Implementation of Fault Tolerance Algorithm to Restore Affected Nodes in Scheduling Clusters

Fadel Hussen[1], Khaled Elleithy[2], and Abdul Razaque[3]

Department of Computer Science & Engineering
University of Bridgeport, Bridgeport, CT- 06604, USA
{fhussen, elleithy, arazaque}@bridgeport.edu

ABSTRACT

Due to the convergence of the networks, the top priority objective of researchers is to get the network fully connected. Several types of networks have been introduced and proposed to improve performance. Cluster environment provides full support for various applications. Scheduling is one of the most important research-focusing areas, where different supporting algorithms are implemented. However, there is still a gap in scheduling to provide best network connectivity to all nodes. This paper targets nodes affected issue that occurs due to scalability, data sharing, while leaving and joining the nodes. To control and retain an affected node in the clustering scheduling, fault tolerance techniques are applied. The base of this technique is Node Recovery Algorithm (NRA). This algorithm supports disconnected nodes and restores them to join the scheduling. Furthermore, this algorithm maximizes the efficiency of the cluster and improves the performance.

KEYWORDS

Fault tolerance, Scheduling, cluster, algorithm.

1. INTRODUCTION

There are lots of challenging issues that affect the routing protocols in WSNs. The following are some of the routing challenges design factors when we design the routing protocols: node deployment, energy consumption, data reporting model, scalability, connectivity, data aggregation, quality of service, transmission media and fault tolerance. We shed the light on fault tolerance technique to provide best network connectivity.

Cluster environments consist of an array of various computers connected by high speed networks to achieve powerful platforms. Cluster computing systems are extensively deployed for executing computationally intensive parallel applications with various computing requirements. For many years, programming a parallel system to implement a single application has been a challenging problem. It is really more challenging than programming a single processor or a sequential system [1].

One of the most important steps in parallel programming is allocating the tasks to the processors and specifying the order of the execution [2]. There are many types of scheduling algorithms which are very well known; some of them are dynamic and static scheduling. In dynamic scheduling, the decision as to which processor executes a task and when is controlled by the runtime system [3]-[5]. There are many advantages of static scheduling which include the

DOI : 10.5121/ijcnc.2012.4101

dependences and communications among the tasks in its scheduling decisions. There are some attempts which aim to develop new techniques to achieve fault tolerance.

Fault tolerance is an important property in distributed computing as the dependability of individual resources may not be guaranteed. A fault tolerant approach may therefore be useful in order to potentially prevent a malicious node from affecting the overall performance of the application [4]-[8]. The main goal of fault tolerant task scheduling algorithms is to find a static schedule of application tasks on the processing elements of a cluster computing system and tolerate a given number of processor failures [10]. The input of the fault tolerant scheduling algorithm is a specification of the application tasks, the computation power of the processing elements, and some information about the execution times of the tasks on the system processors and the communication times between the tasks [11]-[14].

Fault tolerance is the capability of a system to respond to a highly unexpected hardware or software failure [15]. The fault tolerance consists of many levels, the lowest of which has the ability to continue the operation when the power failure. There are a lot of fault-tolerant computer systems where every operation can be performed on two or more duplicate systems. In case one fails the other can take over. In addition, fault tolerance needs several form of redundancy in time, space, or information. For example, when an error occurs, it either needs to be corrected/masked or the operation needs to be retried if it is a temporary fault [8].

Retrying an operation will not solve the problem, if there is a permanent fault. In that case, sufficient redundancy or spare units are required to continue error-free operation, or the part needs to be repaired or replaced. The vast majority of fault-tolerant computer systems are designed to be capable to handle some possible failures, for instance, input or output device failures or hard disk failures or other temporary or permanent failures [13], [15].

The rest of this paper is organized as follows. Section 2 reported related work of fault tolerance.. In section 3, we described proposed fault tolerance algorithm for the connectivity of nodes in cluster. Section 4 presents the simulation setup in this paper. In section 5, we provide the simulation analysis and discussion, followed by conclusion in section 6 and references.

