International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:1, 2014

TTCN-3 Based Conformance Testing of a Node Monitoring Protocol for MANETs

Mallikarjun B. Channappagoudar, Pallapa Venkataram

***Abstract*—**As a node monitoring protocol, which is a part of network management, operates in distributed manner, conformance testing of such protocols is more tedious than testing a peer-to-peer protocol. Various works carried out to give the methodology to do conformance testing of distributed protocol. In this paper, we have presented a formal approach for conformance testing of a Node Monitoring Protocol, which uses both static and mobile agents, for MANETs. First, we use SDL to obtain MSCs, which represent the scenario descriptions by sequence diagrams, which in turn generate test sequences and test cases. Later, Testing and Test Control Notation Version-3 (TTCN-3) is used to execute test cases with respect to generated test sequences to know the conformance of protocol against the given specification. This approach shows, the effective conformance testing of the distributed protocols for the network with varying node density and complex behavior. Experimental results for the protocol scenario represent the effectiveness of the method used.

***Keywords*—**Conformance Testing, FSM, Mobile agent, TTCN, Test sequence.

I. Introduction

MOBILE ad hoc network is a wireless, self-configured, infrastructure less network of mobile nodes. The nodes are highly mobile, which makes the application running on them face network related problems like scarcity of resources, i.e., energy, bandwidth, buffer degradation, and intermittent disconnection. Hence the overall performance degrades. The traditional network monitoring protocol like Simple Network Management Protocol (SNMP) could be preferred for usage to observe the performance of the network, but there might be bottleneck of information and load processing at manager of the network. So node monitoring using mobile agents may be preferred as a solution to reduce the network overhead and improve the performance of the network.

II. Mobile Agents in Network Management

Performance, fault, configuration and accounting management are the diversified areas of network management where mobile agents play an important role [1]. The usage of mobile agents gives the solution to the scalable problem in centralized network management. The use of mobile agents will reduce the network overhead and response time in MANET compared to other techniques and mobile agents are considered as the important tools for the middleware management [2], [3]. The whole task of management is transferred to the agents who carry out the management function in an autonomous and distributed way [4], [5]. The efficiency of the network management is increased by using mobile agents [6].

To design and develop a mobile agent based node monitoring protocol, it must be tested exhaustively to verify and validate the functionality of protocol.

Testing is basically expressed in two ways. *Firstly*, given the diagram of finite state machine (FSM) but information about the present FSM state is not known. Present state is realized by applying the sequence of input to the machine and by interpreting the behavior of its I/O information.

Secondly, a FSM which is provided is considered as black box, with which the I/O information can be observed. The main aim is to test, to know whether the implementation is conformed to the given standard specification. Moreover, it is referred to as *conformance testing or fault detection technique*. Test sequence which finds the solution for this difficulty is referred to as *checking sequence*.

A. Conformance Testing

This represents the main testing process of a protocol to check whether the implementation of the protocol adhere to their given particulars or not. Conformance testing of a protocol can be achieved by testing protocol against the test sequences. Test sequences generation using particulars of formal protocol is hard and needs long time duration. Recent trends tell us various methods exist to obtain the test sequences from formal techniques called FSMs automatically. These particulars represented in a Formal Description Language such as, Specification and Description Language (SDL) together with Message Sequence Charts (MSCs).

MSC is a graphical language, which explains the interaction behavior between entities of system with the environment. These are basically used to describe and define the behavior of communication with the help of message exchange and also with certain procedure calls between many distributed components. MSCs are commonly preferred, for test cases in particular [7]. MSCs basically required for defining and explaining the flow of communication among the entities of the system in which messages play an important role.

B. Necessity of Conformance Testing

The respective product users or their organizations on their behalf, usually do product testing to check whether they are functioning correctly or not. Any of the product prior connection to the network is cross checked, to avoid faulty functioning of a network due to the products which are

Mallikarjun B. Channappagoudar is with the Indian Institute of Science, Bangalore, 560012, India (e-mail: mallikarjun.cp@ece.iisc.ernet).
Pallapa Venkataram, is with the Indian Institute of Science, Bangalore, 560012, India (e-mail: pallapa@ece.iisc.ernet.in).

implemented wrongly.

Conformance testing of any protocol can be achieved using Testing and Test Control Notation version-3 (TTCN-3), which is the specific tool well suited for the systems which has several parallel testers, which is the basic need in testing of monitoring function. The testing details are represented by the test architecture where the required IUT is to be tested.

TTCN tool is recommended by ISO to express the test suite in an abstract way. TTCN-3 is an extended version of TTCN-2; it has the many test components (tc) in the environment which are simultaneously active to execute the test cases [8]. Fig.1 represents the architecture with an MTC (Main Test Component and many PTCs (Parallel Test Components). The test components in coordination with CPs (Coordination Points) inform about the messages. Interfacing between these test components and environment is done through interface called PCOs (Point of Control and Observations).

Conformance testing of the implementations of a node monitoring protocol is considered as the prime importance. Testing the whole scenario i.e., conformance of behavior of the protocol to the particulars, using testing tool called TTCN-3 with the help of MSCs obtained for FSM of a protocol from SDL.

For the analysis of concept, we say the monitoring entity g; there will be an r different concurrent behavior which appears at i_1, i_2 ...i_r (different interaction points) and the $t_1 \parallel t_2 \parallel \ldots \parallel t_r$ are shown as behaviors. If these behaviors are confirmed with particulars of s, then g is said to be conformed implementation.

