Mehran University Research Journal of Engineering & Technology
Vol. 38, No. 2, 321-334 April 2019
p-ISSN: 0254-7821, e-ISSN: 2413-7219
DOI: 10.22581/muet1982.1902.07

Diagnostic Agent Based Inter-Process Communication Aware Monitoring System for Wireless Sensor Networks

AMNA ZAFAR*, AND ALI HAMMAD AKBAR*

RECEIVED ON 24.01.2018 ACCEPTED ON 25.05.2018

ABSTRACT

Process failures are instigated by underlying errors and faults in various layers of WSN (Wireless Sensor Network) communication protocol stack. Therefore, efficient and effective monitoring systems for fault detection and diagnosis are imperative for fault tolerance and robust operation of WSN to meet critical application requirements for reliability and throughput. Existing detection-diagnosis regimen are either centralized or distributed and network monitoring is performed passively or actively. This work presents a diagnostic agent based inter-process communication aware monitoring system for WSNs. Diagnostic agent actively performs probe-based process execution tracking and examines the effects of errors, omissions and channel misbehavior on process execution at node, link and network levels to implement failure detection and fault diagnosis. Such diagnosis is performed through the inference of inter-process communication of stacked and peer layer processes on sender and receiver side. The monitoring system has been implemented in Castalia simulator for WSN. Local diagnostic agent is implemented on sensor nodes for self-monitoring and network wide fault diagnosis is performed by global diagnostic agent on cluster head. Simulation results show that the system performs robust root cause analysis of critical process failures due to errors in stacked and peer layer processes. The decentralized distribution of diagnostic load on sensor nodes and cluster head produces lesser communication overhead and is energy efficient.

Key Words: Agent, Process, Protocol, Fault Diagnosis, Wireless Sensor Networks.

1. INTRODUCTION

WSN has emerged as a new paradigm for pervasive computing and collaboration with advances in embedded systems and wireless technology. There are many distinct features of WSN which differentiate them from traditional wired network environment. The stringent resource constraints of sensor nodes in terms of limited memory, computation capability and energy pose a challenge for researchers to develop protocols and techniques for robust and reliable operation. Autonomous deployment of WSN in unattended and hostile environment results in high frequency of failures due to underlying errors and faults [1]. Fault is an erroneous state of a hardware or software component. Such faults manifest as errors. An error characterizes an incorrect system state that may lead to failure causing aberrations from normal system behavior

Authors E-Mail: (amnazafar@uet.edu.pk, ahakbar@gmail.com)
* Department of Computer Science & Engineering, University of Engineering & Technology, Lahore, Pakistan.

i.e., service interruption. The software layer of sensor nodes comprises of processes executing on the communication protocol stack. These processes communicate to exchange information and therefore, need to be checked for possible failures. Communication errors such as connectivity loss, routing loops and broadcast storms cause network partitioning, reduced throughput and network congestion etc. Additionally, wireless channel errors and radio interference are a source of fading, collisions and packet loss. The QoS (Quality of Service) requirements of critical WSN applications demand a realistic, timely and sufficient visualization of underlying network conditions. It is therefore imperative to design network monitoring systems that perform efficient fault detection and diagnosis.

Generally, WSN monitoring is carried out either actively or passively [2]. In active monitoring, debugging agent on each sensor node periodically collects node status updates and transmits to sink for fault detection and diagnosis [3]. The active monitoring offers detailed information about the whole network whereas increasing communication overhead and energy consumption. Passive monitoring schemes employs packet sniffing to infer node and network status. However, packet sniffing requires specialized hardware increasing monitoring cost. This work proposes a diagnostic agent that actively performs node self-diagnosis and triggers diagnostic communication with coordinator node i.e. CH (Cluster Head) need basis only. The diagnostic work load increases or on nude basis only decreases on sensor nodes and CH based on network dynamics enabling flexible monitoring.

Protocols oversee, regulate the operational specifications and guiding principles, and provide assurance for planned usage of communication networks including WSN [4]. An investigation of dissemination effect of one process failure on stacked or peer layer processes execution can give useful insight for network monitoring. However, most of the existing network monitoring schemes do not handle the systemic impact of communication errors on protocol execution at node, link and network levels. Moreover, the impact of vertical and horizontal propagation of a process failure on interrelated processes has not been examined in detail. This work presents a diagnostic agent based inter-process communication aware monitoring system for WSNs that actively performs probe based analysis of process execution. The rest of the paper is organized as such. In section 2, we review existing network monitoring schemes for fault diagnosis in WSNs. Section 3 presents the proposed system architecture and working in detail. In section 4, the simulation details and results obtained in Castalia simulator [5] are discussed. Section 5 concludes with discussion on research findings.

2. RELATED WORK

A monitoring system for network diagnostics is important for fault tolerant and robust operation of WSNs. Several network health monitoring systems for WSNs have been proposed. Ramanathan et. al. [6] proposed Sympathy, a debugger for active network monitoring. Agent code on each node periodically transmits node statistical metrices such as routing tables and neighbor lists, etc. to sink. The sink analyzes metrics and executes decision tree based tests to identify potential failures. Failure localization is performed by assigning a specific source: self, path or sink. Sympathy achieves high detection accuracy but diagnosis traffic puts added burden on already limited resources of sensor nodes. Moreover, Sympathy lacks the ability to analyse protocol failures.

Liu et. al. [7] presented PAD (Passive Diagnosis Approach) an active monitoring system that employs packet marking scheme at each node. Sink parses marked packets to regenerate network topology and produces preliminary diagnosis report about link failures and packet loss. PAD employs probabilistic inference engine for failure diagnosis using observed positive and negative network symptoms. However, PAD depends upon message transmission to send information increasing detection latency. Moreover, no monitored information to indicate possible abnormal operation is sent by nodes that do not transmit sensed data.

DID (Directional Diagnosis) implements active node level tracing and incremental probing. For fault diagnosis, an interence engine is built at sink [8]. The inference engine reconstructs topical topology involving problematic region using network symptoms detected through node tracing. Afterwards, sink broadcasts incremental probes into the network increasing diagnosis communication overhead. Moreover, the inference engine creates a data vertex for each sensing node. However, the exponential linear increase in computational complexity with addition of data vertex for each sensing node makes DID inapt in case of large data centric network.

