

AN INTEREST MANAGEMENT SCHEME FOR MOBILE PEER-TO-PEER SYSTEMS

Ying Li
Richard Fujimoto

Michael Hunter
Wonho Suh

Computational Science and Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

Civil and Environmental Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

ABSTRACT

Interest management is essential for reducing communication overhead by filtering irrelevant messages in mobile distributed systems. Interest management schemes developed for distributed simulation systems such as those based on HLA can be applied to mobile systems. Sort-based schemes have been proposed and shown to yield good performance. When using sort-based schemes in mobile peer-to-peer systems, questions such as the design of the sorting mechanism and where to perform the sorting process must be addressed. This paper proposes an interest management mechanism for mobile peer-to-peer systems, that divides the entire space into cells and uses a bucket sort to sort the regions in each cell. A mobile landmarking scheme is presented to implement this scheme in mobile peer-to-peer systems. The new mechanism is expected to have better computational efficiency for both static and dynamic matching. Experimental results indicate that this approach yields better performance than several alternate interest management schemes.

1 INTRODUCTION

Peer-to-Peer (P2P) architectures have emerged in the past few years and been applied in many applications such as file sharing (Kubiatowicz 2000), distributed computing (Halevy 2003) and P2P networked virtual environments (Knutsson 2004; Yu 2005; Hu 2006). The computation and communication overhead is shared among joining peers in the P2P architecture, so the bottleneck due to centralized entity management is removed. Among P2P systems, mobile P2P systems (Yu 2005) are especially challenging because of its mobile and dynamic nature. One example application is infrastructure-less intelligent transportation systems (Schroth 2005). Vehicles with computing and communication capabilities communicate with each other without centralized entity management and share information during movement. Broadcasting the data will consume far too many network resources. Message filtering schemes, or interest management schemes, must be used to address this data distribution problem.

The objective of interest management is to have all peers only receive data that are of interest to them. The data distribution management services of the High Level Architecture are based on identifying overlaps among pairs of subscription and update regions. Implementation approaches include cell-based mechanisms (grid-based mechanisms) (Macedonia 1995, Daniel 1996) and area-based mechanisms (region-based mechanisms) (Hook 1998, Wood 2002). The cell-based algorithms divide the entire space into a grid of cells and an update region and a subscription region may overlap if they share at least one common grid cell. The overlapping information obtained by cell-based algorithms is not exact and unnecessary communication may be introduced. Area-based algorithms compare pairs of regions. The computational process is straightforward but may incur a significant amount of computational overhead. The hybrid approach (Pan 2010) improves performance by exploiting the advantages of these two approaches.

Sort-based algorithms (Raczy 2005, Liu 2009) have been used to compute the intersection between update and subscription regions. These project the regions along each dimension. The projections are sorted to identify overlaps. However, current sort-based algorithms have certain drawbacks. First, some algorithms statically process all regions in one round and must resort all regions again if any region changes, which can be very time consuming. Second, current sort-based algorithms are based on comparison sorts, such as heapsort or insertion sort, whose computational complexity is $O(n \log n)$. Third, current sort-based algorithms are all run in a centralized fashion where information concerning update and subscription regions are stored. This makes these algorithms poorly suited for peer-to-peer systems.

To address these drawbacks, we propose an interest management scheme for mobile peer-to-peer systems. Our interest management scheme has three advantages over other sort-based algorithms: 1) The algorithm is applicable to dynamic sorting and matching of region modifications. If region modifications occur, the algorithm does not need to resort the list of projections and conduct the entire matching again. It can do the dynamic sorting and matching without processing all of the regions. 2) Our scheme is based on bucket sort. Bucket sort is not a type of comparison sort and the average running time will be linear if not too many projections fall into the same bucket. 3) We introduce the mobile landmarking design to implement this sort-based scheme in mobile peer-to-peer systems. The design is distributed and does not utilize fixed servers. Rather, each peer can become a mobile landmark node to take the server-like role to sort and match regions.

The paper is organized as follows. Section 2 briefly describes major existing interest management schemes. In section 3 we present the proposed interest management mechanism. Experiments and results are presented in section 4. We conclude the paper and describe future work in section 5.

2 RELATED WORK

Existing interest management schemes include area-based, cell-based, hybrid and sort-based approaches. The area-based or region-based scheme (Hook 1998, Wood 2002) uses a brute force approach. It compares all pairs of regions to determine overlaps. The advantage of this approach is its simplicity since the algorithm is straightforward. The main problem of this mechanism is the high computational overhead and poor scalability. When there are many regions but few intersections between regions, a significant amount of overhead will be introduced.

The cell-based approach (Macedonia 1995, Daniel 1996) is also known as the grid-based approach. The entire space is partitioned into a grid of cells. Each region is then mapped to the grid cells. If a subscription region and an update region intersect with the same grid cell, they are assumed to overlap. The advantage of this approach is that the computational overhead is much less than that of the area-based approach. The main drawback of this scheme is that the overlap information is not accurate because entities will receive irrelevant information when their subscription regions cover the same grid cell as the update region, but the two regions do not overlap. The cell size is an important factor in this approach. Some have proposed a method to determine the optimal cell size (Ayani 2000).