2. RELATED WORK

The paper discusses some salient features of related techniques used in scheduling for recovery of nodes. Previous studies highlight the importance of fault-tolerance in relation to map-reduce discussed in [16]. The authors state that check pointing and replication are frequent moves toward supporting fault-tolerance in clustering. The replication has been a key part in map-reduction implementations, given its usefulness when the file systems replicate the datasets. The processing configuration in map-reduce is of great importance in keeping the reliability and correctness from failures.

The Data Acquisition and Generation (DAG) algorithm is discussed in [17]. It represents an application supported with information regarding the tasks. The information comprises execution time for tasks covering the communication times and target system processors.

The algorithm targets an active replication structure and its correspondence schedules $\varepsilon+1$ that replicas to each task. It supports getting the given fault tolerance. Simulation results highlight an efficiency of the proposed algorithm despite having lower complexity. The authors have described numerous new clustering algorithms for a mobile ad hoc network in [18]. The authors have focused on simplifying the definition of cluster including the formation algorithm. The

proposed framework is based on random unit graphs to support the obstacles. The authors have proposed the distributed network clustering protocol (SDC) in [19], [20]. The authors discuss the clusters which are dynamically made and attuned based on practical clustering accuracy and standard cluster munitions (SCM). The authors claim that accuracy of the network is improved even if node joins or leaves the network but it takes long time for this process as shown in Figure 1 and Figure 2.

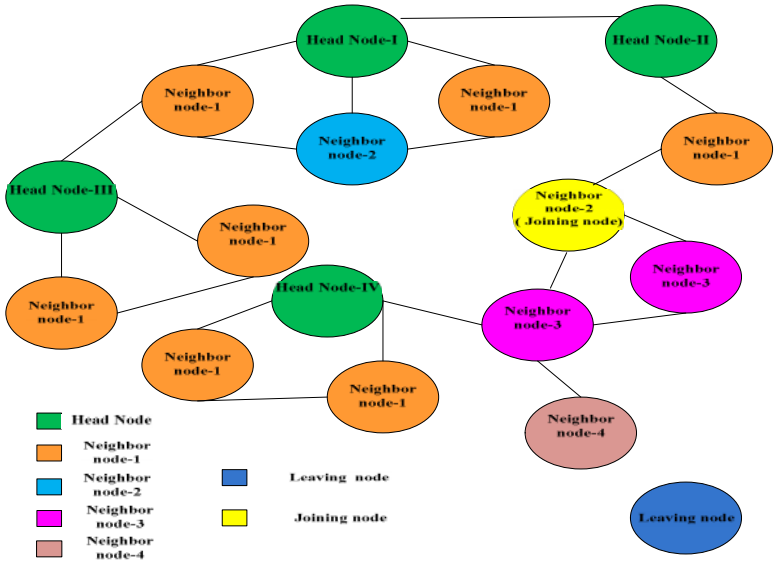


Figure 1. Node leaving process

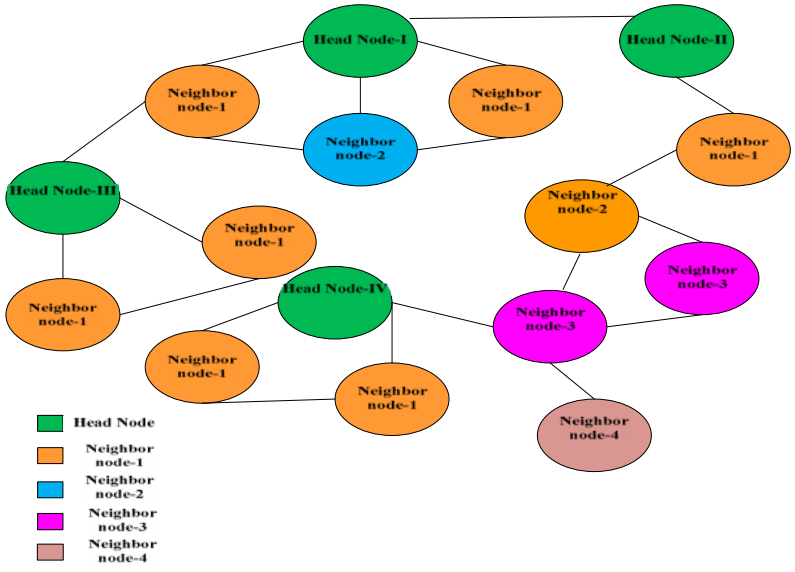


Figure 2. Node joining process

3. PROPOSED FAULT TOLERANCE ALGORITHM FOR THE CONNECTIVITY OF NODES IN CLUSTER

To restore the connectivity of the network because of joining and leaving the nodes, (NDRA) Network Recovery Algorithm is proposed to retain the effective node and maintain the network connectivity. (NRA) maximizes the performance of nods and increases the efficiency of scheduling clusters.