For passive monitoring, SNIF (Sensor Network Inspection Framework) is presented in [9]. To overhear network traffic, multiple sniffing points are used forming a temporary DSN (Deployment Support Network). The receiver part of WSN protocol stack i.e. MAC (Medium Access Control) and PHY (Physical) layers is implemented at each sniffer node in the DSN. Each DSN node transmits the captured data streams to a central host station acting as a SNIF sink. The data stream processor analyzes the data streams to infer and report network problems involving individual nodes, wireless links, routing path and network partitions. However, the scheme is not scalable due to extra maintenance cost for two networks.

A network monitoring and protocol analyzer system Z-Monitor for IEEE 802.15.4 compliant WSNs is implemented in [10]. Z-Monitor employs a single sniffer node to capture network traffic passively. The network behavior is evaluated by statistical analysis of captured traffic at base station. Z-Monitor stores packets in a buffer, performs decoding, parsing and displays network statistics along with parsed frames. However, the system only provides limited protocol analysis and lacks the ability to diagnose network failures. An extended version of Z-Monitor [11] supports multiple sniffers forming a secondary network for network diagnostics at base station. The extended Z-Monitor is capable of identifying the potential causes of connectivity issues. However, due to additional cost to deploy and maintain monitoring network, this scheme is not widely adapted.

Rodenas-Herráiz et. al. [12], proposed an on-site diagnostic system for IPv6 over low power wireless personal area networks (6LowPANs) presented. The system comprised of a traffic monitor to sniff 802.15.4 MAC frames, a frame decoder and a decision tree. To detect network problems, the frame decoder excerpts and examines network and MAC layer headers. Subsequently, a decision tree is activated to perform root cause analysis of network problems which may include unresponsive node, network partition and intermittent dis-connectivity. However, the system is unable to diagnose network partitions due to problem along routing path. Moreover, the decision tree depends on examination of communication protocol control packets for correct diagnosis but lacks functionality to handle protocol failures.

Pimito, a passive monitoring system for WSNs is presented in [13]. The system implements a hierarchical structure consisting of three components: monitoring node, gateway for direct communication with the monitoring node and a dedicated server. The gateway transmits collected monitored data to the server for analysis. However, the system only provides limited topology visualization. Nevertheless, passive monitoring systems capture transmitted frames "on the air" unable to obtain direct information from nodes. An agent based fault diagnosis scheme for WSNs is proposed in [14]. The scheme presents an agent architecture that combines active and passive monitoring. Fault diagnosis agent is deployed at coordinator node i.e. sink that employs a causal model to map network symptoms with fault root causes. However, the system implementation is limited to symptoms detection only.

Existing active and passive monitoring systems generally detect failures by examining WSN communication protocol traffic at MAC and PHY layers. However, inter-process communication of network, MAC and PHY layer protocols needs to be explored further to investigate the impact of faults and errors on protocol failures [15]. To the best of authors' knowledge, existing network monitoring schemes do not establish relationship between fault diagnosis and inter-process communication of functional peers. In [16], a hybrid fault diagnosis architecture based upon intra-process communication and inter-process dependencies is proposed by the authors. This work extends and presents a diagnostic agent based inter-process communication aware monitoring system for WSNs.

3. MONITORING SYSTEM

The monitoring system performs probe based investigation of anomalous behavior of WSN communication protocol stack processes. The system identifies typical processes that run on the protocol stack. For practical considerations, AODV (Adhoc On Demand Distance Vector Routing) [17] and IEEE 802.15.4 MAC and PHY layer processes [18] have been selected. The process flows of these protocols serve as a foundation for the system architecture.

3.1 System Architecture

The system defines LDA (Local Diagnostic Agent) for node self-diagnosis as shown in Fig. 1. GDA (Global Diagnostic Agent) performs fault diagnosis within a cluster. To investigate process execution on network, MAC and PHY layers, a periodic probing scheme has been designed. LDA examines the results of probing to detect process failure and classify error level. Afterwards, fault diagnosis is performed by executing Priority Tests that are based upon inter-process communication of both stacked and peer layer processes at node and network levels. The priority tests are incrementally executed on sensor nodes and CH that is a specific entity in the network. Partial and deployment-specific realization of the proposed monitoring system can be achieved on sensor nodes and CH through LDA and GDA respectively.

3.2 Local Diagnostic Agent

LDA contains Failure Detector and Priority Test modules, as shown in Fig. 1. The procedures executing on peer layers of protocol stack form software components of a networked system. The process execution status depends on status of the underlying procedures. The status of each procedure with in a process is stored as a marker. The procedure execution status is modeled as a binary i.e. normal and error. Under normal circumstances, procedures with in a process address space run without any error and return a normal marker. Exception handling code in protocol implementation is used to generate warnings and alerts, that activates error markers creation on multiple layers. Process execution order and markers are stored dynamically in the form of PESS (Process Execution Status Stack).

3.2.1 Failure Detector

LDA periodically sends marker probe to collect markers by traversing corresponding PESS as shown in Fig. 2. The failure detector module parses probe results to decode error markers that may be representing process failures. It accumulates error count and generates PECs (Procedure Error Counters) for each process. Failure detection is performed by top down comparison of error counters against thresholds. Error classification is done based upon spatio-temporal impact of errors on process execution and node functionality. LDA defines three error levels: critical, warning and alert. It is important to mention here that an error level may indicate either a process failure or an alert for an unavoidable one. Critical errors are major cause of process failures that interrupt communication and may result in sensor node disconnectivity from rest of the network. An instance of such failure could be the disassociation of a sensor node with the coordinator. The errors which may have a transient effect on infrequent occurrences might as well have a completely different effect such as process failure if these continue such as buffer overflow. Such errors are assigned warning level. Alert level is assigned for those errors that cause temporary delays in communication such as radio not in receiving (RX) mode for CSMA/CA (Carrier Sense Multiple Access with Collission Avoidance) process. After failure detection and classification, LDA executes the Priority Tests to perform inter-process communication aware root cause analysis.

