International Journal of Pure and Applied Mathematics
Volume 117 No. 20 2017, 971-983
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue
ijpam.eu

Survey of Mutual Exclusion and its variants in Distributed Systems and MANET's.

Ashish Khanna

Assistant Professor, Department of Computer Science,
MAIT, GGSIPU, Delhi
Ashishk746@yahoo.com

Abstract:

In present world most of the applications developed are of distributed nature. These applications are implemented in distributed environment or systems. Whereas, distributed system can be considered as a system where multiple entities are working in parallel. In distributed system resource allocation is considered as one of the most challenging problem. The resourcesharing is prime motivation behind the evolution of distributed computing systems. Hence, a number of processors may compete to have mutually exclusive access to shared resources. The above discussed problem is considered as mutual exclusion. In the presented paper a survey of Mutual exclusion and its variants detailed discussion is presented.

1. Introduction

A message passing distributed system is a collection of autonomous computers that are geographically dispersed, connected via a network and appears to its users as a single coherent system [1]. The nodes in a message passing system do not share memory or clock; therefore, handling message-passing systems is more complex than shared memory systems [2]. Some typical examples of distributed systems include World Wide Web (www) [3], peer to peer networks [4, 5], cloud computing systems [6, 7], sensor networks [8], cellular networks [9], grid computing systems [10], and mobile ad hoc networks (MANETs) [11]. The major research challenges in distributed systems are consensus [4], mutual exclusion [12], leader election [13], snapshot [14], clock synchronization [15], and termination detection [16].

Due to widespread growth in wireless communication and hardware devices, multiple modern applications using mobile devices have evolved which are distributed in nature

[17-20]. Now a day's, multifunctional powerful wireless devices (laptops, *PDAs*, and Smartphones.) are easily available which can be used for various types of applications. A wireless device may act as a wireless node. Such nodes may form a wireless network as and when required by the application. These networks are created on demand; hence, we call them ad hoc networks. Ad hoc networks have become popular among masses and researchers in recent years.

2. Classification of Mobile Computing Systems

Mobile computing systems can be classified into two variants, namely cellular network [21] and mobile ad hoc networks (MANET) [22, 23]. The architecture of these variants is shown in figure 1.1 and figure 1.2.

2.1**Cellular networks**: A cellular network is an infrastructure-based mobile network.The deployment area is distributed into cells; one fixed host called base station (*BS*) serves each cell.Also, a cell contains some mobile hosts (*MHs*). The base station acts as coordinator for the *MHs* belonging to a particular cell. A mobile host may move from one cell to another. The communication between*MHs* is via a *BS*. To avoid inter-cell interference; the neighboring cells usea different set of frequencies.

