Chapter-9

Conclusions, General Discussion and Recommendations for Future Works

9.1 Introduction

Although a brief conclusion has already been presented in each of the previous chapters, a comprehensive discussion and conclusion on the work carried out under the study has been given in this chapter. The literature review reveals that considerable amount of work has been done on Distributed System Level Fault Diagnosis in addition to the present study, yet there is quite a lot of scope for the extension of author’s work. Therefore, some guidelines for extension of author’s work have also been laid down in this concluding chapter.

9.2 General Discussions and Conclusions

9.2.1 The distributed system level diagnosis algorithms discussed in the present work will provide a comprehensive idea about various issues, which are to be considered in developing such algorithms. Since “System level diagnosis” is one of the steps in the process of building “Distributed fault-tolerant systems”, reliability of such a system depends heavily on proper functioning of the diagnostic algorithm.

9.2.2 Based on the review of literature, as discussed in the Chapter 2, the following observations have been made:

1. In arbitrary networks, distributed algorithm for fault diagnosis uses parallel dissemination of fault event information to minimize the information latency in the network. Algorithm makes evident the tradeoff between information latency
and message overhead. In networks consisting at least partially of point-to-point communication links (which is the case for the majority of arbitrary topologies), it is argued that the increased message overhead that leads to an optimal information latency is tolerable. [Rangarajan et al, 1995]

II. HeartbeatComplete algorithm has a latency of approximately one heartbeat transmission round and a state holding time of about half a round. Algorithm for not-completely connected networks, ForwardHeartbeat has both latency and state holding time equal to approximately one round. Thus, both algorithms are capable of very fast propagation of events and they work in highly dynamic environments. [Subbiah and Blough, 2004]

III. Though the general idea of IP multicast is simple, the message delivery is purely best-effort and there is no guarantee that the data will reach all the multicast receivers. The loss of messages may be due to congestion, failures of nodes, failures of links, etc. To address these concerns, an enhanced category of multicast IP called reliable multicast has been evolved. It defines mechanisms to monitor host population, congestion, transfer speeds, and data loss on a per-host basis, and mechanisms to counter the loss, thus providing a more reliable multicast environment. [H. Eriksson, 1994]

IV. The Totem single-ring protocol provides reliable totally ordered multicasting of messages to processes in process groups over a single local-area network (LAN) using a logical token-passing ring. The protocol provides two levels of message delivery: delivery in agreed order and delivery in safe order. [Jia et al, 1996]

V. A tree-based reliable multicast protocol is designed to provide multicast reliability that scales to a large receiver population. TRAM ensures reliability by using a selective acknowledgment mechanism and scalability by adopting hierarchical tree-based repair mechanism. [Chiu et al, 1998, 2000, 2002]
VI. REED (Robust Energy-Efficient Distributed clustering protocol) for clustering sensors deployed in hostile environments in an interleaved manner with low complexity. REED can be easily incorporated into any clustering protocol. REED constructs $k$-fault-tolerant clustered networks if certain conditions on node density are satisfied. [Younis et al, 2004, 2005]

9.2.3 In Chapter 3, Distributed System-level diagnosis has been discussed. It allows the fault-free components of a fault-tolerant distributed system to determine which components of the system are faulty and which are fault-free. The time it takes for nodes running the algorithm to diagnose a new event is called the algorithm's latency.

9.2.4 In Chapter 4, a distributed system-level diagnosis algorithm for arbitrary network topologies has been presented. Even though it works for arbitrary networks, it assumes no link failures. A newly repaired node can rejoin the system without relying on other nodes to the first detected, that it has been repaired; equivalently, faulty nodes do not have to be periodically tested.

9.2.5 In Chapter 5, Heartbeat algorithm for dynamic fault environments is presented. The notion of Bounded correctness strengthens the properties of distributed diagnosis in the presence of dynamic failures and repairs compared to existing algorithms. Bounded correctness allows arbitrary failures and/or repairs as long as two consecutive state changes on a single node are not too close together in time. All nodes can change state at the same time or a cascade of status changes can occur, while maintaining a notion of correctness at all times. HeartbeatComplete algorithm handles these behaviors effectively in completely-connected networks.

