International Journal of Electrical and Computer Engineering (IJECE)
Vol.3, No.1, February2013, pp. 64~72
ISSN: 2088-8708

AC-RDVT: Acyclic Resource Distance Vector Routing Tables for Dynamic Grid Resource Discovery

Nafiseh Bahrami, Ahmad Habibizad Navin, Mina Alavighi, Ali Asghar Pourhajikazem
Departement of Computer Engineering, Islamic Azad University of Tabriz, Iran

Article Info

Article history:

Received Nov 22, 2012
Revised Jan 10, 2013
Accepted Jan 22, 2013

Keyword:

Grid
Resource Discovery
RDV Routing Tables

ABSTRACT

Since the objective of grid is sharing the numerous and heterogeneous resources, resource discovery is a challenging issue. Recently appeared, Ontosum, is a resource discovery method based on semantically linked organizations and a routing algorithm Resource Distance Vector (RDV), has been presented to forward resource discovery queries into the clusters. Although this framework is efficient for large-scale grids and nodes are clustered automatically based on semantic attributes to constitute a semantically linked overlay network, but the dynamic behavior of grid isn't considered. In this method, deceptive information is stored in RDV tables (RDVT) which cause some problems in routing process. In this paper, a method is proposed to improve the dynamism of RDV routing algorithm, so the consistency with grid environments is increased. The developed algorithm is assessed by investigating the success probability, number of hops and routing time of resource discovery.

Corresponding Author:

NafisehBahrami,
Departement of Computer Engineering,
Islamic Azad University of Tabriz
Pasdaran highway, Tabriz, Iran.
Email: n_bahrami@gmail.com

1. INTRODUCTION

Grid is a specific type of parallel and distributed system that enables the selection, sharing, exchange, and aggregation of geographically distributed autonomous resources such as computers, software, catalogued data and databases, special devices/instruments (e.g. radio telescope), people/collaborators [1]. Since grid environment is dynamic and resources can be connected and disconnected to the system, efficient resource discovery method is required to enable users to access the grid resources. In this study a resource discovery method, Ontosum [2], is discussed to solve its problems. Nodes in Ontosum organize themselves automatically based on semantic attributes and constitute clusters. These clusters form an overlay network. A routing algorithm, RDV, is applied to forward resource discovery requests inside the clusters [3].Each node in the system maintains a routing table that includes the information of local resources and neighbors. One of the most important problems of routing tables in this method is the existence of some additional deceptive information that cause cycles in resource discovery mechanism.Deceptive information in RDVT may lead to some requests wrongly directed to offline resource which could increase false positive. Thus by requesting a special resource, the same resource will be reached several times through different neighbors.Therefore by going a resource to offline mode, information on the tables will guide the request to that offline resource again.

The main contribution of this paper is to eliminate the cycles by modifying RDVT. Elimination of cycles in the tables ensures that the information stored in the tables, will not guide the requests to the same offline resource. So by changing the mode of a resource to offline mode, another resource of the same type can be found instead of visiting that offline resource frequently.Simulation results show that the proposed method has better performance than previous one in comparison of the number of failed requests, routing time and number of hops.

The rest of this paper is organized as follow: section 2 reviews the previous works, proposed method appears in section 3.Simulation results are presented in section 4. Section 5 concludes the paper.

2. RELATED WORKS

According to the diversity and numerous numbers of resources in grid, finding suitable resources is important in these environments. An overview of some important resource discovery methods is given in this section. Resource discovery mechanisms may be centralized or decentralized. The most significant problem of centralized approaches is bottlenecks and also by increasing the number of nodes in system scalability becomes another problem [4-9]. To overcome to this difficulty peer-to-peer (P2P) techniques introduced decentralized methods [10, 11]. Rerouting table mechanism [12], is one of the decentralized methods in which, grid environment is comprised by routers and resources. Routing process in this approach uses routing tables in order to direct the requests for available resources in the system. A method has been proposed [13] to find suitable resource in grid which is based on reservation algorithm. This algorithm consists of forward path and backward path. In forward path, appropriate resources will be reserved and one of these resources will be chosen in backward path to reply the request. The method proposed in [14] uses tree architecture for grid resource discovery, in which query doesn't forward to all nodes unlike the flooding-based technology. Two passes are required in searching the tree. To find a match resource in a node or subtree of that node, first pass is starting from leaf node toward the root and next from this matched node toward leaf nodes (if the requested resource is in a subtree of a node).