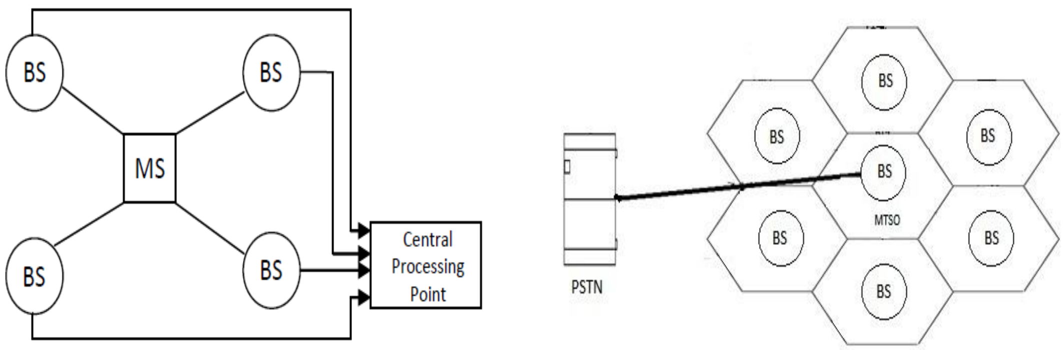

Fig. 1.1: Cellular Network

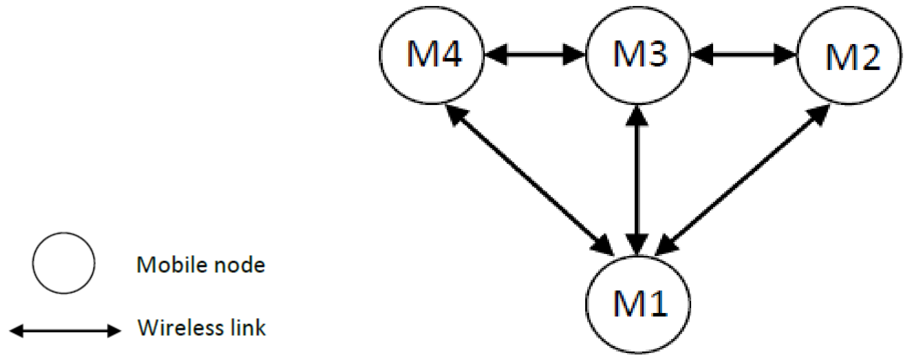

Fig. 1.2: Ad hoc Network (MANETs)

2.2.Mobile adhoc networks (MANETs):

A mobile ad hoc network (MANET) is an infrastructure-less, collection of mobile hosts. The *MHs*communicate with each other overwireless links and cooperate in a distributed mannerto provide the necessary network functionality. MANETis generally deployed where deployment of fixed network is either infeasible or cost ineffective. All the mobile nodes also act as a router for other nodes. MANETacts as a stand-alone network that can be connected to a cellular network or Internet at single or several points. Furthermore, it paves the way for numerous applicationssuch as emergency and rescue operations, conference or campus settings, car networks, and personal networking.

The integration of MANETs with other networks allows us to use them in several day to day applications. Hence, the popularity of MANETsis increasing rapidly. The mobile devices used in adhoc networks are usually resource starved and hence the close cooperation among *MHs* nodes may fulfill the increased computational and storage requirement of modern applications. Some major applications that useMANETsare processing of data gathered from distributed sensors, map update, location-aware services, communication in battlefields, search and rescue, disaster recovery, virtual classrooms, multi-user games, extending cellular network access, personal area network (*PAN*).

In MANETs, the topology of the system is dynamic because of the mobility and failure of nodes. Other characteristics of the MANETs arelow-bandwidth wireless links, limited battery power and the low processing power of *MHs*. Due to these features, the protocols developed for static networks cannot be used directly in MANETs. The popular research problems in MANETsare routing [24, 25], mutual exclusion [17], failure recovery [1], security [26], energy efficiency [27], and quality of service (*QoS*) [28]. However, the present work is focused on mutual exclusion.

3. Mutual Exclusion

The mutual exclusion problem is one of the highly researched problems in distributed computing. If a resourceis shared among several nodes, the contending nodes should access it in a mutually exclusive way. Thus, whenever a node wishes to use the shared resource, it must acquire exclusive control over it. The section of code used for the purpose is called critical section (*CS*). Among several competing processes it allows only one to enter *CS*, at any point in time. This is known as mutual exclusion (*ME*).

Several algorithms have been proposed for mutual exclusion in static networks [29-34] as well as mobile ad hoc networks [17, 35-42]. The solution to mutual exclusion problem can be categorized into two classes: permission-based[43] and token-based[44]. In token-based algorithms, a unique token is shared among the nodes. A node is allowed to enter the *CS* only if it possesses the token. While in a permission-based algorithm, the node that wants to enter the *CS* must obtain the permission from other nodes by exchanging messages.

A solution to the ME problem must satisfy the followingcorrectness properties [45]:

- *Mutual Exclusion (safety)*: At most one process is allowed to enter *CS* at any moment.
- *Deadlock Freedom*: If any process is waiting for the *CS* then in a finite time some process enters the *CS*.
- *Starvation Freedom*: If a process is waiting for the *CS* then in a finite time the process enters the *CS*.