Algorithm 1:

1- Set basic blocks size of node 6 log2 n.
2- Set number of nodes 3t+1 that is upper bounded.
3- Set broadcasting offline & used when node joins & leaves the cluster.
4- Set single message from each source.
5- Set number of sources t + r.
6- Computational complexity O(t2).
7- Joining nodes are Na, U, V for this purpose.
8- Set number of next cluster Na(i); here i= destination of cluster and distance of link from source to destination is set as [a](U(i))=0 .
9- If node Na, U, V ∈ [i], V[i].
10- Set closest to source node [i].
11- Set the lists mode open and close.
12- Set closest that must be considered as consideration & empty.
13- Select cluster from open that possesses maximum distance from source node U(i).
14- Move the selected cluster to close.
15- Add next nodes to clusters.
16- If cluster [x] is moved then cluster [y] is added.
17- Then, set Na(y) as Na(x).
18- If Na(x) ≠ 0.
else
19- Set Na(y) = x.
20- If cluster [y] is adjacent to [x] & ∈ open.
21- Check the next distance in cluster [x] from [y].
22- Modify the value of V(y) & U(y).
23- Set new value U(y).
24- Calculate new value U(x)+R here; R=distance from V(x) and V(y), V(y) is the nearest path to V(x) node in cluster [y].
25- Join the node U(y) & V(x) on selected link for connectivity of networks.
26- end if
27- end if
28- end else
29- end if
30- end if

In this section, we describe the proposed algorithm for joining and leaving the cluster structure in the presence of topological changes. There are four conditions to be considered: node joins or switches off network, a node leaves and switches off network, a link is broken, and a link is made between two existing nodes after they come closer to each other. Node checks whether it is at distance at k hops, when it switches on from any of the cluster heads, and, if so, joins these

clusters. Otherwise, the node makes a new cluster with itself as cluster head, and invites its k-hop neighbors to join the cluster. This method does not diverge from the one when a node shifts off; no change is made if the node is not a CH. In case a CH fails, the nodes in the cluster selects a new CH by using number of k-hop neighbors within the cluster as the key standard (total number of k-hop neighbors as secondary, and ID as ternary criterion).

A node which is not added in the new cluster recurrences this method until all of them are added in a cluster. This method may outcome if a current link is detached; no modification is made if the two nodes relate to different clusters. Else, all nodes in their cluster are notified, and their CH confirms whether all nodes are still on k-hop neighbours in clusters. If so, no modification is made. Otherwise, nodes with hop count from CH bigger than k build a new cluster. Finally, if a new link is made between two nodes A and B, there are many cases to be considered. If there is no cluster head means neither A nor B, so no change in the mechanism is made. However, this is a accurate process only for k = 1. For k > 1, two cluster head may decrease their shared distance to at most k-hops, which disrupts the explanation of k-clusters. The description of the update method in this case is not unimportant, and is misplaced. In practice, it is not projected to apply values of k that are larger than 2, and details for k = 2 may be described in a direct method. If both A and B are cluster heads, first choose the one that conserves the role, using the identical standard outlined above. Nodes from the other cluster join the captivating CH if they are its new k-hop neighbours (note that for k = 1 no such node occurs). Otherwise, they make a new cluster(s). Finally, if a new link is made between two nodes A and B, there are many cases to be considered. If there is no cluster head means neither A nor B, no change in the mechanism is made. However, this is a accurate process only for k = 1. For k > 1, two cluster head may decrease their shared distance to at most k-hops, which disrupts the explanation of k-clusters. The description of the update method in this case is not unimportant, and is misplaced. In practice, it is not projected to apply values of k that are larger than 2, and details for k = 2 may be described in a direct method. If both A and B are cluster heads, first choose which of them conserves the role, using the identical standard outlined above. Nodes from the other cluster join the captivating CH if they are its new k-hop neighbours (note that for k = 1 no such node occurs). Otherwise, they make a new cluster(s).