To make the search easier recent years [15-18] they make different sets of nodes by using similar contents. Nodes with the same interests are clustered irrespective to the forming of the clusters [16]. The method proposed by Ng et al [18], Searching is based on periodic messages between nodes in which the messages impose a heavy overhead to the system. Semantic Overlay Network (SON) model [19] relies on central servers to cluster nodes which is considered bottleneck for the system. [20, 21, 22] add semantic shortcuts to the groups of nodes. Shortcut approaches are based on proximity of interests. A list of shortcuts is made by each peer for nodes that have responded to previous requests. To access the content, the peer will search the list of shortcuts for a request and if it doesn't find any response then the request will be broadcasted to every node.SLN (Semantic Link Network) [23, 24, 25-27] is provided by Zhuge as a semantic data model to organize the various web resources.

Semantic clustering is used to organize the network topology and reducing the search space in ontosum System [6] which is completely decentralized and automatically creates semantic groups. Therefore bottleneck is prevented and nodes are clustered based on certain interests and indexed in special formats which are used for routing in the network. Algorithms in previous works [2, 3] contain deceptive information in routing tables that may lead to cycles in resource discovery mechanism. Thus by requesting a special resource, the same resource will be reached several times through different nodes. So by changing the state of a resource to offline mode the request is forwarded to the same offline resource according to the information of tables. Eliminating the cycles in tables will resolve this problem which will be discussed in the next section.

3. PROPOSED METHOD

The proposed method aims to improve RDV tables in semantic routing architecture to overcome some of the problems. In semantic routing, nodes are clustered based on certain interests. Nodes in clusters form a tree and the root node keeps the information of whole cluster and form an overlay network for sharing their resources. Routing consists of two levels: Intra Cluster and Inter Cluster [3]. Proposed solution is performed in Inter Cluster level. The root of each cluster in an Inter Cluster routing is presented by the index of whole cluster named indicator of the cluster which form an overlay network. Routing requests among clusters is essential to share resources. Thus a routing algorithm called Resource Distance Vector, RDV, is used. By forwarding a request to a node, nearest node with desired resource is chosen through distance vector. Root nodes maintain a routing table in which all entries are set to infinity (not available) at first. Each node keeps the information of local resources and neighbors in the form of vectors of numbers that show nearest distance to each resource. When the information of one node is sent to another node, the vector is incremented. Local resources in RDVT are set to 0. All of vectors in each node is merged and sent to its neighbors and if there are multiple routes to a resource, nodes choose the shortest path for merging. By changing the information of each node, updated information must be sent to all neighbors. By forwardinga request to a node, first the local resources are checked and then the right neighbor is chosen to forward the request if no node is found. Figure1 shows this process with an example. See [2] for more information.