The hybrid approach (Tan 2000, Boukerche 2005) combines the area-based and cell-based approaches and improves matching performance. It first uses the cell-based approach to map all the regions to the grid cells. Then the area-based approach is used to make the exact matching between update and subscription regions within each cell. In this approach, the advantages of both the area-based and cell-based approaches are exploited. The overlap information is exact and the matching overhead is much lower than that of the area-based approach.

Sort-based algorithms, e.g., (Raczy 2005, Liu 2009), are used to compute the intersection between update regions and subscription regions. It projects the regions on each dimension and sorts the projections in order to determine the overlap information. After obtaining the overlap information on each dimension, the matching between update regions and subscription regions can be computed because two regions overlap if the projections of the two regions intersect along each dimension. In (Raczy 2005) sort-based algorithms have been shown to yield favorable results compared with the region-based and hybrid algorithms.

3 INFOMATION MANAGEMENT SCHEME

In this section, we present our interest management mechanism for mobile peer-to-peer systems. We illustrate our sorting and matching algorithm first, and then present the mobile landmarking scheme for implementing the sorting and matching algorithm in mobile peer-to-peer systems.

3.1 Sorting and Matching

Our interest management scheme combines the advantages of cell-based and sort-based algorithms. This scheme first divides the multidimensional space into a grid of cells. Then a sort-based algorithm is used within each cell to compute the exact intersection between update and subscription regions. Specifically, our sort-based algorithm projects the regions on each dimension and uses bucket sort to sort the projections in order to find overlaps. This bucket sort-based algorithm is expected to have better computational efficiency than other algorithms based on comparison sorts if there are not many projection values falling into the same bucket. Our algorithm is applicable to dynamic sorting and matching of region modifications without resorting all regions and matching them again.

In the sort-based algorithm, a dimension reduction approach is used to reduce the multidimensional problem to a one-dimensional problem. A pair of regions overlap if all of their projections on each dimension intersect. For example, in Figure 1, the projections of subscription region A and update region B on both the x-axis and y-axis have a non-empty intersection, so regions A and B overlap. The projections of subscription region C and update region D on the y-axis overlap while the projections of C and D on x-axis have no intersection, so regions C and D do not overlap.

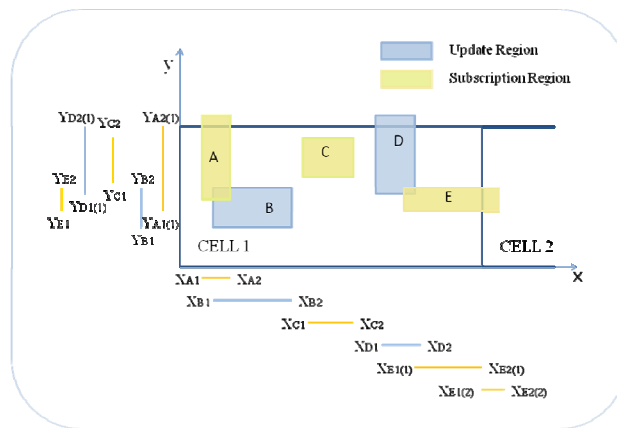


Figure 1: Dimension Reduction Approach

The most important step in the scheme is to sort the projections on each dimension. For the regions residing in multiple cells, we need to record the lower and upper bounds of these regions for each cell. For example, the lower bound of subscription region E is $X_{E1(1)}$ and upper bound of E is $X_{E2(1)}$ in cell 1 for the projection sorting on x-axis, while the lower bound of E is $X_{E1(2)}$ and upper bound of E is $X_{E2(2)}$ in cell 2 ($X_{E2(1)}$ and $X_{E1(2)}$ are cell boundary values).

In our algorithm, we use bucket sort to sort the projections of subscription regions for each dimension. We store the lower bound and upper bound of all the subscription regions in one cell in an array, and then distribute these projections into the buckets.

Bucket i holds values in the half-open interval $[\text{CellBoundValue} + (i/k) * \text{CellLength}, \text{CellBoundValue} + ((i+1)/k) * \text{CellLength})$, where k is the number of buckets and CellBoundValue is the lower bound of each cell. For example, the CellBoundValue is 0 for cell 1, CellLength for cell 2, $2 * \text{CellLength}$ for cell 3 and so on. Suppose CellLength is L and k is chosen to be 10, then bucket 0 for the cell 1 holds values in the interval $[0, 0.1L)$. For the sorting of the elements in each bucket, since there are not many elements in

each one, we can use bucket sort or other sort algorithms, such as insertion sort. The bucket sort-based algorithm works best if the distribution is uniform. In Figure 2(a), we show the bucket sort result for sorting the lower bounds of subscription regions A, C and E in cell 1. The upper bounds can be inserted into buckets similarly and the result is shown in Figure 2(b).