4. SIMULATION SETUP

Table: 1. shows summarized Simulation parameters

Parameters	Specifications
Ns2	2.28 on Red Hat 8
transmission range	250 meter
interference range	550 meter
Protocols for transport	TCP
The length of packet	740 bytes
mobile nodes	14
rectangular field	1000 *1000 meters
Speed	0 to 20 m/sec
Simulation time	60 seconds
Pause time/ warm up time	2 seconds
Mobility model	Random way point mobility model

Block size	6 log2 n
Status of broadcasting	Offline
Upper bounded traffic	T
Lower bounded traffic	R
Computational complexity	O(t2)
Na	Name of node
Name of clusters	Na(i); Na(x)
Initial Energy	0.5 Joule

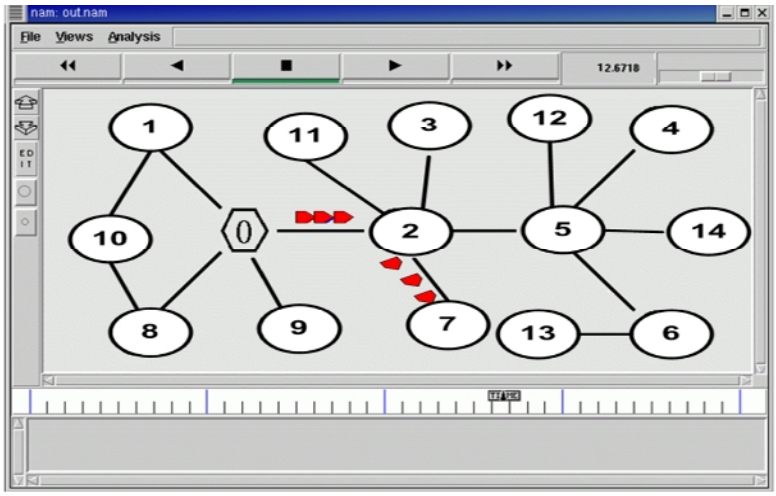


Figure 3. NAM for network simulation

- ***REAL SCENARIO***

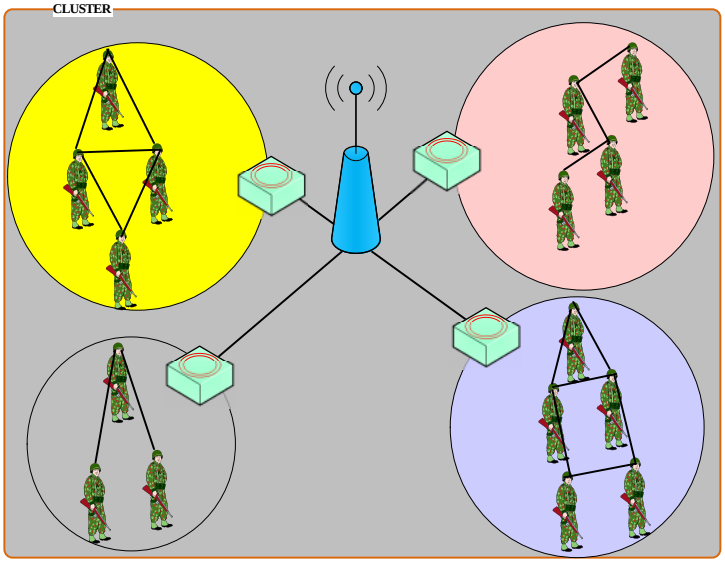


Figure 4. Battle field scenario

5. SIMULATION ANALYSIS AND DISCUSSION

In this part we present simulation results in order to show the actual performance of Cluster. The results are depicted in two parts. First, we focus on the performance of the algorithm used for joining the nodes in the cluster. The second part is about the throughput comparison of existing algorithms with respect to our proposed NRA algorithm. Several performance metrics related to algorithm performance are studied in this simulation method. Ns2 is utilized to build the network simulation model, with results of 14 runs.