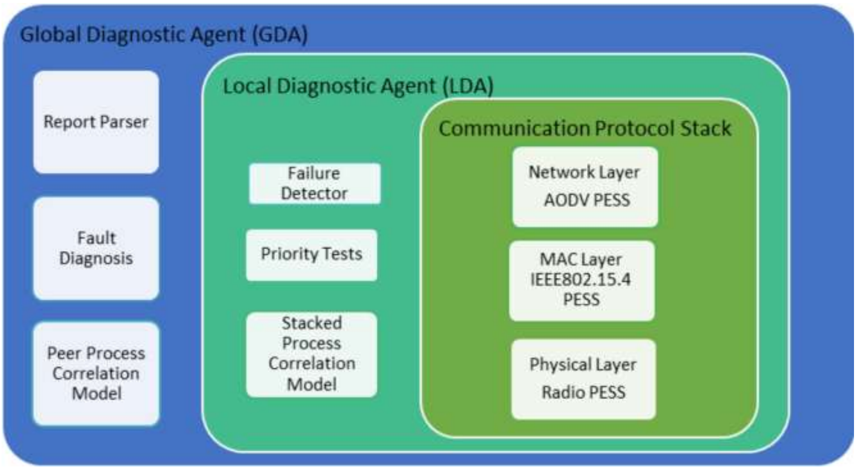
Global Diagnostic Agent (GDA)
Report Parser
Fault Diagnosis
Peer Process Correlation Model
Local Diagnostic Agent (LDA)
Failure Detector
Priority Tests
Stacked Process Correlation Model
Communication Protocol Stack
Network Layer AODV PESS
MAC Layer IEEE802.15.4 PESS
Physical Layer Radio PESS

FIG. 1. MONITORING SYSTEM ARCHITECTURE CONTAINING LOCAL DIAGNOSTIC AGENT DEPLOYED ON SENSOR NODES AND GLOBAL DIAGNOSTIC AGENT ON CLUSTER HEAD

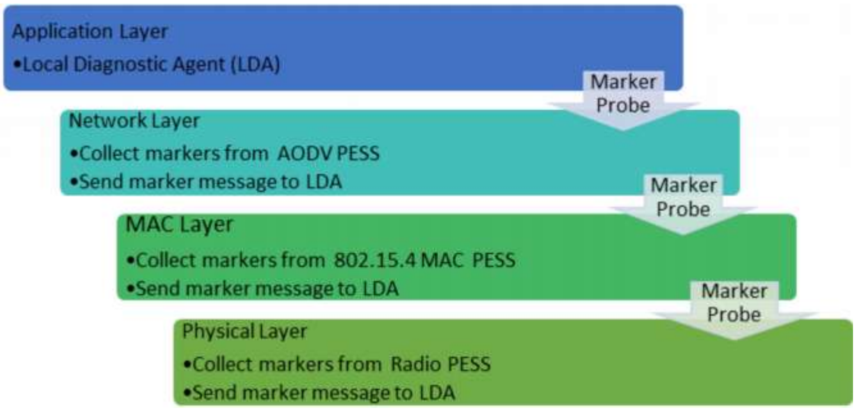
Application Layer
•Local Diagnostic Agent (LDA)
Marker Probe
Network Layer
•Collect markers from AODV PESS
•Send marker message to LDA
Marker Probe
MAC Layer
•Collect markers from 802.15.4 MAC PESS
•Send marker message to LDA
Marker Probe
Physical Layer
•Collect markers from Radio PESS
•Send marker message to LDA

FIG. 2. LDA PERIODICALLY SENDS MARKER PROBE AFTER PROBE INTERVAL TO PROTOCOL STACK LAYERS

3.2.2 Stacked Process Correlation Model

Inter-process correlations for stacked processes are based upon inter-process communication in the form of up/down calls as shown in Fig. 3. These are dynamically represented as a model. The model components are stacked processes on each node, as shown in Fig. 4. The directional links represent dependencies of these processes. According to down call order, discover route process on network layer communicates with MAC layer transmit frame process. Subsequently, transmit frame execution triggers a chain of inter-process communication involving MAC associate, synchronize, CSMA/CA and PHY layer transmit signal processes in order. The failure of any of these interrelated processes may result in transmit frame failure. According to stacked process correlation model, for a down call the failure effect may propagate upward the protocol stack and vice versa for an upcall. Similarly, the peer layer routing processes i.e. RREQ (Route Request) processing. PREP (Route Response) processing and RREP generation may also fail due to errors in correlated stacked processes. Therefore, on peer layers of protocol stack, the failure effect may propagate horizontally in the opposite direction of the to/from calls.

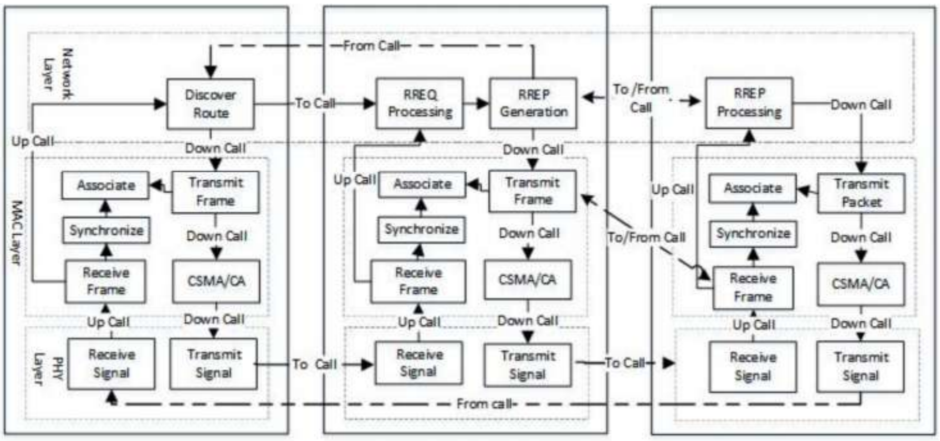

FIG. 3. INTER-PROCESS COMMUNICATION IN THE FORM OF UP/DOWN CALLS FOR NODE LEVEL STACKED PROCESSES AND TO/ FROM CALLS FOR PEER LAYER PROCESSES ON SENDER AND RECEIVER SIDES