The parameters to evaluate the performance of a mutual exclusion algorithm are as follows [45]:

(i) *Response time* - denotes the time interval a request waits for its entry to *CS* after its request messages have been sent out.

(ii) *Synchronization delay* - The time required after a node leaves *CS* and before the next node enters the *CS*.

(iii) *Throughput* - The rate at which the system executes requests for the *CS*.

(iv) *Communication overhead* - This denotes the average number of messages exchanged per *CS* entry.

(v) *Message size* - The average number of bits required to encode the messages exchanged.

4. Variants of Mutual Exclusion Problem

Over the years, following variants of themutual exclusion problem have been discussed.

(a) *Dining philosopher's problem*

In 1965, Dijkstra [46] presented a synchronization problem and named it as the dining philosopher's problem. In the dining philosopher's problem, some philosophers are seated around a table. The thinking philosopher may become hungry arbitrarily. The philosopher requires two forks (left fork and right fork) for each eating session. Each fork is shared between neighboring philosophers. If the hungry philosopher is successful in acquiring both the forks, it starts eating and put down both the forks when the eating session is over. Otherwise, he has to wait for the required forks. The dining philosopher's problem is essential to design a solution so that no philosopher is starved. The dining philosopher's problem can be considered as an abstraction of resource allocation and concurrency control problems in distributed systems.

(b) *k-mutual exclusion problem*

Fischer-Lynch-Burns-Borodin [47] defined *k*-exclusion problem as a natural extension of classical mutual exclusion problem. In a distributed system, there is a possibility that *k* identical copies of a shared resource (for example*k* user licenses of software) exist. Since each of these copies can be accessed by only one user process at a time; up to *k* user processes may be allowed to enter the *CS* simultaneously using these *k* identical copies.

(c) *Group Mutual Exclusion Problem*:Joung [48] formulated the group mutual exclusion (*GME*) problem, as a variant of the mutual exclusion problem, which deals with two contradictory issues namely concurrency and mutual exclusion. In *GME* problem, the requesting process announces a type or group along with its request. The processes, which are requesting for the same group may enter their *CS* simultaneously. However, the processes having conflicting requests (requesting different groups), must execute their *CS* in a mutually exclusive manner [49].

(d)*Group-k mutual exclusion (Gk-ME)*: In [50], Naimi-Thiare proposed group *k*-mutual exclusion problem. The *Gk-ME* problem is similar to group mutual exclusion problem. The only difference is that the concurrent accesses to some shared resource are allowed to at most *k* nodes, unlike group mutual exclusion where there is no such restriction on the number of nodes, which can concurrently access a shared resource. In both the problems, no two uniqueresources can be accessed simultaneously.

(e) *Group Mutual l-Exclusion (GM-l E) problem*: Vidyasankar [51], proposed the group mutual *l*-exclusion problem, which can be considered as a combination of *k*-exclusion problem [47] and the *GME*problem [48]. In contrast, with *GME* problem where only one group may be accessed at any point in time, in *GM-l E* problem, up to *l* resources (groups) can be available for simultaneous access. Hence, the processes interested in any one of these *l* groups may enter their respective *CS*. However, a process interested in a group other than these *l* groups has to wait.

(f) *k-out-of-M exclusion problem*:In*k*-out-of-*M* exclusion problem [52], *M* denotes the total number of resources available in the system whereas *k* denotes the number of resources requested by an individual process. The total number of requests by all processes should not exceed *M*. For example, there are *M* disks available in the system and a node may request at most*k* disks. *k*-out-of-*M* exclusion ensures that the desired number of resources is not exceeding *M* at any point in time.