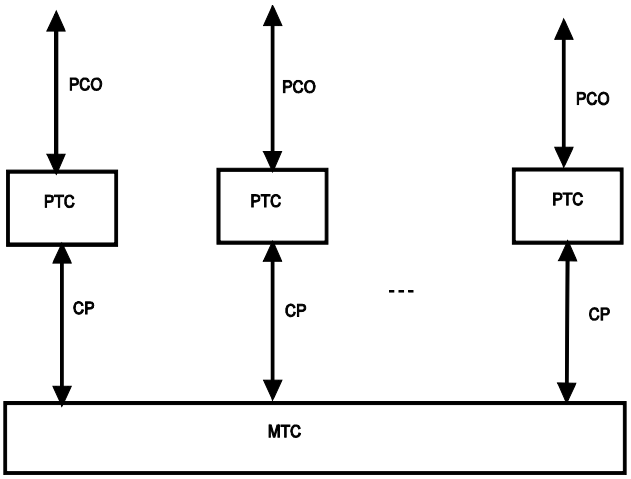

Fig. 1 Test component model of concurrent TTCN

In network management, monitoring the nodes of a network is considered as an important and crucial process to keep both the network and application work properly. Due to dynamic network topology the complexity of node monitoring in turn increases in mobile ad-hoc networks (MANETs). There is a challenge due to dynamicity, i.e., the random node movement and resource scarcity leads to a challenge in monitoring the nodes in a MANET.

We proposed node monitoring protocol for MANETs in [9], briefly explained in section [IV], uses both static as well as mobile agents. These deployed agents will manage the network through two tier architecture. Static agent will reside at the central node, whereas the mobile agents will migrate to the nodes in different clusters of the zones respectively, periodically collects the node status information and provides high level information to the static agent by analyzing the raw information at the nodes. This automatically decreases the network traffic and reduces the workload of the central node to the desired level. The static agent at the central node is available with high level information and in coordination with other modules.

In our proposed work using the concept of deriving the test sequence from the MSCs of the protocol generated by SDL leads to conformance testing of generated test cases of proposed node monitoring protocol using TTCN-3 is accomplished with a scenario, taken a sector (cluster) as a part with few nodes into consideration.

Rest of the paper is organized as follows. Section II discusses some of the related works. Section III gives some definitions related to FSM and testing problems. Section IV discusses about a node monitoring protocol for MANETs. Section V discusses the test sequence derivation. Section VI discusses experimental results. Section VII followed by conclusions.

III. Some of the Related Works

There has been various works on automatic generation of test cases using formal approaches are investigated [10]-[12]. In [13] shows the importance of formal methods. SDL is the language which is generally preferred for formal representation of the protocol specifications [14] and a graphical language referred as Message Sequence Chart (MSC) [15]. Later TTCN-3 is proposed to benefit its features from formal approaches [16]. The concept of testing in real scenario using both tools called TTCN-3 and MSC by SDL are presented in [17]. Ebner presented the work related to test cases, justified that these TTCN-3 test cases basically are obtained from SDL with concept of MSC and discussed about the conversion of MSC into the TTCN-3 elements in [19]. Later the work imposed into verifying and testing mobility protocol [20] by generating the test cases using MSCs, which follows the concept of Ebner in [19]. In above approaches sufficient work is carried out theoretically for conformance testing of the protocol using TTCN-3, i.e., they give methodology to carry out the experiment. In our work as we developed our node monitoring protocol for MANETs [20]for which we use SDL to obtain MSCs that, represents the scenario descriptions by sequence diagrams, which in turn generate test cases and test sequences. Then TTCN-3 is used to execute the test cases with respect to generated test sequences to know the conformance of protocol against the given specification. The method can be carried out for varying node density in the network.

IV. Definitions

In this section we will discuss some of the definitions concerned to FSM and some testing phases to tackle the difficulties in conformance testing of the given FSM.

Mealy machines are regularly used for modeling the systems of finite number of states, which gives the respective outputs, when there is a transition of state due to input reception. Most regularly considered formal models to model the system in various areas are called *Finite state machines (FSM)*, whose coverage area includes circuits with memory (sequential type), some types of programs and most rigorously in communication protocols [21]-[24]. The accuracy of the system is realized if it undergoes testing process to check their current working status and its conformance.

A. Finite State Machine (FSM)

FSM M is represented as quintuple as per the requirement and given by

$$A = (I, O, S, T, F)$$

where I is input, O is output and S represents states correspondingly, which are obviously the finite sets.

T: $S \times I \rightarrow S$ represents the transition of the state.

F: $S \times I \rightarrow O$ represents the function as output.

Since S is the set, in which s is the present state, which goes to next state T(s,i) after receiving the input i of I and produces the respective output F(s,i). In protocols, the given systems should be defined properly and even non deterministic. Incomplete details and unobserved transitions may lead to wrong interpretation of timer and unpredictable situation may happen [25].

Firstly, some specific definitions in FSM are stated as follows with brief example details.

1. *Machine Equivalence:* To each state in one machine there will be a state which is equivalent in other machine, and then those two machines are called as equivalent. To describe this if two states of different machines A and A', are provided the same set of inputs and outputs, represented as A = (I, O, S, T, F) and A' = (I, O, S', T', F') then both A and A' are equivalent.
2. *Isomorphism:* Accordingly homeomorphism from A to A' is a transforming Φ from S to S' in such a way that for each and every state s in S, for each and every input symbol a in I, it holds T (Φ(s), a) = Φ (T(s, a)) and F'(Φ(s), a) = F(s, a). It is referred to as isomorphism if Φ is a bijection. A and A' must contain equal states and they must be identical other than the states which are renamed. If there is an isomorphism between the two machines then they said to be isomorphic. Obviously the two FSMs which are isomorphic are equivalent, but the equivalent machines might not be isomorphic.
3. *Minimization:* It represents the states of the machine which are minimum in number. If no two states in a machine are equivalent then it can be minimized [26].

Secondly, definitions associated with testing phases are discussed.

B. Testing Phases

The following four phases usually considered while testing an FSM.

1. Homing/Synchronizing Sequence: It helps to obtain the final machine state after the test.
2. Identification of State: It helps in finding the initial state of the machine.
3. Verification of State: If it is understood that the machine is at initial state, then it helps to do verification to know whether it is in that particular state.
4. Machine Verification /Fault Detection /Conformance Testing: complete description of other machine B if A is given as particular machine and determination of B is equivalent to A or not.