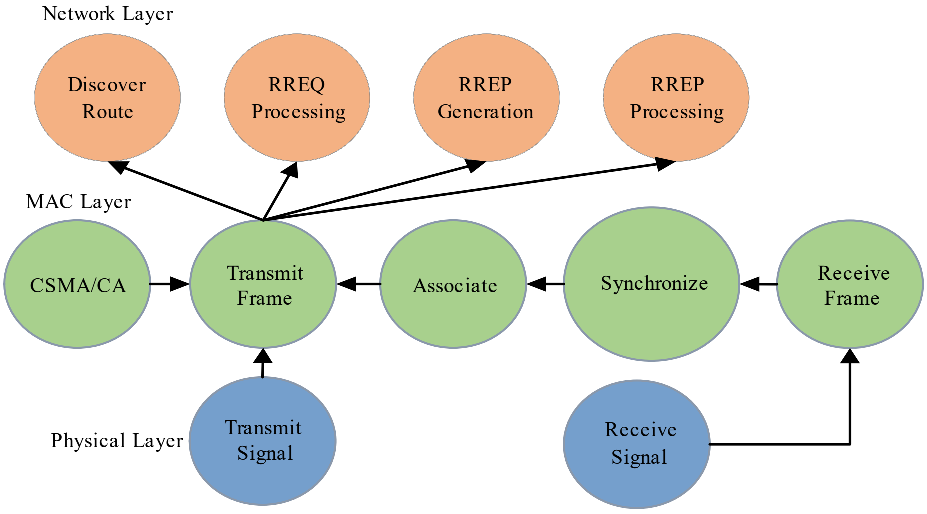

FIG. 4. STACKED PROCESS CORRELATION MODEL: DIRECTIONAL LINKS REPRESENT DEPENDENCY OF UPPER LAYER PROCESSES ON LOWER LAYER AND/OR SAME LAYER PROCESSES

3.2.3 Priority Tests

LDA employs Priority Tests (Algorithm-1) for inter-process communication aware fault diagnosis. The priority tests are executed to identify underlying critical errors, warnings and alerts that may be potential root causes of a stacked or peer process failure. Priority Tests are executed in parts on multiple sensor nodes according to process execution sequence in a particular probe interval. Each priority test compares the error counters of interrelated processes (critical, warnings and alerts) against the corresponding downcall/upcall counters. For example, in case of discover route failure, the PECs of synchronize, associate, transmit frame, CSMA/CA and transmit signal processes are sequentially compared with number of times RREQ packet was sent from network to MAC layer of the node i.e. down call counter. If these process error counters are greater than discover route down call counter, underlying errors are added to the list of probable root causes. For peer routing processes, similar priority tests are executed.

ALGORITHM-1. PRIORITY TESTS EXECUTED BY LOCAL DIAGNOSTIC AGENT

```
Input:  PECs, error levels, process down call/up call counters
Output: fault report
for each critical routing process failure do
    if error level = critical and MAC synchronize failure then
        if error level = warning and MAC receive frame error then
            if PEC [ beacon loss] > synchronize up call counter then
                insert synchronize failure due to beacon loss error in fault report
            end if
        if error level = warning and PHY receive signal error then
            if PEC [ low power signal] > receive signal up call counter then
                insert synchronize failure due to low power signal error in fault report
            if PEC [interference] > receive signal up call counter then
                insert synchronize failure due to interference error in fault report
            if PEC [bit errors] > receive signal up call counter then
                insert synchronize failure due to channel errors in fault report
            end if
        if error level = alert and receive signal error then
            if PEC [RX state] > receive signal up call counter then
                insert Synchronize failure due to radio not in RX state error in fault report
            end if
        end if
    if error level = critical and MAC associate failure then
        if PEC [association denied] > routing process down call counter then
            insert routing failure due to no association error in fault report
        end if
    if error level = warning and MAC transmit frame error then
        if PEC [buffer overflow] > routing process down call counter then
            insert routing failure due to MAC buffer overflow error in fault report
        end if
    if error level = warning and CSMA/CA error then
        if PEC [maximum back-off] > routing process down call counter then
            insert routing failure due to channel collisions error in fault report
        end if
    if error level = alert and CSMA/CA error then
        if PEC [ RX state error] > routing process down call counter then
            insert routing failure due to radio not in RX state error in fault report
        end if
    if error level = warning and PHY transmit signal error then
        if PEC [buffer overflow] > routing process down call counter then
            insert routing failure due to radio buffer overflow error in fault report
        end if
end for
if critical failure diagnosed then
    return
else
    transmit fault report to CH
    return
```

If critical process failure is successfully diagnosed locally, a fault report containing primary root causes is generated. Otherwise, the failure cause may be external. The external causes are un-observable on this node; accordingly, the situation is considered as a peer layer process failure. In this scenario, partial diagnosis results are stored in the fault report and sent to CH for in-depth investigation. For each peer routing process failure, similar reports are generated and transmitted to CH.

3.3 Global Diagnostic Agent

The global diagnostic agent on CH contains Report |Parser and Fault Diagnosis modules. The Report Parser module collects incoming fault reports after each probe interval. The Fault Diagnosis module performs root cause analysis of critical process failures according to peer process correlation model.

3.3.1 Peer Process Correlation Model

The inter-process communication of peer layer processes is represented in the form of to/from calls. Therefore, inter-process correlations are extracted from these calls and a peer process correlation model is proposed as shown in Fig. 5. The peer layer routing processes may fail due to failure of interrelated peer processes or internal procedural errors. Accordingly, the failure effect may propagate horizontally in the opposite direction of the to/from calls on peer layers of protocol stack.

3.3.2 Fault Diagnosis

The fault diagnosis module executes priority tests to investigate impact of peer layer process errors on critical failures according to the peer process correlation model, as shown in Algorithm-2.