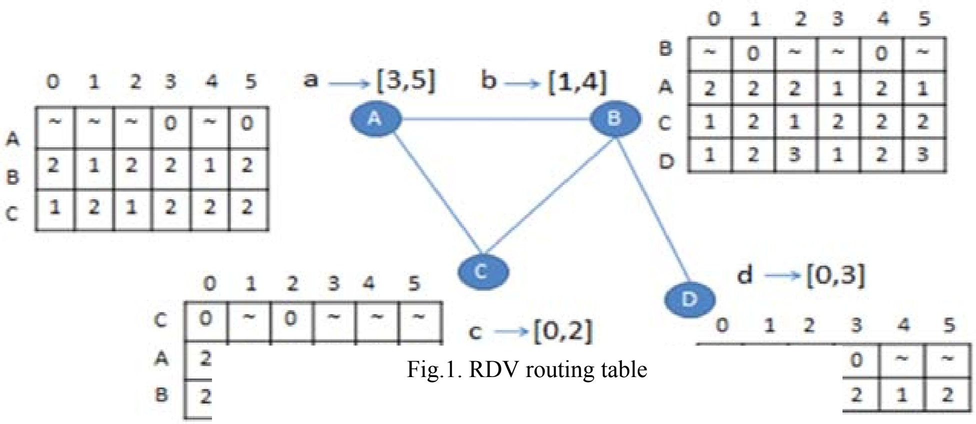

Fig.1. RDV routing table

3.1. AC-RDVT

Two problems exist in RDV tables: 1-Stored information in these tables directs the requests to the same resource through different paths. 2-There are cycles in these tables.
These problems will be overcome by eliminating extra and deceptive information in tables. Figure 2 shows the algorithm of the proposed method.

```
GN: Number of nodes in graph
NRES: Number of resources
rdv: RDV routing table (includes the information of local resources and neighbors of a node)
Neighbor: Number of neighbors (for node i)

for each node j in the graph do
\*Finding the minimum of each column in table
for   k=1    to    NRES
min ( k ) = big number
for   h=1    to   iNeighbor+1
if  (rdv ( h ,k, j ) < min ( k ) )
min ( k )= rdv( h, k, j )
min ( k ) = min ( k ) + 1

\* For each neighbor of node j, minimum of columns in RDV table is calculated in which the information of
node j and common nodes between j and m doesn't interfere in finding minimum.
for each node m that is neighbor with node j do
for  n=1    to   NRES
   min2 (n) = big number
for  h=1    to  iNeighbor+1
     If ( rdv ( h, n, m ) < min2 ( n ) )  and  ( h  ≠ j )   and  ( h ≠ common neighbor's number)
       min2 ( n ) = rdv ( h, n, m )
      min2 ( n ) = min2 ( n ) + 1

  \*  inserting the merged  information of each pair of neighbors in each other
for   p=1    to   NRES
rdv ( m, p, j ) = min2 ( p )
rdv ( j, p, m ) = min ( p )
```

Fig 2. Algorithm of AC-RDVT

The first problem arises by the common neighbors. Maintaining another table named **NON**, will solve this problem. This table includes the information of neighbor of neighbors. If there are any common neighbors between two nodes, the vector of that common node does not interfere in merging process. For example, in Figure 3, **C** is the common neighbor between **A** and **B**.

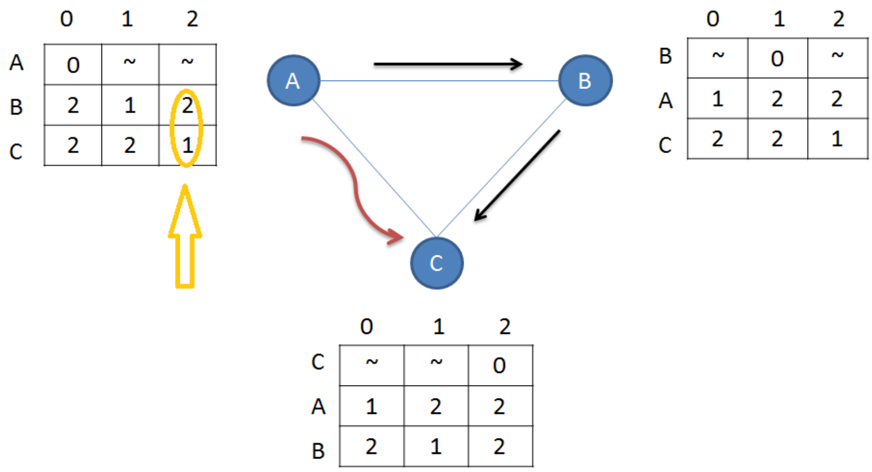

Fig 3. Problems of RDV method

According to the table, **A** accesses the resource 2 through two different paths: **AC** and **ABC**. In RDV method, by entering **C** to the system, it should send its information of its table to **B** and **B** must merge the new information by its vector and send it to **A**. node **B** doesn't know that **C** is a neighbor of **A** and thus the extra information of local resource of **C** that is accessible directly by **C**, is inserted to the table through **B** and **B** sends (2, 1, 2) to **A**. But if **B** was aware of this, it doesn't merge the vector of **C** for sending to **A** and thus we keep **NON** table to solve this problem. **NON** maintains neighbors of each node and their neighbors. For example,in Figure 4, in **A**'s **NON** table, the rows show the neighbor of **A** (**B**, **C**), and the columns show the neighbors of **B** and **C**.