There are four steps involved in the problem statement, they are,

1. Preliminaries: Our motivation is to test to check machine B implementation conforms to the specification of machine A. Surely; the solution to problem without any pre-assumption is not possible. For any test sequence it is easy to construct machine B that may not be equivalent to machine A, but for the given test sequence it generates the same outputs as that of machine A. Few assumptions to be made are:

1. Machine A specifications are strongly connected
2. MachineA has states which are minimal in number (minimized),
3. Implementation of machine B will not change at the time of experiment, when input is same as that of machine A,
4. Machine B should not have more number of States, than machine A.

2. Status Messages and Reset: The present state of the machine is given by status messages. To know the details we need to see the contents of the register which has the storage of states of a sequential circuits and communication protocols, the value present reflects the state.

3. Distinguishing Sequences: The unreliable message status is nothing but the distinguishing sequence, which is obtained by a particular polynomial time algorithm to build the test sequence [26] of required length. It gives various outputs for the state, but does change of state.

4. Identifying Sequences: The sequences by [27] (called as locating sequences in [26]), which does state identification during the middle of the execution process. So usually the testing sequences are derived from the identifying sequences [26], [27].

V. Node Monitoring Protocol

We briefly describe the developed node monitoring protocol in this section and realize the transitions using FSM model. The protocol uses a mobile agent (MA) to collect the information regarding some of the network health conditions (buffer, bandwidth, and energy), mobility rate and misbehavior of the nodes in the network. The monitoring protocol assumes that given MANET is divided into logical zones and deploys a mobile agent for each zone periodically. The coordinate's technique is used to divide the MANET into approximate

number of zones and each zone into required number of clusters with respect to central node where SA resides as shown in Fig. 2.

The node monitoring protocol runs at the node, which is rich in resources, of a MANET and where static agent (SA) is deployed for the collection and interpretation of network status information, and MA for collecting the node status information from a particular zone periodically to monitor the health conditions, mobility rate and misbehavior of nodes in the clusters of particular zone.

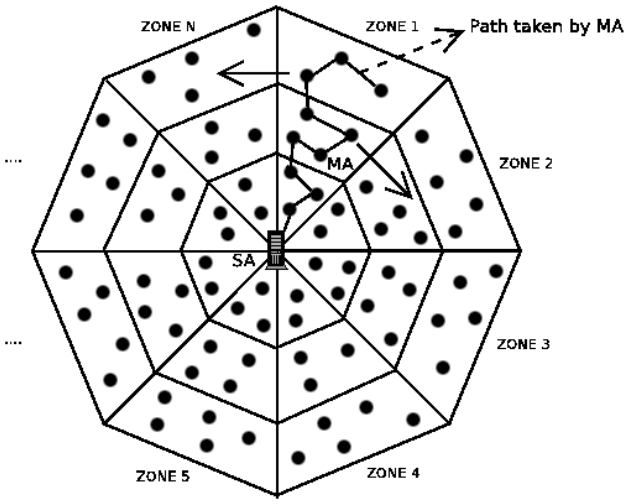

Fig. 2 Zone formation with respect to static node

The main advantage of MAs in the proposed protocol is to reduce the management overheads, network traffic and to overcome increase in response time in monitoring the nodes of a MANET.

A. Architecture

The design of the node monitoring protocol includes two processes:

1. Central Monitoring Process: The main process of the protocol which runs at the selected central node, where SA runs to collect and interpret the information regarding the network status.
2. Zone Monitoring Process: It is the status monitoring process which runs in migrated MA. The MA chooses the node in the cluster which is rich in resources and runs on that node. If there is lack of resources in the selected node where MA is residing, then it migrates to another node in the same cluster which is rich in resources. Initially, SA creates a MA and dispatches to a zone in order to monitor health conditions of nodes of that zone periodically. MA which carries status monitoring process migrates into the cluster and monitors the resources (energy level, throughput, delay, packet delivery ratio etc.), mobility and cooperation level of nodes in every cluster of the zone. The MA travel in the cluster in such a way that, in its vicinity, the nodes available in the sector would be monitored. As MA collect the cluster information instead of moving to next cluster immediately, first it sends the monitored high level information to SA. Later SA interacts with the RCE (Rate of Change of Energy) and Mobility estimation module, Misbehavior detection module, Resource status identifier, and History Analyzer for effective monitoring of the nodes in a MANET.

MA interaction with Nodes: With the help of interaction protocols (Dutch Auction Protocol, Contract Net Protocol) the MA interacts with nodes at the clusters as all the nodes which forms the network are agent platform compliant. In case if the nodes which enter the cluster does not have support agent platform or not compliant with, then MA will communicate with them using remote procedure call (RPC) mechanism. Network management protocols, SNMP (in data network) or CMIP (in telecommunication network) can be used by mobile agents to interact with the nodes.

The Architecture of the main process of a node monitoring protocol is shown in Fig. 3.

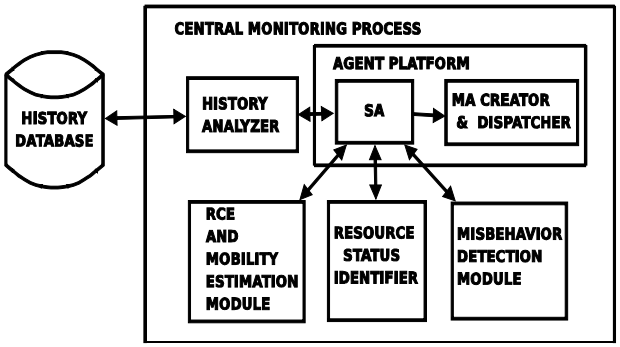

Fig. 3 Architecture of main process of node monitoring protocol

Status monitoring process is a functional unit to gather the information regarding the status of the network, interpret and analyze that information and inform it back to MA, if any variation is observed. Status collector gathers the information regarding the status of the network through MA and the Resource parameter computation module computes the performance metrics like throughput, delay, and cooperation level at the node based on the collected raw information and reports it to the Reporter Module, which, later, informs it to the MA. As shown in Fig. 4.