(g) *Drinking philosopher's problem*: In [53], Chandy-Mishra proposed the drinking philosophers problem as a generalization of the dining philosophers problem. The drinking philosopher's problem essentially abstracts several synchronization problems in distributed systems. In it, the philosophers are considered to be placed at the vertices of an undirected graph *G*. A bottle is associated with each edge in *G*. There is an edge between two philosophers only if they share a bottle. Each philosopher can be in one of the following three states: (i) tranquil (ii) thirsty or (iii) drinking. A tranquil philosopher may eventually become thirsty and may require a subset of bottles for drinking. He may require a different subset of bottles for separate drinking sessions. The problem is to find a solution, which ensures that no philosopher remains thirsty, the philosophers interested in different set of bottles may drink simultaneously, and no two neighboring philosophers can use the same bottle simultaneously.

Conclusion

The mutual exclusion problem is one of the highly researched problems in distributed computing. If a resourceis shared among several nodes, the contending nodes should access it in a mutually exclusive way. Lot of research has been done on mutual exclusion and its variants in distributed systems and its variants. Still lot of efficient algorithms can be designed for these problems. The variants of mutual exclusion problem namely *k*-mutual exclusion (*KME*) problem, group mutual exclusion (*GME*) problem and *h*-group mutual exclusion (*h-GME*) problem need different approaches to handle the complexities of these problems.

[1] Kshemkalyani, A.D., Singhal M., Distributed computing principles, algorithms, and systems. Cambridge University Press, 2009.

[2] Lynch, N.A., Distributed algorithms, Morgan kaufmann, 2003.

[3] Tanenbaum, A.S., Maarten, V.S., Distributed systems principles and paradigms. Pearson Education, 2003.

[4] Ghosh, S., Distributed systems an algorithmic approach, Chapman & Hall/CRC, 2007.

[5] Misra, J., Pal, P., Bagchi, A., SPEAR: Design of a secured peer-to-peer architecture. In the Proceedings of the international conference on information systems security 2005, pp. 322-327, 2005.

[6] Ding, X., Yu, Q., Li, J., Liu, J., Hai, J., Distributed anonymization for multiple data providers in a cloud system, In the Proceedings of International Conference on Database Systems for Advanced Applications (DASFAA) 2013, pp. 346-360, 2013.

[7] Reynolds, C. J., Winter, S., Terstyanszky, G., Kiss, T., Greenwell, P., Acs, S., Kacsuk, P. K., Scientific Workflow Makespan Reduction through cloud augmented desktop grids, In the Proceedings of IEEE 3rd International Conference on Cloud Computing Technology and Science, pp. 18-23, 2011

[8] Vedantham, R., Zhuang, Z., Sivakumar R., Mutual exclusion in wireless sensor and networks, In the Proceedings of International Conference on

Sensor and Ad Hoc Communications and Networks (SECON'06), vol. 1, pp. 346-355, 2006.

[9] Ekiz, N., Salih, T., Kucukoner, S., Fidanboylu, K., An overview of handoff techniques in cellular networks. International Journal of information Technology, vol. 2, no. 2, pp. 132-136, 2005.

[10] Pandey,S.K., Mishra, R.B., Knowledge discovery and ontological based services on the grid, In the Proceedings of IEEE conference on parallel and distributed computing, China, pp. 1-6, 2005.

[11] Loo, J., Mauri, J.L., Ortiz, J.H., Mobile ad hoc networks: current status and future trends, CRC Press, 2012.

[12] Dijkstra, E.W., Cooperating sequential processes. Technological University, Eindhoven, The Netherlands, September 1965, reprinted in programming Languages, F. Genuys, Ed., Academic Press, New York, pp. 43-112, 1968.

[13] Stoller, S.D., Leader election in asynchronous distributed systems. IEEE Transactions on Computers, vol. 49, no. 3, pp. 283-284, 2000.

[14] Chandy, K. M., Lamport, L., Distributed snapshots: determining global states of distributed systems. ACM Transactions on Computer Systems, vol. 3, no. 1, pp. 63-75, 1985.

[15] Lamport, L., Time, Clocks, and the ordering of events in a distributed systems. Communications of the ACM, vol. 21, no. 7, pp. 558-565, 1978.