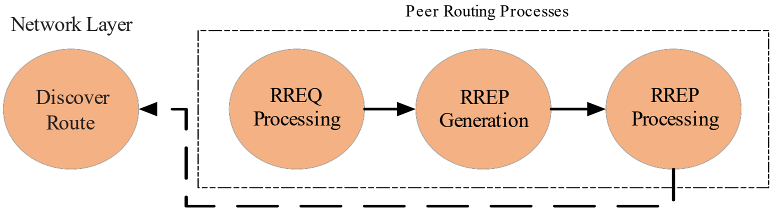

FIG. 5. PEER PROCESS CORRELATION MODEL BASED ON INTER-PROCESS COMMUNICATION AS TO/FROM CALL

ALGORITHM-2. PRIORITY TESTS EXECUTED BY GLOBAL DIAGNOSTIC AGENT ON CH

```
Input: fault reports, fault report counter
Output: diagnosis report
set $i = 0$ ;
repeat
    set $j = i + 1$;
    parse ith fault report
    if discover route critical failure then
        parse jth fault report
        if RREQ Processing failure then
            if error level = alert and source RREQ blacklist error then
                route failure due to uni-directional link fault
            end if
        if RREP Generation failure then
            if error level = warning and no reverse route error then
                route failure due to corrupt routing table fault
            end if
        if RREP Processing failure then
            if error level = warning and no reverse route error then
                route failure due to corrupt routing table fault
            end if
        end if
until i == fault report counter
if fault diagnosed on CH then
    generate diagnosis report
return
```

4. EVALUATION

The system has been developed in Castalia-3.3 simulator for WSN that is based upon OMNET++ ver.4.6 on ubuntu 14.06. A default implementation of WSN protocols is provided in Castalia. A realistic channel module deals with communication between peer layer processes on sender and receiver side. Similarly, a temporal channel and path loss model is used to simulate the effect of channel impairment on node behavior. To store routing, MAC and PHY layer markers in stack data structure, the communication module in Castalia has been altered. The LDA on application layer sends marker probe as a control command to the communication module. The system defines a fault report packet format as an extension of the application packet in Castalia. Performance evaluation is based upon extensive simulation experiments. The sensor nodes are randomly deployed in a two-dimensional grid along with CH and sink. Sensor nodes periodically sends data packets to the sink that triggers stacked and peer processes execution. Table 1 summarizes simulation parameters.

4.1 Performance Evaluation of LDA

The probe interval has been varied for performance evaluation of LDA. The outputs of failure detector and priority tests modules are examined. The impact of stacked and peer layer processes on critical failure is investigated to evaluate inter-process communication aware fault diagnosis.

4.1.1 Impact of Probe Interval

The effect of periodic probing on failure detection and error classification is analyzed. The error classification implementation is based on analysis of long term behavior of WSN under typical and implied faults. To explain effect of probe interval variation, it is important to understand relationship between anomalies classified, either as critical errors, warnings or alerts. The communication protocols implementation and exception handling code for error reporting defines the relationship. The adaptability of the exception handling code implemented through timers and counters is demonstrated through spatial frequency of alerts and warnings with respect to the frequency of critical errors. The short-term manifestations of alerts and warnings through critical errors is either repressed or expanded with increase and decrease in probe interval. The effects are suppressed in case of longer probe interval of 540 secs as shown in Fig. 6.

TABLE 1. SIMULATION PARAMETERS

Simulation Time	3600 seconds (secs)
Network Area mxm	60x60
Network Size	20
Probe Interval	180, 360, 540 (sec)
Packet Rate	1 packet /10 sec, 1 packet /30 sec, 1 packet /60 sec
Routing	AODV
MAC	IEEE 802.15.4
PHY Model	CC2420
Radio Frequency Output Power	-3 dbm
Receiver Sensitivity	-95.0 dbm
Interference Model	Additive interference
Channel Model	Log-normal shadowing
Path Loss Exponent	2.4

The stacked process correlation model based fault diagnosis has been investigated via output of the priority tests. The effect of critical errors, warnings and alerts in stacked processes on routing failure is examined. For probe interval of 360 secs, a higher number of MAC critical errors are classified as potential root cause of the overall routing failure as shown in Fig. 7. It is due to cumulative effect of PHY layer warnings and alerts on MAC critical errors. The Priority Tests infers a comparatively smaller number of MAC critical errors as potential root cause for shorter probe interval of 180 secs. It is due to the fact that network remains stable over a short period of time.

The impact of errors in stacked processes on peer routing process failure has been analyzed. For shorter probe interval of 180 secs, the Priority Tests classifies a large number of MAC and PHY warnings as potential root causes of peer routing process failure as shown in Fig. 8. However, for longer probe interval of 540 secs, the frequency of MAC critical errors is reduced due to network stability. Consequently, root cause analysis done by Priority Tests classifies smaller number of warnings and alerts.

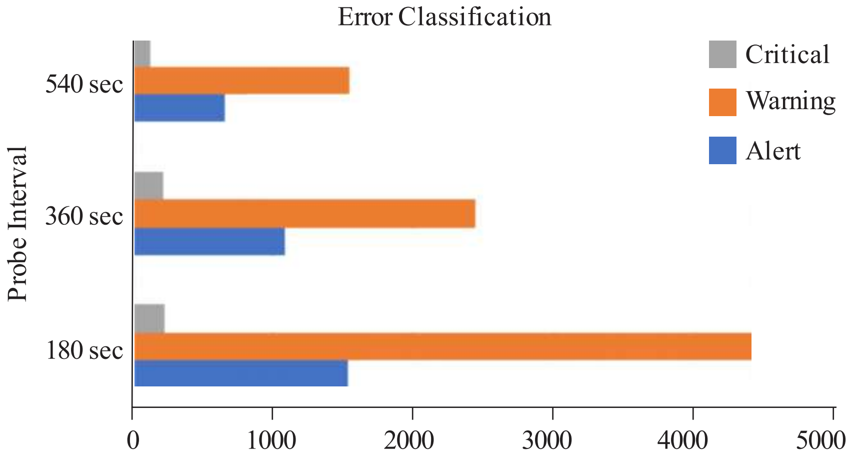

FIG. 6. NETWORK WIDE CRITICAL-ERROR-WARNING-ALERT CLASSIFICATION SHOWS HIGHER FREQUENCY OF WARNINGS FOR SHORTER PROBE PERIOD OF 180 SECS