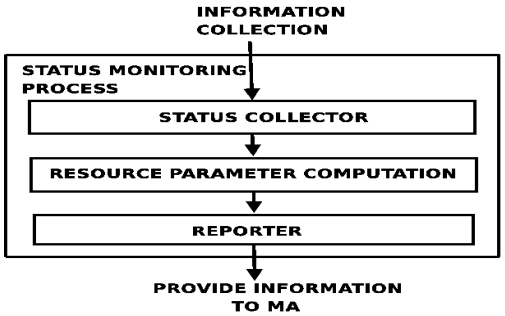

Fig. 4 Architecture of Status monitoring process of node monitoring protocol

The flow chart of the functioning of the node monitoring protocol to monitor the one entire zone of the MANET is shown in Fig. 5.

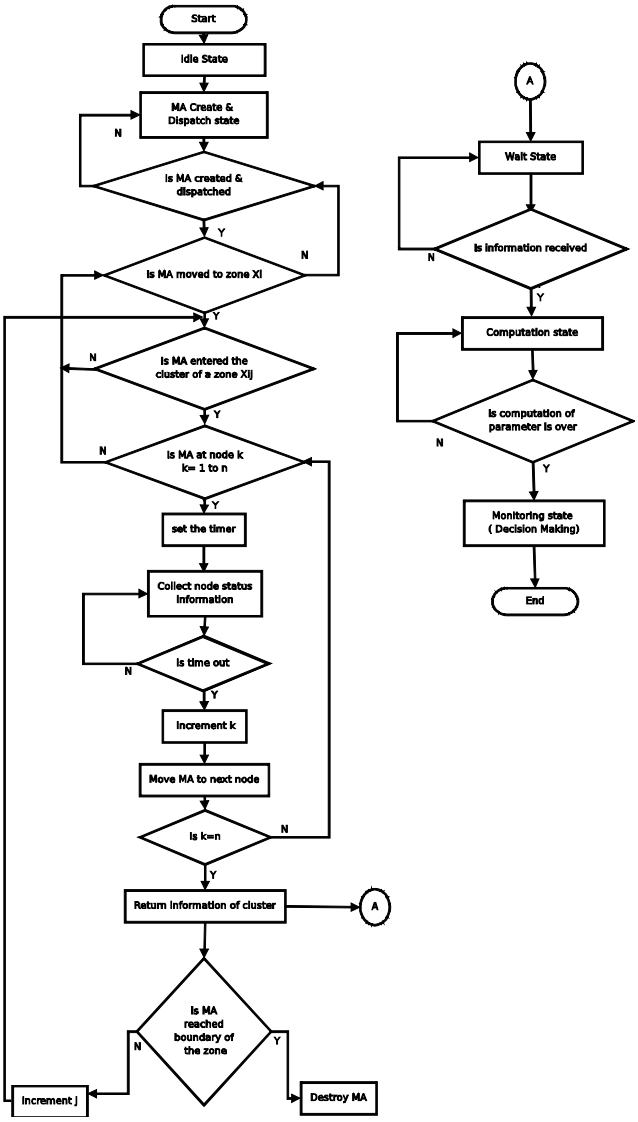

Fig. 5 Flow chart for Node monitoring protocol

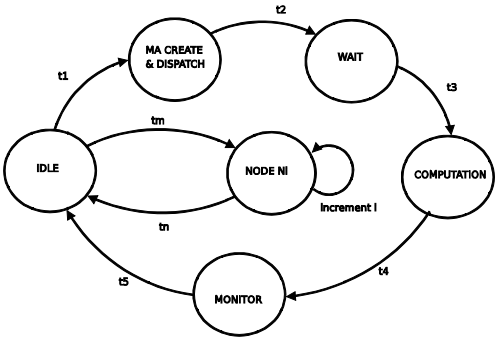

Fig. 6 FSM for Node monitoring protocol

VI. Test Sequence Derivation

Test architecture of the proposed node monitoring protocol has to respond to the behaviors at different interaction points. The architecture which is conceptually represented in Fig. 7 has the related components as shown.

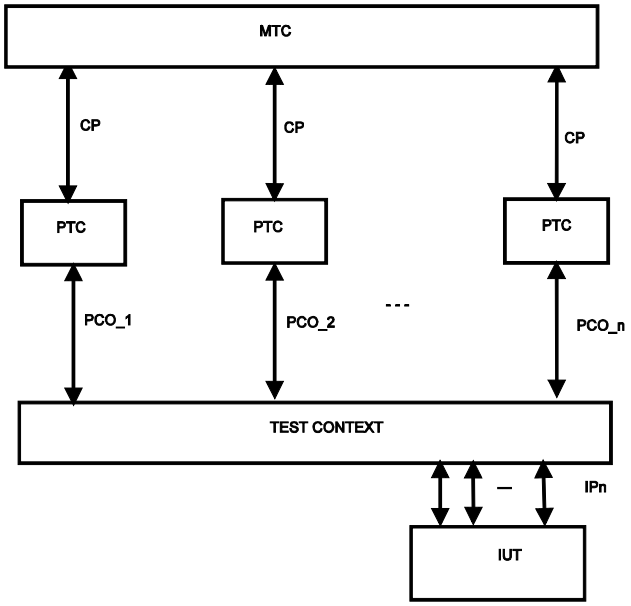

Fig. 7 Test Architecture (a conceptual view) for node monitoring protocol

MTC is the main test component which makes the assignment of all PTCs, which manages all coordination points and point of control and observations. A make operation connects the PTC to a MSC behavior tree. Parallel test components do the execution of assigned behavior of MSC tree simultaneously along with the MTC. Finally the MTC is responsible for terminating the operation of PTC. The executable program of the TTCN deals with the concept of behavior and configuration part. It consists of the components like test system, MTC, different PTCs, PCOs, CPs for interaction and Ports through which communication takes place. The process includes,

1. Initially the required components are defined.
2. The configuration of the test is done by connecting the MTC and PTC components through ports, and
3. Mapping process for ports takes place to interact with Test System Interface.

It generally means interacting with PCOs and CPs to communicate with SUT (IUT).