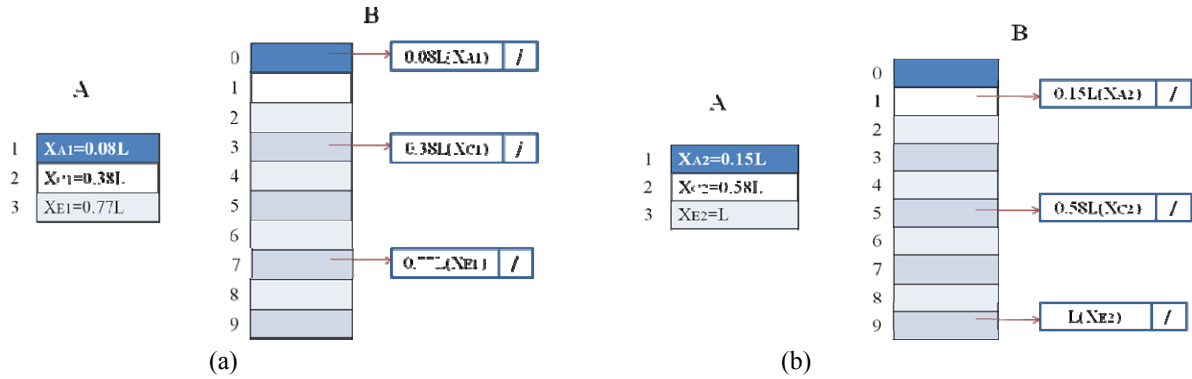


Figure 2(a) and 2(b): Bucket Sort Result for Sorting the Projections of Subscription Regions

After distributing the projections of subscription regions into buckets, we need to process the update regions. If an update region does not overlap with a subscription region, there are only two possibilities:

- (1) The lower and upper bounds of the subscription region are less than this update region, which means the upper bound of the subscription region must be less than the lower bound of the update region.
- (2) The lower bound and the upper bound of the subscription region are greater than this update region, which means the lower bound of the subscription region must be greater than the upper bound of the update region.

Based on case (1), we first determine the bucket index if we insert an update region lower bound into the buckets used to sort subscription region upper bounds. Then all the subscription regions whose upper bounds are in the bucket preceding this bucket do not overlap with the update region; all the subscription regions whose upper bounds are in the same bucket but less than this update region lower bound do not overlap with this update region. Similarly, we can also determine the bucket index if we insert an update region upper bound into the buckets used to sort subscription region lower bounds. Then all the subscription regions whose lower bounds are in the bucket after this bucket do not overlap with the update region; all the subscription regions whose lower bounds are in the same bucket but more than this update region upper bound do not overlap with the update region.

Consider the above example illustrated in Figure 3. Suppose $X_{B1} = 0.12L$, $X_{B2} = 0.36L$, $X_{D1} = 0.62L$, $X_{D2} = 0.79L$. X_{C1} and X_{E1} (the lower bounds of subscription regions C and E) are more than X_{B2} (the upper bound of update region of B), so B does not overlap with C and E on the x-axis, as shown in Figure 3(a). Similarly in Figure 3(b), X_{A2} and X_{C2} (the upper bounds of subscription regions A and C) are less than X_{D1} (the lower bound of update region of D), so D does not overlap with A and C on the x-axis.

After we have determined if there are overlaps for each dimension, we can combine them to determine overlaps for the multidimensional space. The algorithm for static matching of all regions is shown in Figure 4. As mentioned before, some sort-based algorithms cannot deal with dynamic matching of a region modification without repeating the static matching of all regions. Here, if the modified region R is a subscription region, the original overlap information related to R and the projections of R are removed. Then the scheme inserts the new bounds of R into the list of the corresponding sorting buckets. Then the new overlap information is obtained by comparing the lower bound of each update region with the upper bound of R as well as comparing the upper bound of each update region with the lower bound of R. If the modified region R is an update region, the original overlap information related to R is removed. Then the scheme determines the overlap information between R's new projections with the bounds of each subscription region. The algorithm for dynamic matching of one region modification is shown in Figure 5.

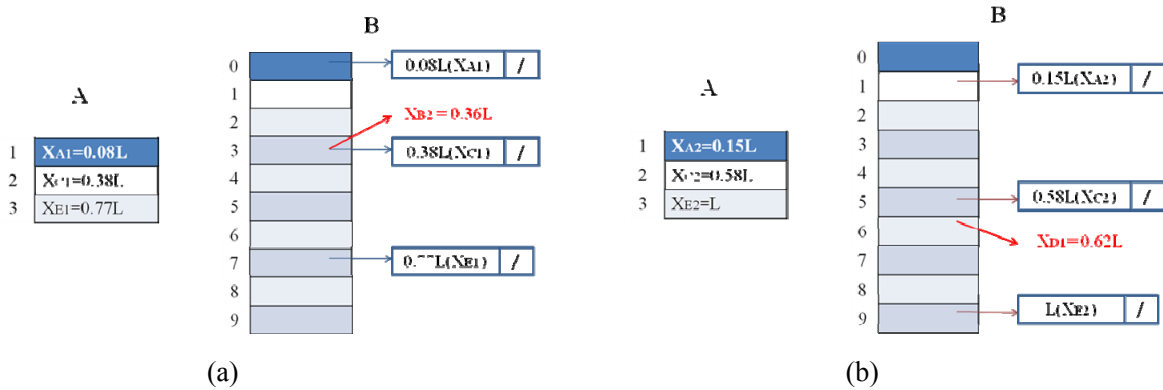


Figure 3(a) and 3(b): Determining Overlap Information

3.2 Mobile Landmarking

When dividing the entire space into cells and sorting within each cell in a mobile peer-to-peer system, one important question concerns where the sorting and matching computation is performed. Conventional landmarking technology (Ratnasamy 2002) suffers from the limitation that it assumes a set of fixed, stationary landmark nodes. All entities are expected to know the landmark nodes and communicate with other entities through the landmark nodes. This requires a server, but in mobile peer-to-peer systems there are usually no sets of fixed nodes available. Therefore, we introduce the mobile landmarking concept into mobile peer-to-peer systems.