9.2.6 In chapter 6, modified algorithms have been considered for both Totem Ring Multicast Protocol and Tree Based Reliable Multicast protocol. In the Ring approach by using the virtual token and logical clock, the algorithm can achieve the total ordering of multicast messages and atomicity efficiently. It can detect and recover from communication faults without token loss problem. Two design improvements to TRAM
have been tested and the modified TRAM protocol is implemented under similar conditions to TRAM. The comparison brings out that both protocols are scalable in latency in the range in which the experiments are done. Modified TRAM incurs a minor latency penalty in failure-free conditions. If constraints are enforced on the intermediate nodes, then Modified TRAM can enforce the conditions and yet localize the disruption to the system due to a few slow or malicious nodes. Modified TRAM also includes an algorithm for pruning nodes deemed too slow or suspected to be malicious. Hence author has concluded by saying that the modified TRAM proves to provide an efficient retransmission scheme that minimizes the resource utilization at intermediate hosts and localizes the effect of slow and malicious receivers on normal receivers.

9.2.7 Clustering Protocol (REED) in sensor networks is taken into account in chapter 7. The primary advantage of this approach is that the node quickly finds an alternate head and alternate communication paths, without waiting until re-clustering is triggered. The REED design, which interleaves the selection of cluster heads for different overlays, has an important advantage (in addition to low complexity): it eliminates any need for synchronizing the start of each cluster head overlay construction. Failures are implicitly detected without extra overhead if reactive routing is used. Regular nodes are not part of the routing infrastructure and their failure does not impact network connectivity.

9.3 Recommendations for Future Work

9.3.1 Future Work for arbitrary network topologies
The distributed system-level diagnosis algorithm for arbitrary network topologies which is presented in Chapter 4 can be extended in several ways as follows:

(i) Strengthening the algorithm to detect the occurrence of the simultaneous failure of a group of nodes that are in a jellyfish test configuration. This can be accomplished, by scheduling periodic, but less frequent, tests of all of the neighbors of fault-free nodes.

(ii) Extension of the fault model can be done in such a way, to allow faulty nodes to issue spurious or even malicious messages. In this case, the node
receiving a message should first check the validity of it, by performing a test on message sending node to know its status similar to the fault model presented in chapter 4. If the message-sending node is faulty, then it discards the message, otherwise accepts it or to acknowledge a new message after the time-out has expired;

(iii) Extension of the fault-model to handle communication link failures
(iv) Extension of these results to other application domains, such as distributed job scheduling, where it would be useful to know whether a given node is faulty or too busy and to the mobile computing (to main distributed location directories).

9.3.2 Future Work for Distributed Diagnosis in Dynamic fault environments
Similarly as in Chapter 5 the HeartbeatComplete algorithm, Forward Heartbeat algorithm may also be another effective fault diagnosis tool for arbitrary network (which is not completely connected) in dynamic fault environment, which could not be incorporated here in present work. Algorithms of bounded correctness for completely-connected network and not-completely-connected network are capable of very fast propagation of events and they work in highly dynamic environments.

9.3.3 Limitation and Future Work for Reliable Multicast Protocols
(i) Single sender: All the observation and implementations have been made based on the single sender multiple receiver scenarios. Thus the implementation here considers that the trees do not overlap each other.
(ii) Reliance on TTL: Here all the repair heads have used the TTL (Time To Live) values. Thus, extension of this work can be done for networks where there is no reliance on TTL values by redefining the criteria to identify repair heads.

9.3.4 Future work for Improvement / Modification of clustering protocol in sensor networks
Parallelization of intra-cluster and inter-cluster communications can be recommended by assigning two different CDMA (Code Division Multiple Access) codes: one for intra-
cluster communication and the other for inter-cluster communication so that both can proceed simultaneously. At the inter-cluster level, contention-based MAC (Medium Access Control) schemes, such as the 802.11 DCF (Distributed Coordination Function), can be used. At the intra-cluster level, it is typically easy to synchronize a regular node with its cluster head(s) using techniques. Intra-regional synchronization facilitates constructing TDM (Time Division Multiple access) schedules for each cluster. Therefore, after the $k$ overlays are constructed, every node contacts each of its $k$ cluster heads to be assigned a time slot in the TDM frame of the cluster head.