The other main part of the test includes the module which has the definition and control part. They include in detail about data types and methods. The main test component defines the behavior of the test case, along with PTC functionality if required. Usually the PTCs are created by MTCs as and when required. Individual test cases are to be monitored by the control portion of the module.

TTCN-3 test case generation based on MSC, and the MSCs are generated using SDL [20]. Even though, there is a use of

formal techniques for protocol representation, but there may be many implementations possible, which are difficult to handle. However, there is a requirement for testing an implemented protocol with its specification. Hence, testing process of the protocol is effectively handled through the test-sequences which are derived from the protocol specifications using MSCs. Actually test sequence is the set consists of input/output combinations which effectively obtained by the specifications represented in any form. These test sequences are applied as inputs to an IUT which is always a black box. The results of the IUT represent the outputs which in comparison with the output part of test sequences. However, the verdict assigned informs whether the implemented protocol conforms to its specifications or not.

TTCN-3 generates test cases using SDL and MSC particulars. It is taken into fact by the proposed work of Ebner [19] on translation of MSC elements into TTCN-3 statements. In adverse to the Ebner technique in which message sequence chart is realized with Unified Modeling Language (UML). Even the author proposed a tool for conformance of UML specification in [18]. Our proposed idea is to generate MSC automatically using SDL model. Conformance testing of the protocol is accomplished on the given protocol by generating the test cases from the obtained MSCs by SDL and testing using the test sequences from MSCs as follows,

1. Using the particular SDL, the simulations automatically generate the MSCs. The generated MSCs with associated simulation useful for testing particular system components. However, the MSC obtained during complete simulation helps in testing the complete system scenario and components for a particular given purpose of the test.
2. Based on the architecture of the test, the given MSC is converted into the MSC test cases by IUT and PCO elements respectively.
3. The test cases of TTCN-3 are obtained from MSC test cases with the concept of translation. The concept of translation includes two levels,

*Level-I:*TTCN-3 data types, the TTCN-LINK of TTCN-3 tool are used to automatically obtain the declarations of TTCN from the SDL requirements.

Level-II:For the concept of distributed test configuration in TTCN-3, we used the concept of mapping elements of SDL into TTCN-3

1. The SUT (IUT) is selected from the represented SDL segments (block) in SDL tool and each segment represents a process.
2. One or More SDL segments as the MTC (Main test component), and other remaining segments of SDL becomes the Parallel test components (PTC). Consequently, the channels of SDL between the segments of SDL (remaining) referred as CPs of TTCN-3 and are with the System. But channels of SDL between the IUT (chosen SDL segments) and other SDL segments are defined as ports of TTCN-3.
3. The respective messages of input and output exchanged through TTCN-3 ports are the SDL signals accomplished by channels of SDL.

TTCN-3 test case behavior: To obtain the behavior of test case in TTCN-3, we map the test case in MSC to test case in TTCN-3 as proposed in [19].

We need not to derive the test cases from TTCN-2, since TTCN-3 has the feature of distributed test configuration and especially for synchronization of parallel test components. PASS, FAIL or INCONC are the verdicts to be assigned to every leaf of the MSC test tree, when these MSCs test cases are mapped to different test cases in TTCN-3. Finally, these test cases are gathered with some rules, and are mapped to an overall TTCN-3 test suite.

Main test component remains as the control section of the execution of test. Usually interpreter provides the executable test case to the MTC, and main test component executes the test case by going through the test case tree structure. The main test component generates parallel test components which are processes involved in other functionality like, listening to ports. An event takes place called ``make'', so when MTC executes ``make`` event, then particular parallel test component is active and run test behavior of MSC tree assigned by MTC. Every parallel test component has the log file for the execution of respective test case of the tree structure. Later the main test component collects all log files and informs to result section. The Axioms followed are as listed below.

Axiom-1: Every leaf of MSC is outlined to one TTCN-3 test case.

Axiom-2: In one MSC test case, it may have several forms, $p_1, p_2 \ldots p_n$ then attach MSC-tree i+1 to MSC-tree i.

TR-M: $p_1, p_2 \ldots p_m$

Axiom-3: An action in a form p, if $h_1 >> h_2 >> \ldots >> h_n$, then h_1, h_2... hn are various behaviors and it is effectively completed ifh_{i+1} does occurs just after effective completion of h_i.

TR-M:Specifications of Behavior of M
MAKE (SP1, TR-SP1)
INITIATE-T1
P_1? M_1
OVER (SP_1) PASS
? TIME_OUT-T_1 FAIL
TR-SP1: Specifications of behavior of SP_1
h1, h2... hn
P_1! M_1

*Axiom-4:*An action in a form p, if h1 [] h2 []... [] hn, so any one among them is matched if the behavior of the other one happens.

TR-M: Specifications of Behavior M
MAKE (SP_1, TR-SP_1)
...
MAKE (SP_n, TR-SP_n)
INITIATE-T1
P_1 ? M_1
OVER (SP_1) PASS
? TIME_OUT-T_1 FAIL
TR-SP1: Specifications of Behavior of SP_1
h_1

$P_1!\ M_1$
h_2
$P_1!\ M_1$
...
h_n
$P_1!\ M_1$

Axiom-5: In one of the action c from the relation p, if $h_1 \parallel h_2 \parallel ... \parallel h_n$ then h_1, h_2...hn are initiated simultaneously and the parallel tree completes only if h_1, h_2...h_n complete.

TR (tree)-M: Specifications of behavior of M
MAKE (SP_1, TR-SP_1)
...
MAKE (SP_n, TR-SP_n)
INITIATE-T_1
P_1? M_1
OVER (SP_1)
INITIATE-T_2
P_2? M_2
OVER (SP_2)
INITIATE-T3
...
P_n?M_n
OVER (SP_n) PASS
? TIME_OUT-T_{n+1} FAIL
P_n?M_n
OVER (SP_n)
...
? TIME_OUT-T_2FAIL
P_n?M_n
OVER (SP_n)
P_{n-1}? M_{n-1}
OVER (SP_{n-1}) PASS
? TIME_OUT-T_1 FAIL

Axiom-6: Thetimers, constants, variables declared are represented as Timer, Cons and Var in fixed start value.