This scheme works without any fixed landmark nodes. Instead, it uses a set of landmark keys. For example, a landmark key can be defined as the center point of a cell. Rather than having dedicated landmark nodes, those nodes which are currently closest to one of the landmark keys become mobile landmark nodes. When one of the current mobile landmark nodes fails or resigns, another node (which is now closest to the landmark key) will automatically become the new mobile landmark node.

An example is shown in Figure 6. The entire environment is partitioned into sixteen cells and each landmark key is responsible for each cell. The node which is currently closest to one of the landmark keys becomes a mobile landmark node. In cell 16, the current mobile landmark node resigns and another node which is now closest to the landmark key becomes the new mobile landmark node automatically.

At the initialization stage, each entity will send a message about their subscription and update regions to the mobile landmark node responsible for the cell in which the entity resides. The data format of the message is (EntityID, EntityLocation, URLowerBound, URUpperBound, SRLowerBound, SRUpperBound), where URLowerBound refers to the lower bound of the update region, URUpperBound refers to the upper bound of the update region, SRLowerBound refers to the lower bound of the subscription region, and SRUpperBound refers to the upper bound of the subscription region. Then the mobile landmark node performs the static sorting and matching. The mobile landmark node then sends a message to each entity in the cell about other entities whose subscription regions overlap with this entity's update region. The data format of the message is (EntityID1, EntityLocation1, EntityID2, EntityLocation2, EntityID3, EntityLocation3,...). When entities receive the information, they will transmit state update messages to these entities listed in the message. In this scheme, we assume all the communications are reliable.

When there is update or subscription region modification, the entity will send the modification information to the mobile landmark node responsible for the cell in which the entity resides. However, it will incur too much communication overhead if an entity sends the updates each frame. We can have each entity send the modification information periodically or in a space-driven manner, where an entity sends the region modification information to the mobile landmark node only when its position shifting from the last

```

1. for each subscription region
2.   for each dimension
3.   {
4.     insert its lower bound into SRegionLowerBoundSet;
5.     insert its upper bound into SRegionUpperBoundSet;
6.   }
7. for each update region
8.   for each dimension
9.   {
10.    insert its lower bound into URegionLowerBoundSet;
11.    insert its upper bound into URegionUpperBoundSet;
12.   }
13. for each dimension
14. {
15.  insert each element in SRegionLowerBoundSet into the list of the corresponding bucket for
    lower bound sorting;
16.  insert each element in SRegionUpperBoundSet into the list of the corresponding bucket for up-
    per bound sorting;
17.  for each bucket
18.    sort the list with insertion sort;
19.  }
20. for each dimension
21. {
22.  for each update region lower bound in URegionLowerBoundSet
23.  {
24.    determine the bucket index if this update region lower bound is inserted into the buckets
      used to sort subscription region upper bounds;
25.    all the subscription regions whose upper bounds are in the bucket before this bucket do not
      overlap with this update region;
26.    all the subscription regions whose upper bounds are in the same bucket but less than this
      update region lower bound do not overlap with this update region;
27.  }
28. }
29. for each dimension
30. {
31.  for each update region upper bound in URegionUpperBoundSet
32.  {
33.    determine the bucket index if this update region upper bound is inserted into the buckets used
      to sort subscription region lower bounds;
34.    all the subscription regions whose lower bounds are in the bucket after this bucket do not
      overlap with this update region;
35.    all the subscription regions whose lower bounds are in the same bucket but more than this
      update region upper bound do not overlap with this update region;
34.  }
35. }

```

Figure 4: Algorithm for Static Matching of All Regions

```

1. if (R is a subscription region)
2. {
3.   for each dimension
4.   {
5.     remove the overlap information related to subscription region R;
6.     remove the original lower bound and original upper bound of R in the list of the corresponding
       sorting bucket;
7.     insert the new lower bound and new upper bound of R in the list of the corresponding sorting
       bucket;
8.   }
9.   for each dimension
10.  for each update region
11.  {
12.    if the lower bound of the update region is more than the upper bound of R,
13.    this update region does not overlap with R;
14.    if the upper bound of the update region is less than the lower bound of R,
15.    this update region does not overlap with R;
16.  }
17. }
18. else           // if R is an update region
19. {
20.  for each dimension
21.  remove the overlap information related to update region R;
22.  for each dimension
23.  {
24.    determine the bucket index if the lower bound of R is inserted into the buckets used to sort
       subscription region upper bounds;
25.    all the subscription regions whose upper bounds are in the bucket before this bucket do not
       overlap with R;
26.    all the subscription regions whose upper bounds are in the same bucket but less than this
       update region lower bound do not overlap with R;
27.  }
28.  for each dimension
29.  {
30.    determine the bucket index if the upper bound of R is inserted into the buckets used to sort
       subscription region lower bounds;
31.    all the subscription regions whose lower bounds are in the bucket after this bucket do not
       overlap with R;
32.    all the subscription regions whose lower bounds are in the same bucket but more than this
       update region upper bound do not overlap with R;
33.  }
34. }