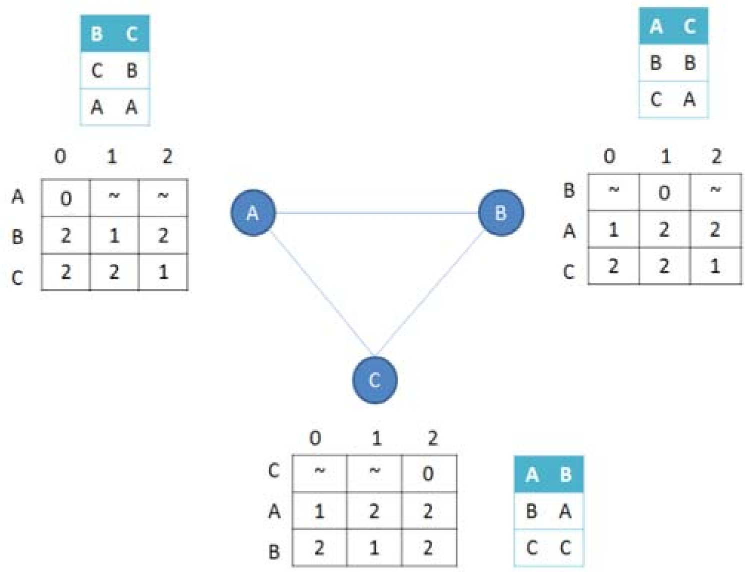

Fig 4. NON tables

When **B** wants to merge its information**,** first it checks its common neighbors between **A** and itself. If there is any common neighbor, their information doesn't merge. Since **C** is common neighbor of **A** and **B**, **B** doesn't merge vector **C** which is shown in Figure 5.

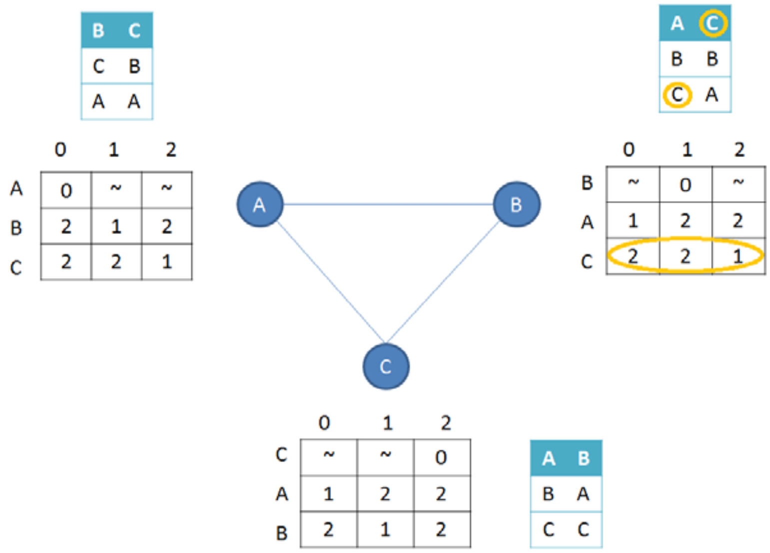

Fig 5. showing common neighbor between A and B by NON table

Current merging in RDVT causes some cycles in routing tables. When a node sends its information to its neighbors, the neighbors shouldn't return this information to it again. Otherwise, additional deceptive information in tables will be found that cause cycles. So by changing the state of a resource to offline mode, the algorithm may direct the request to that offline resource by information of tables. By eliminating these cycles in the tables, this problem will be solved. In the other words, in RDV method, additional information are merged and sent to the neighbors. Merging this information will create deceptive information in the tables which lead to cycles in routing mechanism. In the proposed method, this extra information doesn't merge therefore cycles are prevented. For example, in Figure 6, when **A** sends its information to **B**, merging by RDV method is as following:

Node **A** merges vectors **A** and **B**, lead into the vector A (~, 2, ~, 1, 2, 1). This vector is sent to node **B**. Table of node **B** shows that we can access the resource (1,4) by 0 hop and 2 hops through its local resources (0 hop) or trough its neighbor **A** (2 hops). According to Figure 6, node **B** returns to itself through node **A**. By the proposed method, when **A** wants to merge its vectors to send **B**, it shouldn't merge vector **B**. So the merged vector for node A is: A′ (~, ~, ~, 1, ~, 1).