VII. Experimental Results

With the experiments performed on node monitoring protocol for MANETs, we have examined the efficiency of the TTCN-3 test notation for conformance testing of the distributed protocols in general. In this technique we used SDL model for generating MSCs in turn generates test sequences and test cases compatible with TTCN-3, these test cases are executed using TTCN-3 tool with respect to generated test sequence, to confirm the complete behavior of the protocol. The experiment conducted on FSM of the protocol behavior, at the sector of a zone, with few nodes taken into consideration as shown in Fig. 8. Fig. 9 shows SDL model for the protocol.

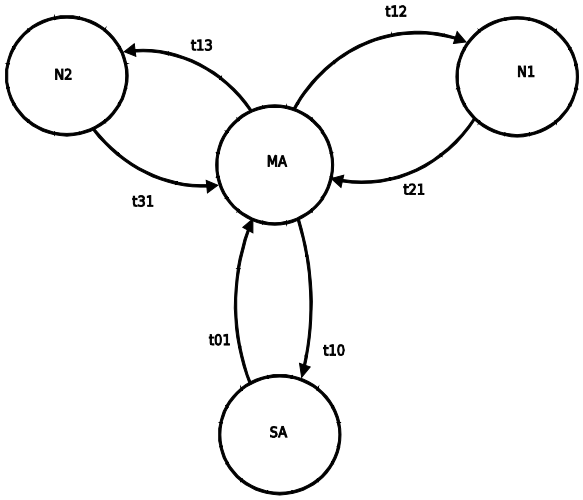

Fig. 8 FSM of protocol at the sector of a zone

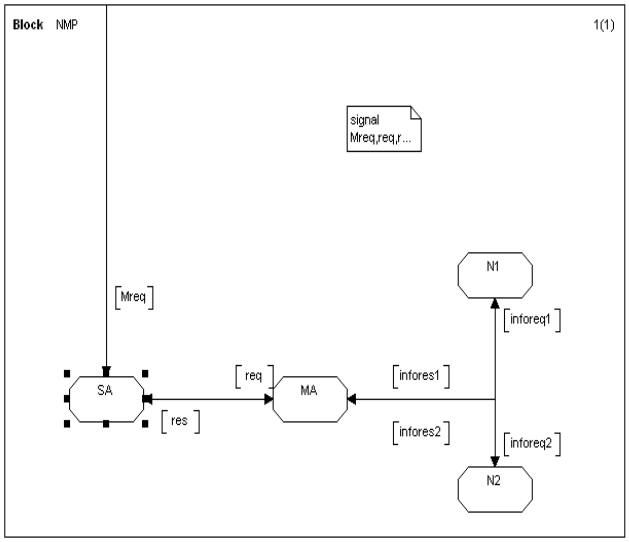

Fig. 9 SDL model of the protocol at the sector of a zone

After the simulation procedure, the respective MSCs are generated as shown in Fig. 10. MSCs are translated into MSC test cases and later into TTCN-3 test cases [19]. The corresponding test sequences are obtained from generated MSCs with the SDL Model. The conformance of the protocol with the given specification is done by executing the test cases with respect to the generated test sequences. Results of conformance test details for the part of protocol monitoring functionality are captured and are as shown in Figs.11-13 and the response of the protocol behavior conformance for varying node density is captured as shown in Fig. 14.

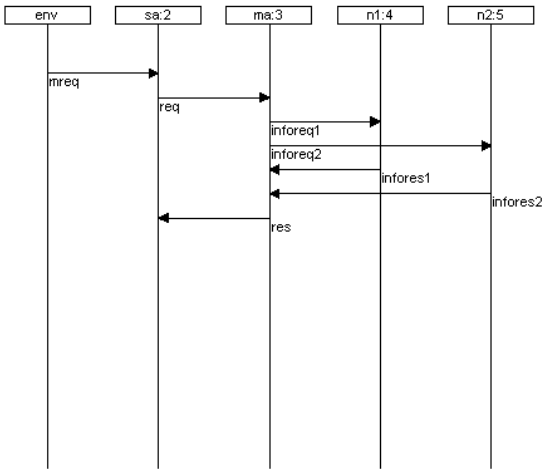

Fig. 10 MSC for the node monitoring protocol

The test sequence derived from the above method for the protocol running at the sector of the MANET is represented as,

$$t_1 = t_{01} = M_{req} / Req,\ t_2 = t_{12} = Req / Info_{req1},\ t_3 = t_{21} = Info_{req}/ Info_{res1},\ t_4 = t_{13} = Req / info_{req2},\ t_5 = t_{31} = Info_{req2}/Info_{res2},\ t_6 = t_{10} = Info_{res2} / Res$$

The generated test sequence of the respective finite state machine of the protocol for the given sector is, $T = t_{01}, t_{12}, t_{21}, t_{13}, t_{31}, t_{10}$.

In TTCN-3, execution of the generated test cases subject to the test sequences leads to conformance of the protocol with the given specifications. For the complete analysis of the protocol for different zones can be accomplished by generating the respective MSCs and combining them as the tree structure to cover all the states, and the entire behavior of the protocol conformance can be checked with their specification.