```

Figure 5: Algorithm for Dynamic Matching of One Region Modification

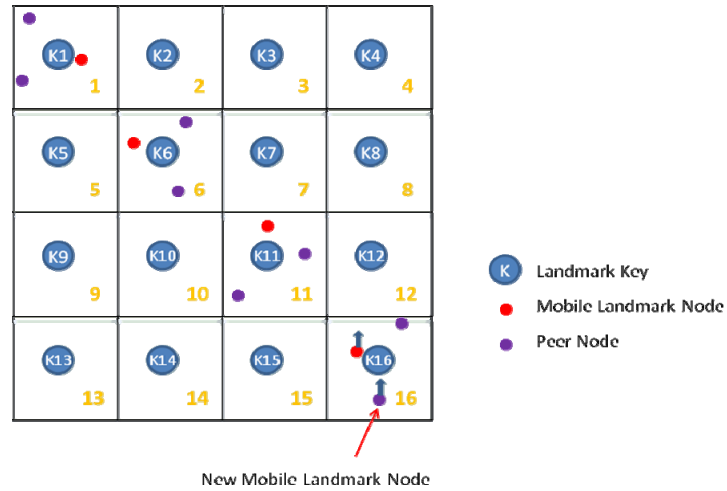


Figure 6: Mobile Landmarking

update exceeds DistanceThreshold in either dimension. The data format of the message is (EntityID, EntityLocation, OldURLowerBound, OldURUpperBound, OldSRLowerBound, OldSRUpperBound, NewURLowerBound, NewURUpperBound, NewSRLowerBound, NewSRUpperBound).

After the mobile landmark node receives the message, it will perform the dynamic matching of this region modification. If the modified region is an update region, the mobile landmark node will send back a message to the entity about the subscription regions that overlap with either the original update region or the new update region but not both. The data format is (EntityID₁, EntityLocation₁, Removed, EntityID₂, EntityLocation₂, Removed, ..., EntityID_k, EntityLocation_k, Added, ...). The removed entity refers to the entity whose subscription region overlaps with the original update region while the added entity refers to the entity whose subscription region intersects with the new update region. Then the entity will transmit its update information to the overlapped subscription regions. If the modified region is a subscription region, the mobile landmark node will record the information of update regions that overlap with either the original subscription region or the new subscription region but not both and send the information to the entities to which these update regions belong. The data format of the message to each update region is (EntityID, EntityLocation, Removed) if the update region only overlaps with the original subscription region or (EntityID, EntityLocation, Added) if the update region only intersects with the new subscription region. Then the affected entities will send the update information based on the new overlap information. The algorithm for mobile landmark node is shown in Figure 7.

Since peer nodes communicate with mobile landmark nodes all the time, they will have knowledge of the mobile landmark nodes. For that purpose, each mobile landmark node sends out beacons periodically within the cell in which this mobile landmark node resides. Whenever a node overhears a beacon, it stores the information of this mobile landmark node. Nodes periodically examine whether they have moved closer to a new mobile landmark node, for example, when they have moved into a new cell.

4 EXPERIMENTAL EVALUATION

To evaluate the performance of the proposed interest management mechanism, a mobile peer-to-peer transportation system is simulated as a routing agent in the ns2 simulator. All simulations that we carried out modeled wireless networks over the course of one (simulated) hour. The simulated system is a two-dimensional space of 1000 distance units by 1000 distance units. Its input is the traces obtained from VISSIM (PTV Vision VISSIM), a microscopic multi-modal transportation simulator. The simulated traffic network has 10*10 intersections with each intersection under two-phase signal control. The cycle length of the traffic signal is 120 seconds. Vehicles entering an intersection will maintain a straight path


```

At Initialization:
1. Receive information from peers in the cell about the subscription and update
   Regions;
2. Sort all the regions and perform the static matching;
3. Send a message to each peer in the cell about the overlap information;

After One Region Modification:
4. Receive region modification information from the peer in the cell;
5. if (the modified region is an update region)
6. {
7.     Perform the dynamic sorting and matching for update region modification;
8.     Send back a message to the peer about the subscription regions who overlap with
       either the original update region or the new update region but not both;
9. }
10. else // the modified region is a subscription region
11. {
12.     Perform the dynamic sorting and matching for subscription region modification;
13.     Send a message to each peer whose update region overlap with either the original
       subscription region or the new subscription region but not both;
14. }
    
```

Figure 7: The Algorithm for Mobile Landmark Nodes

with 95% probability, or turn left with 2% probability or turn right with 3% probability. Each roadway is a two-way arterial, with one lane in each direction. Vehicles enter this network from each entrance (in total 40 entrances) at initialization.