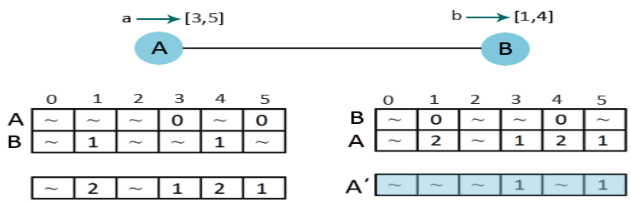

Fig 6. Solving the cycles in RDVT

4. EXPERIMENTAL RESULT

In this section experimental results are discussed to show the better performance of the proposed method. An overlay network is considered to be as a graph. Considering the difficulty of actual implementation, three experiments are simulated by a Pentium in C# environment. We setup experiments under following condition for both new method and pervious one. The average number of nodes in the system is considered to be 480 and the total number of requests is equal to 10000. The number of neighbors for each node is uniformly distributed in {1, 2, 3, 4, 5, 6}.Offline and online rates of resources follow Poisson distribution with the rate of 0.02. The experimental results show that the proposed method improves the efficiency of the system by considering three criteria: the number of failed requests, routing time and number of hops. Figure 7 shows the result of first experiment which compares the number of failed requests in both methods. The number of failures for each 1000 requests is calculated. Obtained results show that by increasing the number of requests, failure numbers is reduced in improved RDV. Proposed method improves the number of failed requests 16.87%. Figure 8 indicates the result of second experiment. The number of hops for each 1000 requests is calculated in both methods and the results are compared. This improvement is 16.98%. Figure 9 shows routing time in RDV and Improved RDV for each 1000 requests. Proposed method improves routing time 36%. Experimental results show the efficiency of proposed method in comparison with RDV method that is the result of eliminating extra information in merging stage and preventing to produce the deceptive information.