- ***Clustering Results***

In order to determine connectivity of nodes in clusters, several connections are broken to identify the real complexity of the algorithm. We used random way point mobility model. In addition, a set of new cluster heads is elected for each round leaving remaining nodes to be cluster members.

In the data period, the cluster head receives 540 bytes data packets in every turn from each node. We measure the performance of network connectivity through commonly used methods of active nodes percentage, where the number of rounds before individual sensor node energy is exhausted. There are two ways to define connectivity and life time of a network; either by a known percentage of disconnected nodes or by the number of rounds until the first node leaves. We question use of this definition for large scale networks. Also for the reason that some sensor nodes continue to be operational, maintaining a certain degree of connectivity is required for data gathering.

In addition to the discontinuation of first node, the network life is measured when 10% and 20% of the nodes are disconnected. For the ns2 clustering simulations, three parameters were considered important to cluster formation, namely; node to cluster head distance, sink to cluster head distance, and the cluster head residual energy.

In order to calculate performance of the whole network, we first use two parameters, while the cluster head is selected based on higher residual energy. Figure 5 shows the whole performance of network when nodes either join or leave.

Network lifetime is a highly significant aspect for connectivity of nodes. Connectivity of clusters is evaluated under different topological configurations and network sizes. The other important factor of this contribution is to find out the time complexity of the network. It helps to determine the mean time for each node to join and leave the cluster shown in Figure 6.

On basis of above mentioned concepts and parameters of our study, we conclude that network connectivity is significantly improved by NRA comprised to other algorithms, as NRA algorithm shows that joining and leaving capability of nodes do not affect the performance of the whole network. In addition, the network receives maximum data packets as compared with other approaches.

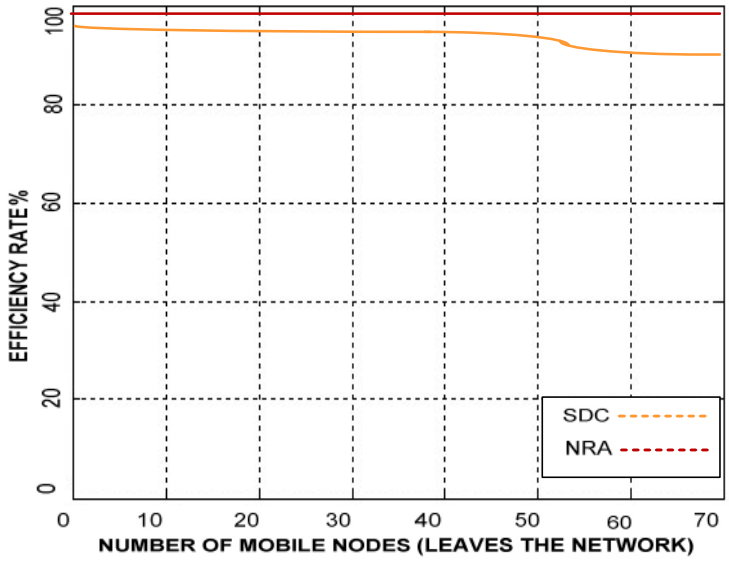


Figure 5. Showing the whole performance of network when nodes either join or leave.

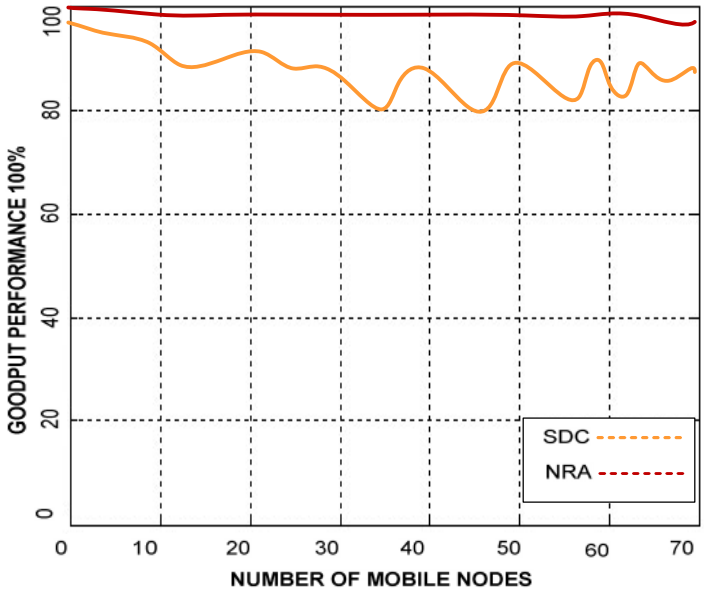


Figure 6. Showing mean time for each node to join and leave the cluster.