In this simulated mobile peer-to-peer transportation system, vehicles communicate with each other without infrastructure. Each vehicle is associated with a update region and a subscription region. Our scheme is used to obtain the overlap information between update regions and subscription regions in order that each vehicle only receives the information of interest. We conducted three sets of experiments to evaluate our scheme: Comparison of communication cost with other schemes; Comparison of execution time of static matching using different schemes; Comparison of execution time of dynamic matching under region modifications with other schemes. For the first set of experiments, the range per dimension is randomly generated with a uniform probability density function between 50 distance units and 150 distance units. We use the space-driven manner when sending the region modification information and the DistanceThreshold is chosen to be 10 distance units here. For the second and third set of experiments, the region definition is related to overlapping rate, which will be defined below. All our experiments run on a single PC with Intel (R) Core(TM) 2 Duo CPU P7550 2.26GHz, 2GB RAM and Windows XP OS.

4.1 Comparison of Communication Cost

First, we evaluate the communication cost of our scheme and compare it with another three schemes: the region-based, cell-based and hybrid schemes. The communication cost refers to the number of messages that are sent by all vehicles at each frame. The grid cell size for cell-based scheme is 100 distance units in this set of experiments.

As is shown in Figure 8, all the communication cost increases linearly with an increasing number of vehicles, as expected. The cell-based scheme has the highest communication cost. The reason is that the overlapping information computed by the cell-based algorithm is not exact and this lack of precision creates unnecessary data communication. A relatively large grid cell size also causes more irrelevant data to be received by entities. The hybrid scheme has lower communication cost than cell-based and region-based schemes. That is because the hybrid scheme reduces the communication cost in the region-based mechanism by utilizing the cell-based mechanism. The figure also shows that the communication cost of

the proposed scheme is slightly higher than that of the hybrid approach, which suggests that the mobile landmarking design does not introduce an excessive amount of communication overhead in order to improve the computation performance. Although the mobile landmark node periodically sends beacons, the beacon message is restricted to be transmitted within the cell rather than the entire system. Therefore, the communication cost is only slightly higher than that of the hybrid approach.

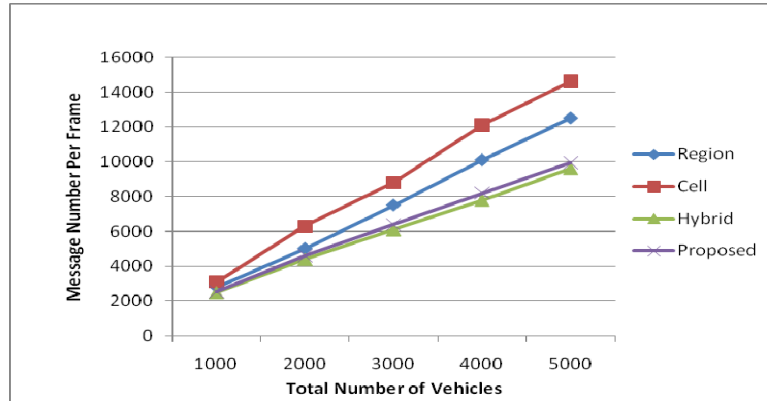


Figure 8: Comparison of Message Number

4.2 Comparison of Execution Time of Static Matching

In this set of experiments, each scheme (region, hybrid, sort and our proposed scheme) performs static matching of all regions and the execution time of static matching is recorded. One of the important factors affecting the performance is the overlapping rate, which is defined as follows.

$$Overlapping\ rate = \frac{\sum area\ of\ regions}{area\ of\ the\ entire\ space}$$

If the space is 1000*1000, and one region is 10*10, where the number of regions is fixed at 1000, the overlapping rate is

$$0.1 = \frac{1000 * (10 * 10)}{1000 * 1000}$$

The higher overlapping rate implies greater probability of region overlap. Two different values 0.01 and 1 are used in the experiments.

Figure 9(a) shows the execution time of three schemes with the overlapping rate set to be 0.01. The hybrid and the proposed algorithms have similar performance, while the region-based approach yields very poor results, especially when the number of vehicles is increased. Both the proposed algorithm and the hybrid algorithm with grid cell size 100 distance units perform well because the regions are very small and usually they do not cover more than one cell when the overlapping rate is 0.01. The execution time of the proposed algorithm is half of that of the hybrid approach. Hence the proposed scheme is the best choice among the three schemes in this scenario.

Figure 9(b) shows the experiment results when the overlapping rate is 1. Our proposed algorithm also performs best with higher overlapping rate because it is less affected by the overlapping rate. The hybrid algorithm with grid cell size 100 distance units has high overhead when the regions become large and covering more than one cell. The region-based algorithm still shows poor results because it exhaustively compares all regions to determine the overlap information.