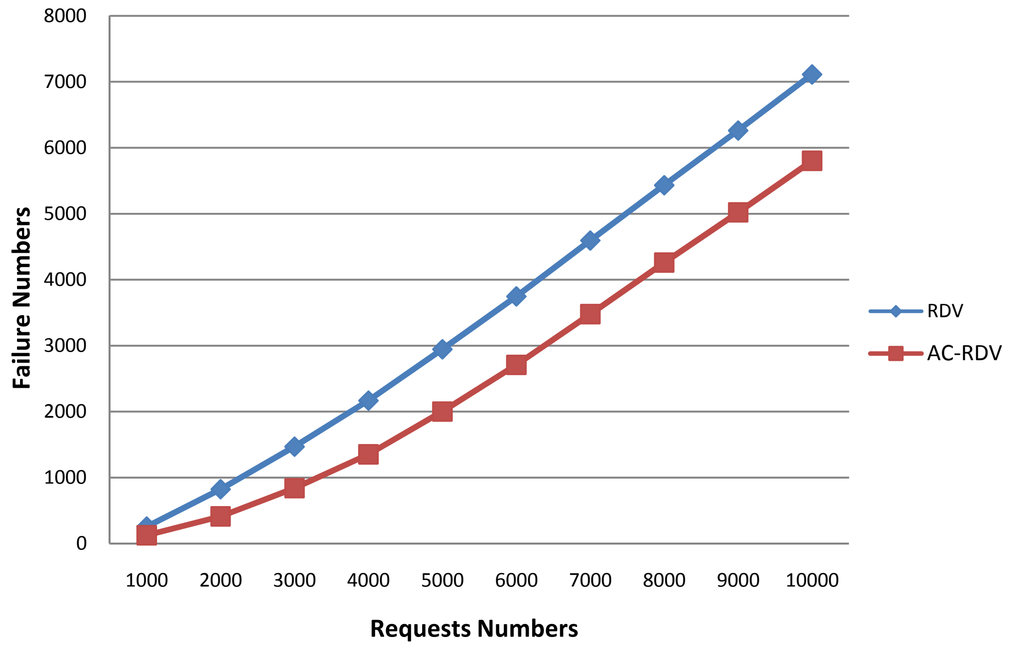

Fig 7**.** Average number of failed requests per each 1000 requests

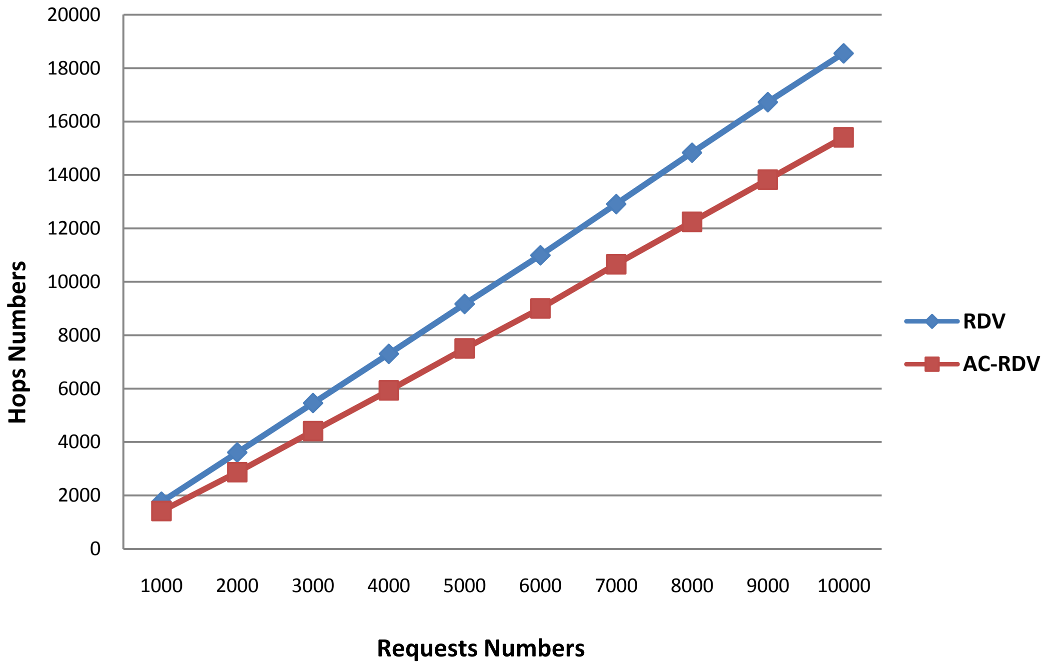

Fig.8. Average number of hops per each 1000 requests

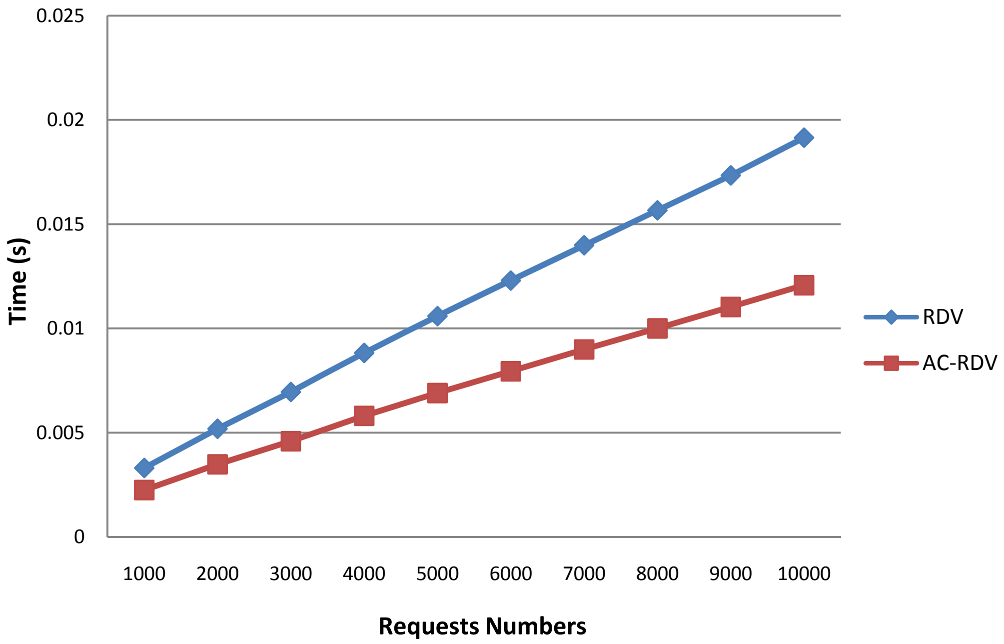

Fig 9. Routing time per each 1000 requests

5. CONCUSION

In this paper, an algorithm is proposed to solve two problems of RDV routing tables in Ontosum method, which prevents directing requests to offline resources frequently. The idea is based on the omission of merging the unnecessary information of neighbors in filling process of routing tables and therefore preventing the creation of deceptive information in tables and so preventing cycles in routing mechanism.The performance of proposed method is evaluated with simulation experiments which confirm the efficiency of algorithm in three aspects: the number of failed requests, routing time and number of hops. Experimental results show that proposed method improves the number of failed requests 16.87%, routing time 36% and number of hopes 16.98% in comparison with RDV method.

REFERENCES

[1] R. Buyya, S. Venugopal, A. Gentle Introduction to Grid Computing and Technologies, in: ComputerSociety of India (CSI), vol. 29, no. 1, pp. 9-19, 2005.