6. CONCLUSION

In this paper, Node Recovery Algorithm (NRA) is presented. NRA controls the connectivity of a network. The disconnected network disturbs the communication; resulting nodes cannot send and receive data. Furthermore, a very challenging issue is how to maintain the process of leaving and joining the nodes. The proposed NRA algorithm not only controls the network but also handles the routing issue. To validate the performance of NRA, the realistic scenario of a battlefield is

chosen. The scenario consists of a mixed environment that covers mobility and static state of nodes. NRA algorithm is implemented in ns2 and mapped on the scenario. On the basis of simulation, we obtain very motivating findings. We also compare our algorithm with an already existing popular SDC algorithm. Our algorithm provides a better efficiency rate as compared with SDC. The accuracy of our algorithm is almost 99. 5%.

NRA increases the quality of service of nodes for providing maximum of amount of data with negligible loss. In the future, NRA will be enhanced to support various routing protocols. Furthermore, the broader impact of this research is to support the highly congested network because nodes frequently keep on changing their positions. We believe that NRA will perform better and be employed for various types of mobility models.

REFERENCES

[1] Buyya, R.: High Performance Cluster Computing: Architectures and Systems, 1st edn. Prentice Hall PTR, pper Saddle River. 1999.

[2] Sinnen, O.: Task Scheduling for Parallel Systems, 1st edn. John Wiley and Sons Inc, NewJersey. 2007.

[3] Zheng, Q., Veeravalli, B., Tham, C.K.: On the design of fault-tolerant schedulingstrategies using primary-backup approach for computational grids with low replicationcosts. IEEE Transactions on Computers 58(3), 380–393. 2009.

[4] Sathya, S.S., Babu, K.S.: Survey of fault tolerant techniques for grid. Computer ScienceReview 4(2), 101–120. 2010

[5] Guowei Wu , Chi Lin , Feng Xia , Lin Yao , He Zhang, and Bing Liu "Dynamical Jumping Real-Time Fault-Tolerant Routing Protocol for Wireless Sensor Networks", Sensor 2010.

[6] M.Zahid Khan, M. Merabti, B. Askwith, F. Bouhafs, "A Fault-Tolerant Network Management Architecture for Wireless Sensor Networks", PGNet, 2010.

[7] M. Zaharia, D. Borthankur, J. Sarma, K. Elmellegy, S. Shenker, and I. Stoica, Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in cluster Scheduling. In EuroSys 2010, pp. 265-278. ACM, New York (2010).

[8] Oh, Y., Son, S.H.: Scheduling real-time tasks for dependability. Journal of OperationalResearch Society 48(6), 629–639. 1997.

[9] Parsa, S., Entezari-Maleki, R.: RASA: A new grid task scheduling algorithm. InternationalJournal of Digital Content Technology and its Applications 3(4), 91–99. 2009.

[10] K. Arisha, M. Youssef, M. Younis, "Energy-Aware TDMABased MAC for Sensor Networks," IEEE Workshop on Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT 2002), May 2002.

[11] Ghosh, S., Melhem, R., Mosse, D.: Fault-tolerance through scheduling of aperiodic tasksin hard real-time multiprocessor systems. IEEE Transactions on Parallel and Distributed Systems 8(3), 272–284, 1997.

[12] Manimaran, G., Murthy, C.S.R.: A fault-tolerant dynamic scheduling algorithm formultiprocessor real-time systems and its analysis. IEEE Transactions on Parallel andDistributed Systems 9(11), 1137–1152, 1998.

[13] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Scalable coordination in sensor networks. Proc. of ACM/IEEE MobiCom 1999, Seattle, Washington, August, 1999.

[14] G. Gupta, M. Younis, "Load-Balanced Clustering in Wireless Sensor Networks", Submitted to the IEEE International conference on communications (ICC 2003), Anchorage, Alaska, May 2003.