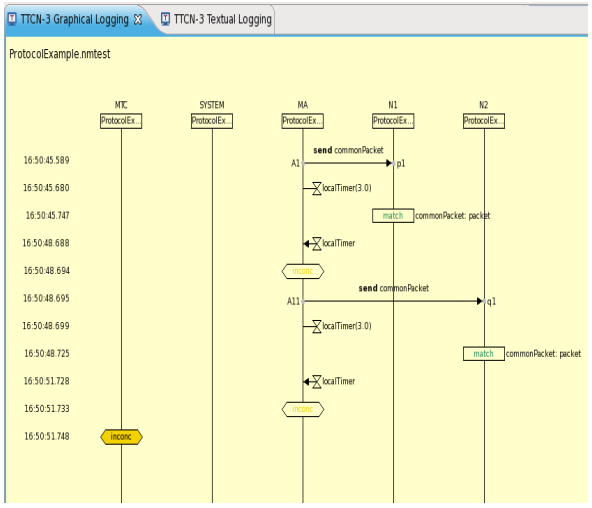

Fig. 11 Protocol behavior conformation for the given sector

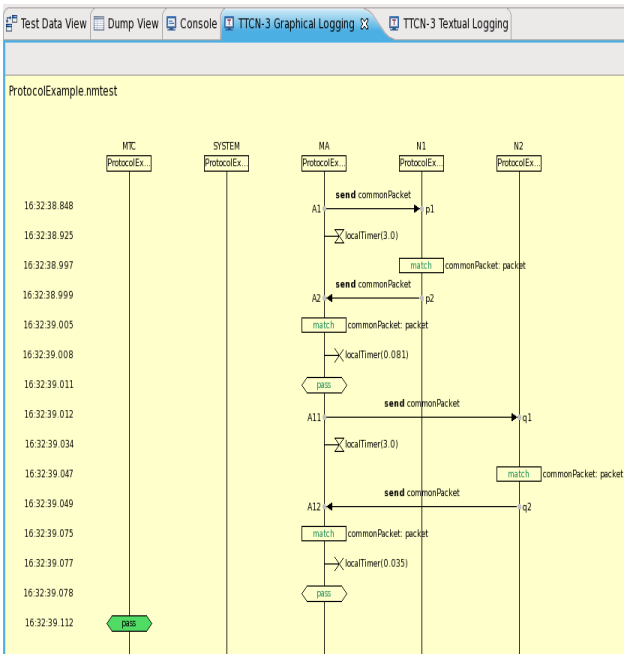

Fig. 12 Fault analysis - Inconclusive state

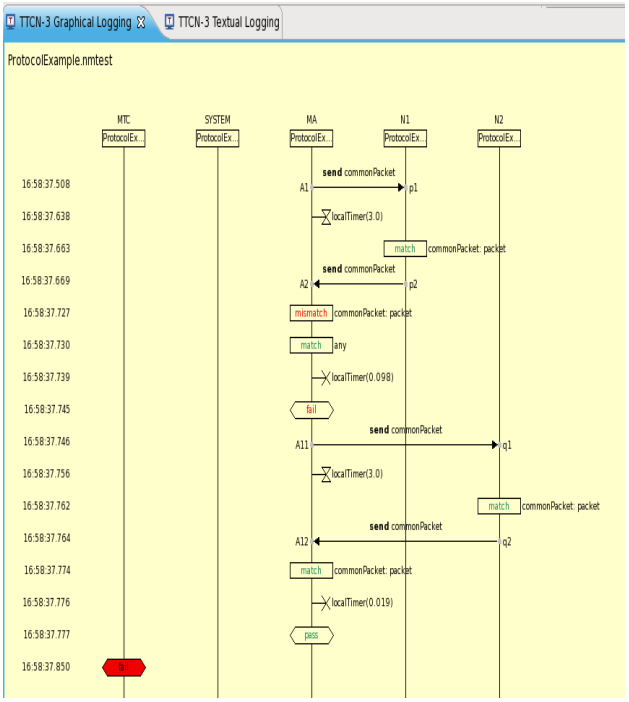

Fig. 13 Fault analysis - Fault Detected

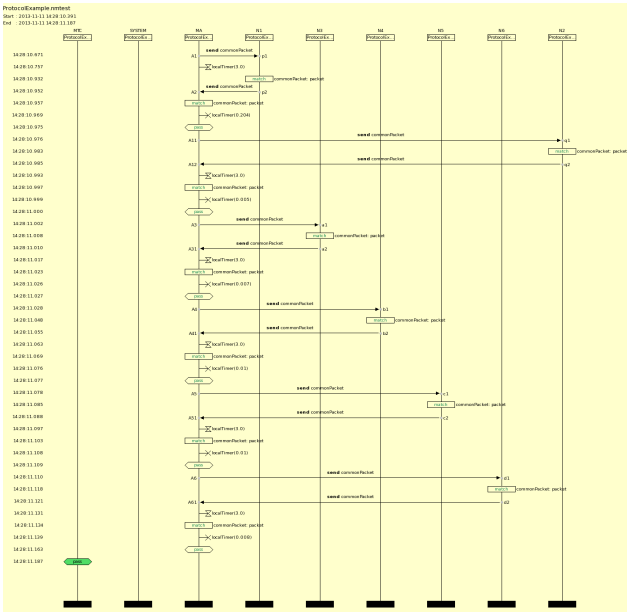

Fig. 14 Fault analysis with varying node density

VIII. Conclusion

In summary a derivation module leads to a test suite from TTCN-3. In our work we considered the TTCN-3 with MSCs from SDL to fulfill the needs of node monitoring protocol conformance testing. In this work it leads to modeling technique that has flexible test environment. The testing of node monitoring protocol conformance with the proposed approach is realized by deriving the test cases from MSCs with SDL (with mapping concept) and executing the test cases with TTCN-3 tool. However, execution of test cases in TTCN-3 for conformance testing with distributed test architecture is done in accordance with the test sequence generated using MSC.

With mapping of MSCs test cases into TTCN-3 test cases, tree structure of MSC is outlined to respective test cases in TTCN-3. Finally, all test cases are combined with specific conditions/rules gives overall test suite in TTCN-3. Conformance of the given protocol can be achieved when subjected to respective test sequence, obtained by MSCs. The results depicted in figures shows the method to be carried out for conformance testing of any distributed protocol with the proposed approach. The future work concerns the automation of test case generation in TTCN-3 and test sequence generation using MSCs through SDL tool.

References

[1] C. Bohoris, G. Pavlou, and H. Cruickshank, "Using mobile agents for network performance management", in Network Operations and Management Symposium, IEEE/IFIP,2000,pp. 637-652.

[2] H. H. To, S. Krishnaswamy, and B. Srinivasan, "Mobile agents for network management: when and when not!,"in Proceedings of the 2005 ACM symposium on Applied computing, SAC'05. New York, NY, USA: ACM,2005, pp. 47-53.