4.3 Comparison of Execution Time of Dynamic Matching

For the dynamic matching, a random subscription or update region is modified in one iteration and each scheme (region-based algorithm, Raczy’s sort-based algorithm (Raczy 2005) and the proposed algorithm)

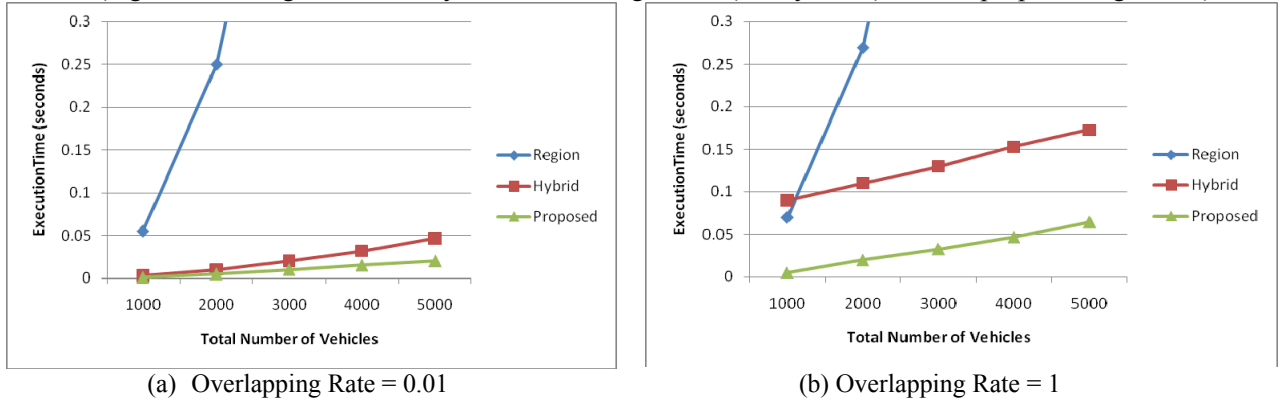


Figure 9: Comparison of Execution Time of Static Matching

performs dynamic matching of a region modification for 100 iterations and calculates the average performance of the 100 iterations. Figure 10 shows that the dynamic matching time of region-based and Raczy’s sort-based algorithm is almost the same as the static matching time of all regions. This is because they cannot perform dynamic matching without processing all the regions again. The dynamic matching time of the proposed algorithm is almost zero no matter whether the overlapping degree is 0.01 or 1 because our proposed algorithm only removes the overlap information related to original region bounds and obtains the overlap information related to new region bounds without resorting all the regions.

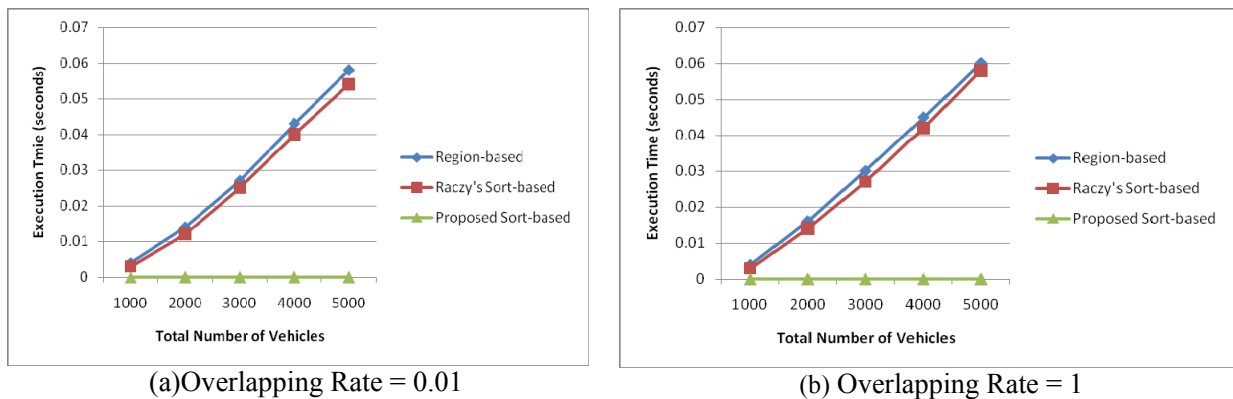


Figure 10: Comparison of Execution Time of Dynamic Matching

5 CONCLUSION AND FUTURE WORK

In this paper, we have proposed an interest management scheme for mobile peer-to-peer systems, that divides the entire space into cells and uses bucket sort to sort the regions in each cell. The interest management scheme is applicable to dynamic sorting and matching of region modifications. If region modifications occur, the algorithm does not need to re-sort the projections and conduct the entire matching again. The bucket sort-based scheme is expected to have better computational efficiency than other algorithms based on comparison sorts. We also introduce the mobile landmarking design to implement this sort-based scheme in mobile peer-to-peer system simulation. The design does not have servers, but every peer can become a mobile landmark node to take the server-like role to sort and match the regions. To evaluate

the new scheme, a mobile peer-to-peer 10*10 transportation system is simulated. The performance results show that the scheme has better computational efficiency for both static matching and dynamic matching. Furthermore, it is also shown that the mobile landmarking design does not introduce a lot of communication overhead in order to improve the computation performance.

For future work, we plan to build an analytical model for the proposed scheme in order to analyze its performance theoretically. We will also implement the scheme in parallel with CUDA to improve the performance. We plan to carry out more experiments to evaluate the proposed scheme under different bucket sizes, cell sizes, entity movements and occupation densities.