[2] Juan Li, Grid resource discovery based on semantically linked virtual organizations , Future Generation Computer Systems, pp. 361_373, 2010.

[3] Juan Li, Son Vuong, A Scalable Semantic Routing Architecture for Grid Resource Discovery , Conference on Parallel and Distributed Systems (ICPADS'05) , 2005.

[4] I. Foster, C. Kesselman, Globus: A metacomputing infrastructure toolkit, Int. J. High Perform. Comput.Appl. 2, pp. 115–128, 1997.

[5] M. Mutka, M. Livny, Scheduling remote processing capacity in a workstation processing bank computing system, in: Proc. of ICDCS, September 1987.

[6] C. Germain, V. Neri, G. Fedak, F. Cappello, XtremWeb: Building an experimental platform for global computing, in: Proc. of IEEE/ACM Grid, December 2000.

[7] A. Chien, B. Calder, S. Elbert, K. Bhatia, Entropia: Architecture and performance of an enterprise desktop grid system, J. Parallel Distrib. Comput.,2003.

[8] F. Berman, et al., Adaptive computing on the grid using AppLeS, TPDS, 2003.

[9] M.O. Neary, S.P. Brydon, P. Kmiec, S. Rollins, P. Capello, JavelinCC: Scalability issues in global computing, Future Gener. Comput.Syst. J. pp. 659-674, 1999.

[10] M. Cai, M. Frank, J. Chen, P. Szekely, MAAN: A multi-attribute addressable network for grid information services, in: The 4th International Workshop on Grid Computing, pp. 184-191, 2003.