[3] S. Han and Y. Zhang, "Performance evaluation of mobile service design paradigm in ubiquitous computing environments,"in Proceedings of the 2009 Fourth International Conference on Frontier of Computer Science and Technology, ser. FCST '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 89-95.

[4] M. Konaand C. Z. Xu, "A framework for network management using mobile agents,"in Proceedings of the 16th International Parallel all-optical networks, Network and Service and Distributed Processing Symposium, ser. IPDPS'02. Washington, DC, USA, IEEE Computer Society,2002, pp. 7-14.

[5] D. Gavalas, G. E. Tsekouras, and C. Anagnostopoulos, "A mobile agent platform for distributed network and systems management," J. Syst. Softw., vol. 82, no. 2, Feb. 2009, pp. 355-371.

[6] D. Gavalas, D. Greenwood, M. Ghanbari, and M.O'Mahony, "Using mobile agents for distributed network performance management,"in 3rd International Workshop on Intelligent Agents for Telecommunication Applications, 1999.

[7] J. Grabowski, B. Koch, M. Schmitt, and D. Hogrefe. "SDL and MSC Based Test Generation for Distributed Test Architectures",in SDL Forum SDL'99, June 1999.

[8] Collin willcock, Thomas Deib."*An Introduction to TTCN-3*", Publisher, John wiley and sons ltd, 2005.

[9] ChannappagoudarMallikarjun B., VenkataramPallapa, "Mobile agent based node monitoring protocol for MANETs", National Conference on Communications NCC, vol.no.1, pp.5, February 2013.

[10] A. Kerbrat, T J´ron, and R Groz. "Automated test generation from sdl particulars". In RachidaDssouli, Gregor von Bochmann, and YairLahav, editors, SDL-99 The Next Millennium, 9th International SDL Forum, pp. 135-152, June 1999.

[11] B. Koch, J. Grabowski."Autolink - A Tool for Automatic Test Generation from SDL Specifications". In IEEE International Workshop on Industrial Strength Formal Specification Techniques, October 1998.

[12] R. L. Probert, H. Ural, and A. W. Williams. "Rapid generation of functional tests using mscs, sdl and ttcn", Computer Communications, 24(3-4),2001, pp-374–393.

[13] M. MatashBorujerdi, S.M. Mirzababaei. "Formal Verification of a Multicast Protocol in Mobile Networks". World Academy of Science, Engineering and Technology, Vol:11, 2007-11-27,2007.

[14] ITU-T. "Specification and Description Language (SDL)", Recommendation Z.100, ITU-T, August 2002.

[15] ITU-T. "Message Sequence Chart (MSC)", Recommendation Z.120, ITU-T, November 1999.

[16] ETSI. "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3"; Part 1: TTCN-3 Core Language. ETSI Standard 201 873-1 v2.2.1, ETSI February 2003.

[17] Dai, Z., Grabowski J., Neukirchen, H "TIMED TTCN-3, A Real-Time Extension for TTCN – 3". In Schieferdecker, I., König, H., Wolisz, A., eds.: Proceedings of the I F I P TC6/WG6.114th International Conference on Testing of Communicating Systems, (TestCom 2002),Berlin, Germany, The International Federation for Information Processing, IFIP, Kluwer Academic Publishers, 407-424, 2002.

[18] Rosziati Ibrahim, Noraini Ibrahim. "A Tool for Checking Conformance of UML Specification". World Academy of Science, Engineering and Technology, Vol: 27, 2009-03-25, 2009.

[19] M. Ebner., "TTCN-3 Test Case Generation from Message Sequence Charts," in Workshop on Integrated-reliability with Telecommunications and UML Language (ISSRE04:WITUL), France, November 2004.

[20] FrancineNganiNoudem and cesarViho, "Modeling, Verifying and Testing Mobility Protocol from SDL Language", LNCS 3530, pp. 198-209, 2005.

[21] A. D. Friedman and P. R. Menon, "Fault Detection in Digital Circuits", Prentice-Hall 1971.

[22] Z. Kohavi, "Switching and Finite Automata Theory", 2nd Ed. McGraw-Hill, 1978.

[23] A.V. Aho,J. E. Hopcroft,and J. D. Ullman, "The Design and Analysis of Computer Algorithms",Addison-Wesley,1974.

[24] G. J. Holzmann, "Design and Validation of Protocols", Prentice-Hall, 1990.

[25] Lee, D. Yannakakis,Mihalis,"Principles and methods of testing finite state machines-a survey," Proceedings of the IEEE, vol.84, no.8, pp.1090, 1123, Aug 1996.

[26] F. C. Hennie, "Fault detecting experiments for sequentialcircuits", Proc.5th Ann. Symp. SwitchingCircuit Theory and Logical Design, pp. 95-110, 1964.

[27] Z. Kohavi, "Switching and Finite Automata Theory", 2nd Ed. McGraw-Hill, 1978.

Mallikarjun B. Channappagoudarreceived his bachelor's degree in electronics and communication engineering from Gulbarga University and master's degree from Bangalore University. He is a research scholar in PET unit, Department of Electrical Communication Engineering in Indian Institute of Science, Bangalore, India, from 2011.

His research interestincludes protocol engineering, network management in mobile adhoc networks,application of agents inmobile adhoc network management, and protocol testing.

PallapaVenkataramreceived his Ph.D degree in Information Sciences from the University of Sheffield, U.K.in 1986. He is currently a Professor of Electrical Communication Engineering with Indian Institute of Science, Bangalore, India.

Prof. Pallapas research interests includes protocol engineering, wireless networks, network management, computational intelligence applications in communication, mobile computing security and multimedia systems, protocol testing. He is a Fellow of IEE (England), Fellow of IETE(India) and a Senior member of IEEE Computer Society. Dr. Pallapa is the holder of a distinguished visitor diploma from the Orrego University, Trujillo, Peru. He has published over 200 papers in International/national Journals/conferences.