REFERENCES

- Ayani, R., F. Moradi, and G. Tan. 2000. "Optimizing cell-size in grid-based DDM," *Proceedings of the Fourteenth Workshop on Parallel and Distributed Simulation*.
- Boukerche, A., N. J. McGraw, C. Dzermajko, and K. Lu. 2005. "Grid-Filtered Region-Based Data Distribution Management in Large-Scale Distributed Simulation Systems," *Annual Simulation Symposium*.
- Daniel, S. R., and D. J. V. Hook. 1996. "Evaluation of grid-based relevance filtering for multicast group assignment," *Proceedings of the Distributed Interactive Simulation*.
- Halevy, A. Y., Z. G. Ives, P. Mork, and I. Tatarinov. 2003. "Piazza: data management infrastructure for semantic web applications," *World Wide Web Conference*.
- Hook, D. V., and J. Calvin. 1998. "Data distribution management in rti 1.3," *Proceedings of the 1998 Spring Simulation Interoperability Workshop*.
- Hu, S. Y., J. F. Chen, and T. H. Chen. 2006. "Von: A scalable peer-to-peer network for virtual environments," *IEEE Network*.
- Knutsson, B., H. Lu, W. Xu, and B. Hopkins. 2004. "Peer-to-peer support for massively multiplayer games," *Proceedings of IEEE INFOCOM*.
- Kubiatowicz, J., D. Bindel, Y. Chen, S. E. Czerwinski, P. R. Eaton, D. Geels, R. Gummadi, S. C. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Y. Zhao. 2000. "Oceanstore: An architecture for global-scale persistent storage," *International Conference on Architecture Support for Programming Languages and Operating System*.
- Liu, E. S., and G. K. Theodoropoulos. 2009. "An Approach for Parallel Interest Matching in Distributed Virtual Environments," *Proceedings of the 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2009)*.
- Macedonia, M. R., M. J. Zyda, D. R. Pratt, D. P. Brutzman, and P. T. Barham. 1995. "Exploiting reality with multicast groups," *IEEE Comput. Graph.*
- Pan, K., S. J. Turner, W. Cai, and Z. Li. 2007. "An Efficient Sort-Based DDM Matching Algorithm for HLA Applications with a Large Spatial Environment," *ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation*.
- Pan, K., W. Cai, X. Tang, S. Zhou, and S. Turner. 2010. "A hybrid interest management mechanism for peer-to-peer networked virtual environments," *IEEE International Symposium on Parallel and Distributed Processing (IPDPS)*.
- PTV Vision VISSIM-Multi-Modal Traffic Flow Modeling. <http://www.ptvag.com/software/transportation-planning-traffic-engineering/software-system-solutions/vissim/>
- Raczy, C., G. Tan, and J. Yu. 2005. "A sort-based ddm matching algorithm for hla," *ACM Trans. Model. Comput. Simul.*, 15 (1): 14–38.
- Ratnasamy, S., M. Handley, R. Karp, and S. Shenker. 2002. Topologically-Aware Overlay Construction and Server Selection, *IEEE Infocom '2002*.
- Schroth, C., F. Dotzer, T. Kosch, B. Ostermaier, and M. Strassberger. 2005. "Simulating the traffic effects of vehicle-to-vehicle messaging systems," *Proc. of the 5th International Conference on ITS Telecommunications*, Brest, France.
- Tan, G. S. H., Y. Zhang, and R. Ayani. 2000. "A Hybrid Approach to Data Distribution Management," *4th International Workshop on Distributed Simulation and Real-Time Applications (DS-RT 2000)*.

- Wood, D. D. 2002. "Implementation of DDM in the MAK High Performance RTI," *Proceedings of the 2002 Spring Simulation Interoperability Workshop*.
- Yu, A. P., and S. T. Vuong. 2005. "Mopar: a mobile peer-to-peer overlay architecture for interest management of massively multiplayer online games," *Proceedings of the international Workshop on Network and Operating Systems Support for Digital Audio and Video*. New York, NY.

AUTHOR BIOGRAPHIES

YING LI is a Ph.D. candidate in Computational Science and Engineering at the Georgia Institute of Technology. Her current research interest is in modeling and simulation of mobile peer-to-peer systems and intelligent transportation systems. Her email address is yingli@cc.gatech.edu.

RICHARD FUJIMOTO is a Professor and Chair of the Computational Science and Engineering Division at the Georgia Institute of Technology. He received his M.S. and Ph.D. degrees from the University of California (Berkeley) in 1980 and 1983. He lead the definition of the time management services for the Department of Defense High Level Architecture (HLA) effort (IEEE Standard 1516), is an area editor for ACM Transactions on Modeling and Computer Simulation, and has also been chair of the steering committee for the Workshop on Principles of Advanced and Distributed Simulation since 1990. His email address is fujimoto@cc.gatech.edu.

MICHAEL HUNTER is an Associate Professor in the School of Civil and Environmental Department at the Georgia Institute of Technology. His primary areas of interest are traffic safety, operational performance and control, and simulation. He earned his Ph.D. at the University of Texas at Austin in August 2003. His e-mail address is michael.hunter@ce.gatech.edu.

WONHO SUH is a Ph.D. candidate at the Georgia Institute of Technology. His current research interest is in microscopic traffic simulation. He holds an M.S. in Civil Engineering from Seoul National University (2000). His email address is wonho.suh@gatech.edu.