[16] Dhamdhare, D. M., Iyer, S. R., Reddy, E. K. K., Distributed termination detection for dynamic systems. Parallel Computing, vol. 24, no. 14, pp. 2025-2045, 1997.

[17] Badrinath, B.R., Acharya, A., Imelinski, T., Structuring distributed algorithms for mobile hosts, In the Proceedings of 14th International Conference on Distributed Computing Systems, pp. 1-28, 1994.

[18] Lee, W., Kang, S., Lim, S., Shin, M-K., Kim, Y-K., Adaptive hierarchical surrogate for searching web with mobile devices, IEEE Transactions on Consumer Electronics vol. 53, no. 2, pp. 796- 803, 2007.

[19] Benkrid, S., Bellatreche, L., Alfredo, C., A global paradigm for designing parallel relational data warehouses in distributed environments, Large-scale Data- and Knowledge Centered Systems Journal, vol. 15, pp. 64-101, 2014.

[20] Joshi, R.K., Harikrishnan, C.R., Ansari, M.H., Mobile agent abstractions: formulations and implementations, In the Proceedings of International Conference onMTCOORD, 2007.

[21] Swaroop, A., Singh, A.K., A token based group mutual exclusion algorithm for cellular wireless networks, In the Proceedings of International Conference on IEEE India Conference (INDICON), pp. 1-4, 2004.

[22] Roy, R.R., Handbook of mobile ad hoc networks for mobility models, Springer US, 2011.

[23] Agrawal D.P., Zeng Q.A., Introduction to wireless and mobile systems, Thompson (Nelson), 2007.

[24] Cabaniss, R., Madria, S., Rush, G., Trotta, A., Vulli, S.S., Dynamic social grouping based routing in a mobile ad-hoc network, In the Proceedings of International Conference on High Performance Computing Systems (HiPC 2010) Mobile data management 2010, pp. 295-296, 2010.

[25] Perur, S., Iyer, S., Characterization of a connectivity measure for sparse wireless multi-hop networks, Adhoc and Sensor Wireless Networks, vol. 3, no. 4, pp.311-330, 2007.

[26] Jain, S., Singhai, J., Chawla, M., A review paper on cooperative blackhole And grayhole attacks in mobile ad hoc networks, International Journal of Ad hoc, Sensor & Ubiquitous Computing, vol. 2, no. 3, pp. 71-80, 2011.

[27] Agarwal, M.M., Govil, M.C., Jhankal, A.K., Analysis of energy consumption in mobile ad-hoc routing protocols through route recovery and maintenance, International Transactions in Mathematical Sciences and Computers, vol. 5, pp. 121-130, 2012.

[28] Sarma, N., Nandi, S., Route stability based QoS routing in mobile ad hoc networks, Wireless Personal Communications vol. 54, no. 1, pp. 203-224, 2010

[29] Singhal, M., A Taxonomy of distributed mutual exclusion, Journal of Parallel and Distributed Computing, vol. 18, no. 1, pp. 94-101, 1993.

[30] Singhal, M., Manivannan D., A distributed mutual exclusion algorithm for mobile computing environments, Intelligent Information Systems (ICIIS'97) IEEE, pp. 557-561, 1997.

[31] Ricart G., Agrawala, A. K., An optimal algorithm for mutual exclusion in computer networks, Communication of the ACM, vol. 24, no. 1, pp. 9-17, 1981.

[32] Sopena, J., Arantes, L., Bertier, M., Sens, P., A fault-tolerant token-based mutual exclusion algorithm using a dynamic tree, In the Proceedings of International Conference euro-par 2005, LNCS-3648, pp. 654-663, 2005.

[33] Erciyes, K., Distributed mutual exclusion algorithms on a ring of clusters, In the Proceedings of the International Conference on computer Science and its Applications, LNCS-3045, pp. 518-527, 2004.

[34] Rahman, M.A., Akbar, M.M., A quorum-based distributed mutual exclusion algorithm for multi-level clustered network architecture, In the Proceedings of International Conference on WALCOM, pp. 121-135, 2007.