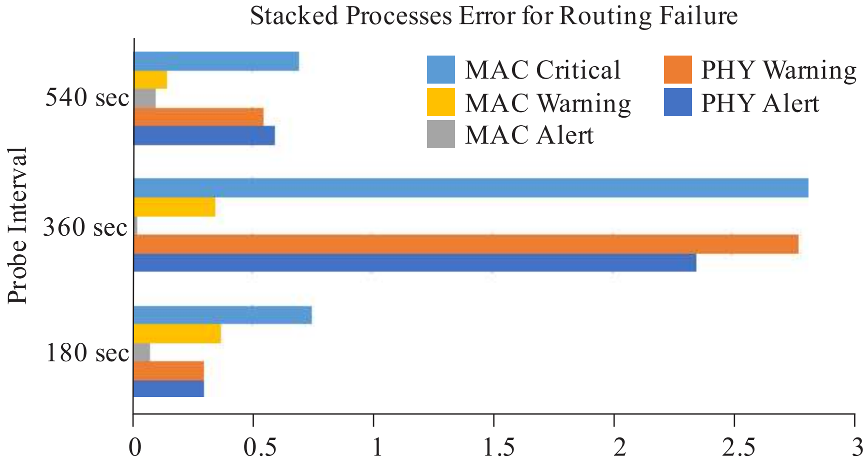

FIG. 7. ERRORS IN STACKED PROCESSES CLASSIFIED AS POTENTIAL ROOT CAUSES FOR ROUTING FAILURE

Fig. 9 shows the underlying errors that are diagnosed as root cause of routing failure. The Priority Tests infers synchronize process failure i.e., SYN loss as dominant root cause of the overall routing failure. The Synchronize failure is caused by beacon packet loss due to PHY layer errors such as interference and radio not in RX state. The spatio-temporal frequency of critical errors is changed with length of probe interval as discussed previously. Similar results are obtained in case of peer routing process failure diagnosis as shown in Fig. 10.

4.2 Performance Evaluation of GDA

To analyze the impact of errors in peer routing processes on routing failure, the output of Priority Tests executed by GDA is analyzed. For this purpose, exceptions have been introduced in Castalia implementation of AODV routing protocol along with exception handling code. In this case, ideal radio communication model is considered. Peer routing process failure is caused by procedural errors or errors in correlated stacked processes. The failure effect is propoagted on peer layers of protocol stack causing routing failure. Subsequently, LDA infers

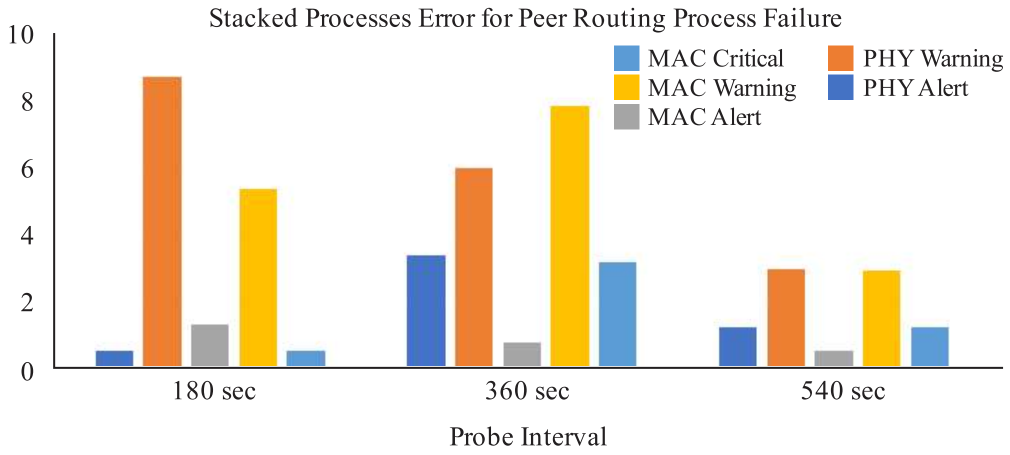

FIG. 8. ERRORS IN STACKED PROCESSES CLASSIFIED AS POTENTIAL ROOT CAUSE OF PEER ROUTING PROCESS FAILURE FOR VARYING PROBE INTERVALS

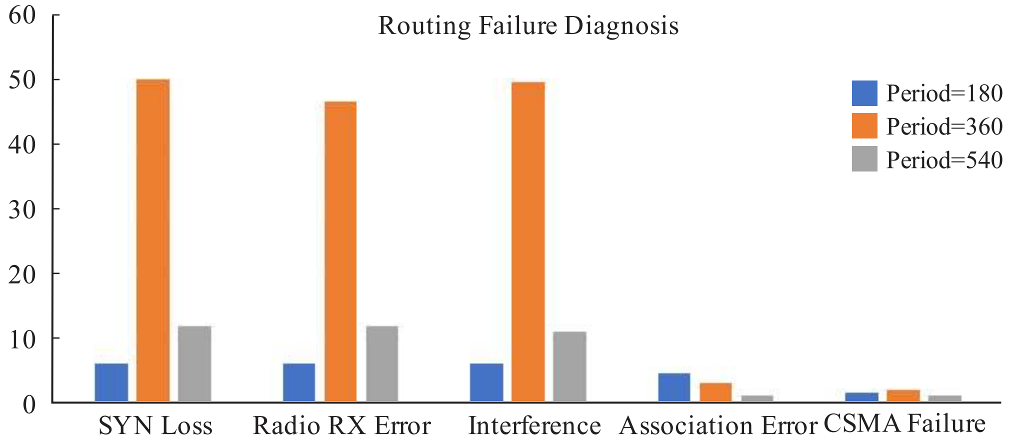

FIG. 9. ROUTING FAILURE DIAGNOSIS SHOWS DOMINANT IMPACT OF SYN LOSS (CRITICAL ERROR) FOR LONGER PROBE PERIOD INTERVAL OF 360 SEC

external failure source and sends fault report to GDA on CH. The GDA infers errors in RREP processing and RREP generation as major source of routing failure. These errors are caused by corrupt routing table entries for reverse route to the originator node initiating discover route process. Due to ideal radio communication model, fewer links outside the disk range are unidirectional. Therefore, on unidirectional links RREQ processing fails due to black listed RREQ source error. Subsequently, RREQ processing error is also inferred as potential root cause of routing failure (Fig. 11).