[11] A. Iamnitchi, I. Foster, On fully decentralized resource discovery in grid environments, in: Proceeding of the 2nd IEEE/ACM International Workshop on Grid Computing, Denver, pp. 51-62, 2001.
[12] K. Karaoglanoglou, H. Karatza “Resource discovery in a dynamical Grid system based on re-routing tables”, Journal of Simulation Modelling Practice and Theory, vol. 16, pp. 704-720, 2008.
[13] S. Tangpongprasit, T. Katagiri, H. Honda, T. Yuba, A time-to-live based reservation algorithm on fully decentralized resource discovery in grid computing, Parallel Computing, pp. 529-543, 2005.
[14] R.-S. Chang, M.-S.Hu, A resource discovery tree using bitmap for grids, Future Gener.Comput.Syst., pp. 29–37, 2010.
[15] M. Bawa, G.S. Manku, P. Raghavan, SETS: Search enhanced by topic segmentation, in: Proceedings of ACM SIGIR, pp. 306-313, 2003.
[16] A. Iamnitchi, M. Ripeanu, I.T. Foster, Locating data in peer-to-peer scientific collaborations, in: Proceedings of International Workshop on Peer-to-Peer Systems, IPTPS, pp. 232-241, 2002.
[17] W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M.T. Schlosser, I. Brunkhorst, A. Lser, Super-peer-based routing and clustering strategies for RDF-based peer-to-peer networks, in: Proceedings of International World Wide Web Conference, WWW, pp. 536-543, 2003.
[18] C.H. Ng, K.C. Sia, C.H. Chang, Advanced peer clustering and firework query model in the peer-to-peer network, in: Proceedings of International World Wide Web Conference, WWW, Poster ID S130.
[19] A. Crespo, H. Garcia-Molina, Semantic overlay networks for p2p systems, Technical report, Stanford University, 2002.
[20] X. Tempich, S. Staab, A. Wranik, REMINDIN': semantic query routing in peerto- peer networks based on social metaphors International World Wide Web Conference (WWW), New York, USA, pp. 640-649, 2004.
[21] S. Castano, A. Ferrara, Montanelli, D. Zucchelli, Helios: A general framework for ontology-based knowledge sharing and evolution in P2P systems, in: IEEE Proc. of DEXA WEBS 2003 Workshop, Prague, Czech Republic, 1(5), pp. 597-603, 2003.
[22] S. Castano, A. Ferrara, S. Montanelli, E. Pagani, G. Rossi, Ontology addressable contents in P2P networks, in: Proceedings of the WWW'03 Workshop on Semantics in Peer-to-Peer and Grid Computing, pp. 55-68, 2003.
[23] [23] H. Zhuge, RuixiangJia, Jie Liu, Semantic link network builder and intelligent semantic browser, Concurrency - Practice and Experience, pp. 1453_1476, 2004.
[24] H. Zhuge, Yunchuan Sun, RuixiangJia, Jie Liu, Algebra model and experiment for semantic link network, IJHPCN, pp. 227_238, 2005.
[25] H. Zhuge, Communities and emerging semantics in semantic link network: Discovery and learning, IEEE Transactions on Knowledge and Data Engineering, pp. 785-799, 2009.
[26] H. Zhuge, X. Li, Peer-to-peer in metric space and semantic space, IEEE Transactions on Knowledge and Data Engineering, pp.759-771, 2007.
[27] H. Zhuge, Semantics, resource and grid, Future Generation Computer Systems, pp. 1-5, 2004.

BIOGRAPHIES OF AUTHORS

NafisehBahramireceived her B.S. (2008) from Azad university of Tabriz, Iran. She is currently M.S student in Azad university of Tabriz, Iran. Her research interests include grid computingand wireless sensor network.

Ahmad Habibizad Navin He received the B.S. degree in applied mathematics from Tabriz University, Tabriz, Iran, in 1999. He received the M.S. degree in computer architecture from University of Science and Research Branch, Islamic Azad University, Tehran, Iran, in 2003 and the Ph.D. in computer architecture from University of Science and Research Branch, Islamic Azad University, Tehran, Iran, in 2007.His research interest includes data-oriented approach, robotic, soft computing and probability and statistic.

Mina alavighireceived her B.S. (2007) from Azad university of Shabestar, Iran and her M.S (2011) from Azad university of Tabriz, Iran. Her research interests include grid computing and wireless sensor network.

Ali AsgharPourhajiKazemreceived a B.Sc. degree in computer engineeringfrom University of Isfahn and also a M.S. degree in computer engineering fromShahidBeheshti University in Tehran. He is currently a Ph.D. student of computerengineering in Science and Research branch of Islamic Azad University in Tehran.His current research interests include distributed systems, Grid computing and Cloud computing.