[15] Al-Omari, R., Somani, A., Manimaran, G.: A new fault-tolerant technique for improvingschedulability in multiprocessor real-time systems. In: The 15th International Parallel andDistributed Processing Symposium, pp. 32–39, 2001.

[16] Zheng, Q., Veeravalli, B., Tham, C.K.: Fault-tolerant scheduling of independent tasks incomputational

grid. In: The 10th IEEE International Conference on Communications Systems, pp. 1–5, 2006.

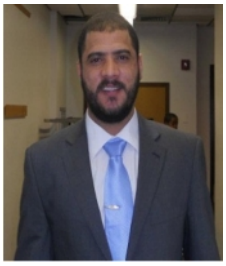
[17] Nabil Tabbaa, Reza Entezari-Maleki, and Ali Movaghar, "A Fault Tolerant Scheduling Algorithm for DAG Applications in Cluster Environments", ICDIPC 2011, Part I, CCIS 188, pp. 189–199, Springer-Verlag Berlin Heidelberg 2011.

[18] FABIAN GARCIA NOCETTI and JULIO SOLANO GONZALEZ, "Connectivity Based k-Hop Clustering in Wireless Networks", Telecommunication Systems 22:1–4, 205–220, Kluwer Academic Publishers. Manufactured in The Netherlands. 2003.

[19] Yan Li, Li Lao and Jun-Hong Cui", SDC: A Distributed Clustering Protocol", International Journal of Computer Networks, (IJCN), Volume (2): Issue (6), 2005.

[20] R. Pandi Selvam and V.Palanisamy, "An Efficient Cluster Based Approach for Multi-Source Multicast Routing Protocol in Mobile Ad Hoc Networks",International Journal of Computer Networks & Communications (IJCNC) Vol.3, No.1, January 2011, pp 154-166.

Authors



Fadel Hussen is M.S. student in Computer Science & Engineering Department at The University of Bridgeport. He received his High Diploma in Computer Science from The Higher Professional Center for Comprehensive Professions (HPCCP), Ghadamis City, Libya in May 2002. He worked as a teacher in high school for more than five years from 2002 to 2008. Fadel Hussen worked as a teacher assistant at (HPCCP) one of the best high institutes in Libya. Fadel Hussen is currently doing his Master Degree in Computer Science at the University of Bridgeport, and he is expecting to graduate in the Fall semester 2011. Also he is planning to pursue his PhD in Computer Science at the same university UB. He has published one research paper in very reputed conference ASEE.

Dr. Elleithy is the Associate Dean for Graduate Studies in the School of Engineering at the University of Bridgeport. He has research interests are in the areas of network security, mobile communications, and formal approaches for design and verification. He has published more than one hundred fifty research papers in international journals and conferences in his areas of expertise. Dr. Elleithy is the co-chair of the International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE). CISSE is the first Engineering/Computing and Systems Research E-Conference in the world to be completely conducted online in real-time via the internet and was successfully running for four years.Dr. Elleithy is the editor or co-editor of 10 books published by Springer for advances on Innovations and Advanced Techniques in Systems, Computing Sciences and Software. Dr. Elleithy received the B.Sc. degree in computer science and automatic control from Alexandria University in 1983, the MS Degree in computer networks from the same university in 1986, and the MS and Ph.D. degrees in computer science from The Center for Advanced Computer Studies at the University of Louisiana at Lafayette in 1988 and 1990.

Abdul Razaque is Ph.D. student of computer science and Engineering department in University of Bridgeport. His current research interests include the design and development of learning environment to support the learning about heterogamous domain, collaborative discovery learning and the development of mobile applications to support mobile collaborative learning (MCL), the congestion mechanism of transmission of control protocol including various existing variants, delivery of multimedia applications. He has published over 30 research contributions in refereed conferences, international journals and books. He has also presented his work more than 10 countries. He completed his Bachelor and Master degree in computer science from university of Sindh in 2002. He obtained another Master degree with specialization of multimedia and communication (MC) from Mohammed Ali Jinnah University, Pakistan in 2008. Abdul Razaque has been directly involved in design and development of mobile applications to support learning environments to meet pedagogical needs of schools, colleges, universities and various organizations