4.3 Diagnosis Communication Overhead

To evaluate diagnosis communication overhead, ECDF (Empirical Cumulative Distribution Function) are computed with varying values of probe interval and packet rate. In Fig. 12 , ratio of the fault report bytes sent to overall transmitted bytes is represented on x-axis and the corresponding ECDF is on y-axis. Lesser overhead is represented by smaller ratios. However, for 360 sec probe interval, the communication overhead is comparatively higher due to increase in detection latency (time between error marker generation and reporting through probing).

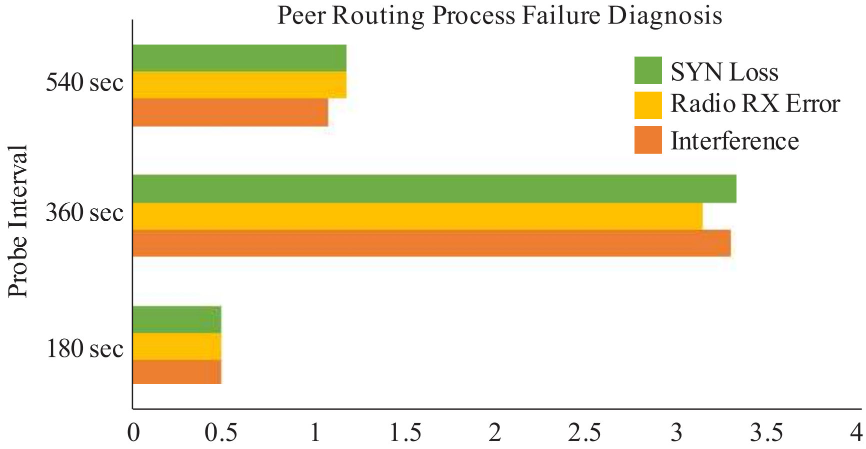

FIG. 10. PEER ROUTING FAILURE DIAGNOSIS: SYN LOSS DUE TO PHY LAYER INTERFERENCE AND RADIO RX STATE ERROR DIAGNOSED AS DOMINANT ROOT CAUSE.

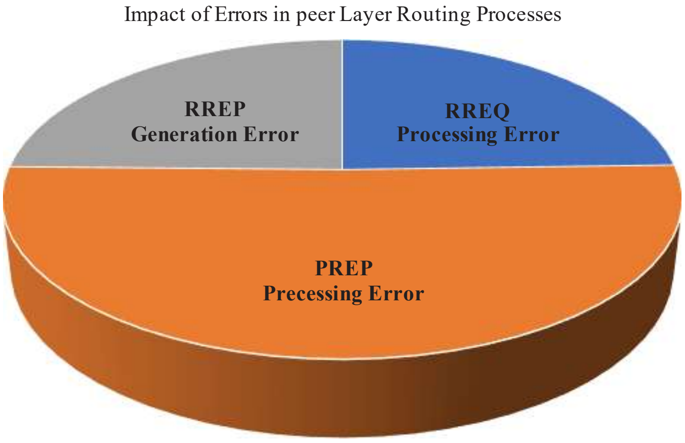

FIG. 11. IMPACT OF ERRORS IN PEER LAYER ROUTING PROCESSES: ERRORS IN RREP GENERATION AND RREP PROCESSING ARE DIAGNOSED AS MAIN SOURCE OF ROUTING CRITICAL FAILURE

In case of 180 sec probe interval and 1 packet per 10 secs, the diagnosis communication overhead is lesser as compared with other cases. This is due to smaller detection latency for a shorter probe period. Consequently, LDA performs better and transmits fewer fault reports per probe interval, reducing diagnosis communication overhead.

4.4 Discussion and Comparison

The additional diagnosis traffic overhead produced by the proposed monitoring system is compared with Sympathy [6] that is designed to collect all necessary node metrices for root cause analysis at sink. However in the proposed monitoring system, diagnosis communication with CH only takes place if LDA deduces external failure source. This decentralized distribution of diagnosistic work load is energy efficient and generates less overhead. As, wireless communication process incurs more energy than computation. In comparison, Sympathy [6] produces 30% overall diagnosis overhead due to periodic transmission of node metrices even in case of no network exception. To compare overhead, ECDFs have been selected. In **Fig. 13**, ratio of the diagnosis traffic to the overall network traffic is represented on x-axis, and ECDF on y-axis. The proposed system is compared with varying the probe interval against different metric periods of Sympathy. As shown in Fig.13, Sympathy is significantly outperformed by the proposed system as LDA transmits fault reports to CH on need basis only reducing diagnosis communication overhead.

5. CONCLUSION

This work presents a diagnostic agent based inter-process communication aware monitoring system for wireless sensor networks. The diagnostic agent performs probe based examination of process execution in communication protocol stack. Based upon the simulation results, it can be concluded that various factors affect network wide fault diagnosis. The first is granularity of error classification based upon implementation of communication protocols. Secondly, root cause analysis of process failures depends upon the scope and complexity of exception handling code. Lastly, successful process execution both at node and network levels relies on inter-process communication of stacked and peer layer processes.

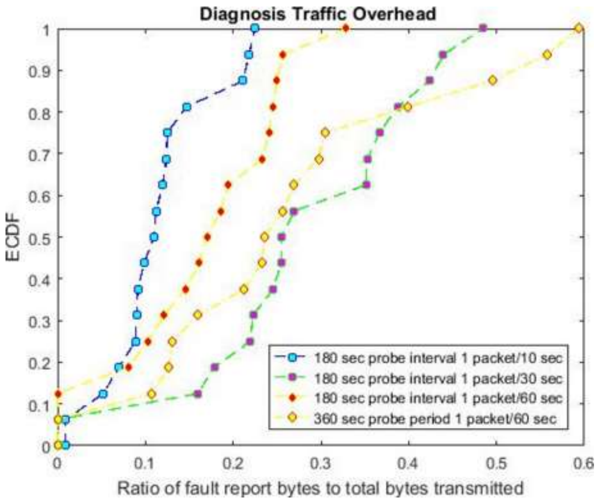

FIG. 12. DIAGNOSIS COMMUNICATION OVERHEAD. EACH LINE DENOTES A SPECIFIC SIMULATION RUN AND EACH POINT REPRESENT RATIO OF FAULT REPORT BYTES TO THE TOTAL BYTES TRANSMITTED

