *vomax*, respectively, such that:
    , where *b* < *a*

$$\int_{v_{omin}^{1-\beta}}^{v_{omax}^{1-\beta}} pdf_v(v_o)dv_o = 1 - \beta$$

- Fil ributions

Indexing

- *UC-Grid* Index

PTCQ Query Processing

- Find influence region of *q*
  - *Vmax* – global maximum velocity
- For cells intersecting with the influence region
  - Apply pruning methods
- The remaining candidates are refined

Experimental Evaluation

- Experimental Settings
  - Real data: *CA* (California spatial data set)
  - Synthetic data: *lUrU*, *lUrG*, *lSrU*, and *lSrG*
    - Velocity *vo* is randomly generated within [*Vmin*, *Vmax*]
  - Competitor:
    - *Basic* method: first compute P*k*NN answers at each timestamp via *UC-Grid* index, and then combine the P*k*NN results
  - Measures:
    - Filtering time through *UC-Grid* index
    - Speed-up ratio compared with *Basic*

Performance vs. *a*

IU data sets

IS data sets

data size N = 30K, *radius range* [*rmin*, *rmax*] = [0, 0.0005], *T* = 8, [0, *ted*] = [0, 20], [*Vmin*, *Vmax*] = [0, 0.001] , and *k* = 10

Conclusions

- We formulate the problem of probabilistic time consistent query (PTCQ). In particular, we consider PTCQ with probabilistic time consistent *k*NN query
- We propose 3 effective pruning methods and 2 probabilistic filtering approaches to reduce the search space
- We design the grid index for storing uncertain moving objects and present efficient query processing approach
- We demonstrate the efficiency of our proposed query answering approaches

Thank you!

Email: XLIAN@UST.HK

Q/A