[35] Walter, J.E., Kini, S., Mutual exclusion on multi–hop mobile wireless networks, Technical Report, Texas A&M University, College Station, TX 77843–3112 TR97–014, 1997.

[36] Walter, J.E., Welch J., Vaidya N.H., A mutual exclusion algorithm for ad hoc mobile networks, Wireless Networks, vol. 7, pp. 585–600, 2001.

[37] Baldoni, R., Virgillito, A., Petrassi, R., A distributed mutual exclusion algorithm for mobile ad hoc networks, In the Proceedings of 7th IEEE International Symposium on Computers & Communications, pp. 539–544, 2002.

[38] Benchaiba, M., Bouabdallah, A., Badache, N., Ahmed-Nacer, M., Distributed mutual exclusion algorithms in mobile ad-hoc networks: an overview, (SIGOPS) ACM Operating Systems Review, vol. 38, no. 1, pp. 74-89, 2004.

[39] Derhab, A., Badache, N.A., Distributed mutual exclusion algorithm over multi-routing protocols for mobile ad-hoc networks, International Journal of Parallel, Emergent and Distributed systems, vol. 23, no. 3, pp. 197-218, 2008.

[40] Wu, W., Cao, J., Raynal, M., A dual-token-based fault tolerant mutual exclusion algorithm for MANETS, In the Proceedings of International Conference on Mobile Ad Hoc and Sensor Networks, pp. 572–583, 2007.

[41] Weigang, W.,Jiannong, Ca., Jin, Y., A fault tolerant mutual exclusion algorithm for mobile ad hoc networks, Pervasive and Mobile Computing, vol. 4, no. 1, pp. 139-160, 2008.

[42] Zheng, W., Song, X.L., Meian L., Ad hoc distributed mutual exclusion algorithm based on token-asking, Journal of Systems Engineering and Electronics, vol. 18, no. 2, pp. 398-406, 2007.

[43] Saxena, P.C., Rai, J., A survey of permission-based distributed mutual exclusion algorithms, Computer Standards & Interfaces, vol. 25, no. 2, pp. 159-181, 2003.

[44] Swaroop, A., Singh, A.K., A study of token-based algorithms for distributed mutual exclusion, International Reviews on Computers and Software, vol. 2, no.4, pp. 302-311, 2007.

[45] Swaroop, A., Efficient Group mutual exclusion protocols for message passing distributed computing systems, Doctoral Diss., NIT, Kurukshetra, India, 2011.

[46] Dijkstra, E.W., Hierarchical ordering of sequential processes, ActaInformatica, pp. 115-138, 1971.

[47] Fischer, M.J., Lynch, N.A., Burns J.E., Borodin A., Resource allocation with immunity to limited process failure, In the Proceedings of 20th annual symposium on Foundation of Computer Science, pp. 234–254, 1979.

[48] Joung, Y.J., Asynchronous Group mutual exclusion (extended abstract), In the Proceedings of 17th annual ACM symposium on Principle of Distributed computing, pp. 51-60, 1998.

[49] Joung, Y.J., Asynchronous group mutual exclusion. Distributed Computing, vol. 13, no. 4, pp. 189-206, 2000.

[50] Naimi M., Thiare O., A Group *k*-Mutual Exclusion Algorithm for Mobile Ad Hoc Networks, In the Proceedings of the International Work-Conference on Artificial Neural Networks (IWANN), pp. 58-66, 2009.

[51] Vidyasankar, K., A simple group mutual-*l* exclusion algorithm, Information Processing Letters, vol. 85, no. 2, pp. 79-85, 2002.

[52] Raynal, M., A distributed solution to the *k*-out of-*M* resource allocation problem. In the Proceedings of the International Conference on Computing and Information: Advances in Computing and Information, LNCS-497, pp. 599-609, 1991.

[53] Chandy, K.M., Mishra J., The drinking philosopher's problem, ACM Transactions on Programming Languages and Systems, vol. 6, no.4, pp. 632-646, 1984.