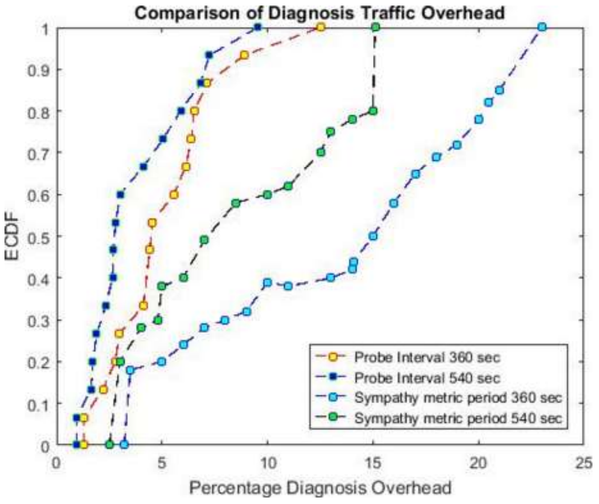

FIG. 13. DIAGNOSIS TRAFFIC OVERHEAD COMPARISON WITH SYMPATHY FOR VARYING VALUES OF PROBE INTERVAL AND SYMPATHY METRIC PERIOD

ACKNOWLEDGEMENT

The authors acknowledge the Department of Computer Science & Engineering, University of Engineering & Technology, Lahore, Pakistan, for continuous support to carry out the research.

REFERENCES

[1] Munir, A., Antoon, J., and Gordon-Ross, A.N.N., "Modeling and Analysis of Fault Detection and Fault Tolerance in Wireless Sensor Networks", ACM Transactions on Embedded Computing Systems, Volume 14, No. 1, pp. 3, 2015.

[2] Muhammed, T., and Shaikh, R.A., "An Analysis of Fault Detection Strategies in Wireless Sensor Networks", Journal of Network and Computer Applications, Volume 78, pp. 267-287, 2017.

[3] Qiu, L., Bahl, P., Rao, A., and Zhou, L., "Fault Detection, Isolation, and Diagnosis in Multihop Wireless Networks", Technical Report MSR-TR-2004-11, Microsoft Research, Redmond, WA, 2003.

[4] Fahmy, H.M.A., "Wireless Sensor Networks: Concepts, Applications, Experimentation and Analysis", Springer, 2016.

[5] Pediaditakis, D., Tselishchev, Y., and Boulis, A., "Performance and Scalability Evaluation of the Castalia Wireless Sensor Network Simulator", Proceedings of 3rd International ICST Conference on Simulation Tools and Techniques, Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, pp. 53, 2010.

[6] Ramanathan, N., Chang, K., Kapur, R., Girod, L., Kohler, E. and Estrin, D., "Sympathy for the Sensor Network Debugger", Proceedings of 3rd ACM International Conference on Embedded Networked Sensor Systems, pp. 255-267, 2005.

[7] Liu, Y., Liu, K., and Li, M., "Passive Diagnosis for Wireless Sensor Networks", IEEE/ACM Transactions on Networking, Volume 18, No. 4, pp. 1132-1144, 2010.

[8] Gong, W., Liu, K., and Liu, Y., "Directional Diagnosis for Wireless Sensor Networks", IEEE Transactions on Parallel and Distributed Systems, Volume 26, No. 5, pp. 1290-1300, 2015.

[9] Ringwald, M., Römer, K., and Vitaletti, A., "Passive Inspection of Sensor Networks", In International Conference on Distributed Computing in Sensor Systems, pp. 205-222 Springer, Berlin, Heidelberg, 2007.

[10] Koubâa, A., Chaudhry, S., Gaddour, O., Chaari, R., Al-Elaiwi, N., Al-Soli, H., and Boujelben, H., "Z-Monitor: Monitoring and Analyzing IEEE 802.15. 4-Based Wireless Sensor Networks", IEEE 36th Conference on Local Computer Networks, pp. 939-947, 2011.

[11] Tennina, S., Gaddour, O., Koubâa, A., Royo, F., Alves, M., and Abid, M., "Z-Monitor: A Protocol Analyzer for IEEE 802.15.4-Based Low-Power Wireless Networks", Computer Networks, Volume 95, pp. 77-96, 2016.

[12] Rodenas-Herráiz, D., Fidler, P.R., Feng, T., Xu, X., Nawaz, S., and Soga, K., "A Handheld Diagnostic System for 6LoWPAN Networks", IEEE 13th Annual Conference on Wireless On-Demand Network Systems and Services, pp. 104-111, 2017.

[13] Awad, A., Nebel, R., German, R., and Dressler, F., "On the Need for Passive Monitoring in Sensor Networks", IEEE 11th Euromicro Conference on Digital System Design Architectures, Methods and Tools, pp. 693-699, 2008.

[14] Carrera, A., and Iglesias, C.A., "Towards Fault Diagnosis based on Agent Technology for Wireless Sensor Networks", IEEE 4th International Conference on Future Generation Communication Technology, pp. 1-6, 2015.

[15] Al-Anbagi, I., Erol-Kantarci, M., and Mouftah, H.T., "A Survey on Cross-Layer Quality-of-Service Approaches in WSNs for Delay and Reliability-aware Applications", IEEE Communications Surveys & Tutorials, Volume 18, No. 1, pp. 525-552, 2016.

[16] Zafar, A., Wajid, B., and Akram, B.A., "A Hybrid Fault Diagnosis Architecture for Wireless Sensor Networks", IEEE International Conference on Open Source Systems & Technologies, pp. 7-15, 2015.

[17] Perkins, C., Belding-Royer, E., and Das, S., "Ad Hoc On-Demand Distance Vector Routing", No. RFC 3561, 2003.

[18] "IEEE Standard for Low Rate Wireless Networks, Standard 802.15.4", IEEE Computer Society and IEEE Standards Association, 2015.