Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 286424, 9 pages
doi:10.1155/2012/286424

Research Article

Remote Industrial Sensor Network Monitoring Using M2M Based Ethical Sniffers

Syed Muhammad Asad Zaidi,[1] Jieun Jung,[1] Minsoo Kang,[2] Byunghun Song,[1] and Ki-Hyung Kim[3]

[1] *IoT Convergence Research Center, Korea Electronics Technology Institute (KETI), Seongnam 463-816, Republic of Korea*
[2] *RFID/USN Program, Korea Evaluation Institute of Industrial Technology, Seoul 135-080, Republic of Korea*
[3] *School of Information & Computer Engineering, Ajou University, Suwon 443-749, Republic of Korea*

Correspondence should be addressed to Syed Muhammad Asad Zaidi, zaidix@gmail.com

Received 11 May 2012; Revised 1 November 2012; Accepted 1 November 2012

Academic Editor: Chuan Foh

Diagnosing the deployed network efficiency and anomaly detection, which is an important research issue in traditional networking systems, has not been carefully addressed in industrial wireless sensor networks. Although recent wireless systems for industrial automation such as ISA100.11a employ device management protocols, these protocols generate and report a large amount of status information from individual sensor nodes. Also, these protocols do not capture influences on network performance from external sources such as malicious nodes or interference from other networks. We propose a latent network diagnosis system (LaNDS) for industrial sensor networks. LaNDS employs a packet sniffing method for efficiently evaluating network performance and instantly identifying degradation causes of networking performance. LaNDS adopts an efficient network evaluation approach for detecting abnormalities from both internal and external causes. In our proposed monitoring scenario, special sniffer devices having M2M capability (WiMAX interface) are used to monitor the industrial sensor network by employing ethical sniffing. Our approach does not incur additional traffic overhead for collecting desired information. For evaluation, we have tested LaNDS locally on an ISA100.11a based sensor network in a lab environment and have validated the efficiency of the system based on the possible erroneous cases of industrial sensor network.

1. Introduction

Wireless sensor networks (WSNs) enable users to interact with the physical environment at an unprecedented level. Their tiny, cheap, energy efficient, robust, and scalable properties have resulted in their deployment for a wide range of applications, such as military, health monitoring [1, 2], data acquisition in hazardous environments; and habitat monitoring [3–5]. Much of the research has been done on WSN in general issues like routing, reliability, QoS, energy consumption, and security, but very little work has been done targeting the in situ network diagnosis for testing operational sensor networks. Existing tools (debugger, testbed, simulation, and emulation) do not work for deployed networks, thus it is of great importance to provide network developers and administrators with some useful information on a system's working status. WSNs are by nature error prone and have unsatisfactory reliability, encountering various faults and failures during their operation. Diagnosis and evaluation of the deployed networks will enhance the applicability, reliability, and efficiency of WSNs.

While significant work on conventional network management tools exists [6], WSN counterparts have been slow to gain such efficient and useful tools. One of the main challenges for WSN anomaly detection is determining where to embed the intelligence for detecting and localizing anomalies. Another key requirement for any anomaly detection strategy is to cater for the needs and feedback of the human operator. A user-friendly detection strategy should provide several modes of notification, such as email and SMS alerts, and adapt its frequency of alerts to user feedback [7].

The next challenging problem for WSN management is not to generate a lot of control packets. For proactive approach, most of the current tools implant debugging tools into the sensor nodes that periodically report the predefined parameters and internal status to the sink. For example, sympathy [8] actively collects run-time status from sensor nodes like routing table and flow information and detects possible faults by analyzing node status together with observed network exceptions, while, in the proposal by Zhao et al. [9], each node locally collects residual energy, link loss rate, and packet count and transmits it to sink for detailed analysis. These proactive techniques not only cause excessive computational operations on sensor nodes but also result in early depletion of the remaining energy level.

A recent review article on anomaly detection in WSNs [10] focuses on data anomalies, mainly due to security attacks, and the statistical approaches for detecting them. Because of their tight coupling to often harsh physical environments, WSNs and other networks used in extreme conditions (e.g., in space [11]) are more likely to experience anomalies related to connectivity or hardware failures than conventional networks. Recent work also focuses on devising detection strategies that target network level [8, 12], data level [13, 14], or node and data level [15, 16] anomalies.

One shortfall of the existing strategies is that none of them comprehensively addresses network, node, and data level anomalies in WSNs. Moreover, these problems are often not encountered during predeployment tests also, because the environmental conditions that trigger these problems are hard to simulate in the lab. One common difficulty is determining which metrics should be used to evaluate the health of the sensor network. Many of these sensor networks are used to report measurements about the environment, such as temperature, light level, and sound level. These nodes also have awareness of internal metrics such as processor utilization, current draw, and battery voltage level. When fielded to monitor the environment, these nodes are usually configured to sense and report on only a few of these data streams. The lack of comprehensive anomaly detection strategies for WSNs contributes to slower adoption and more frustration in deploying and maintaining these networks. Therefore, sensor networks have to be inspected in-situ on the deployment site to identify and locate failures and their causes. Determining the health of a sensor network is a difficult, yet important task. It is crucial that all sensor data reported by a sensor network is accurate so that it can be trusted by its user. Nodes may malfunction due to loss of power, extreme environmental conditions such as temperature or precipitation, or physical damage caused by falling debris or wildlife. Monitoring the health of a sensor network helps increase its trustworthiness by reporting nodes that may be malfunctioning.

In this paper, we propose a portable and user friendly diagnosis and monitoring tool "Latent Network Diagnosis System (LaNDS)" that observes, monitor and evaluates the ISA100.11a [17] based industrial sensor network using Freescale MC1322X USB dongle. Motivated from wireshark, a network protocol analyzer, our Java based application receives the ongoing transmission in the surrounding area of M2M enabled sniffer device through WiMAX interface and shows each OSI Layer parameters in a live view tab in a tree form. From this view, the user can perform deep packet inspection (DPI). Moreover there is a statistical view tab which, as the name suggests, outputs the live sniffed data in the shape of charts and graphs. From the graphs we can see the size and number of data and acknowledgment packets sent in the network, the composite packet rate, channel activity statistics and the number of packets each participating node has sent. A topology view tab shows virtual network topology showing nodes interconnected with other nodes based on their communication pattern (i.e., data packets sent to and from). Based on the output from the live view, the statistical view, and the topology view tab, we can get a quick and complete picture of the network environments. We can also identify the problematic node, inspect message contents down to the bit level, and share scenarios with vendors. Through the heedful analysis, we can keep an eye on network anomalies (loss of connectivity, intermittent connectivity, and broadcast storm) and hardware anomalies (node failure and node resets). Moreover, we have also devised ways not only to identify common security attacks like denial of service (DoS) and sinkhole attack [18, 19], but also to identify the source malicious node responsible for the attack.

The rest of this paper is organized as follows. Section 2 gives an abridged version of related work done in the past. Monitoring scenario proposed by us for efficient and large scale monitoring of industrial sensor network through LaNDS has been proposed in Section 3. Section 4 describes the design and description of our developed tool. ISA100.11a network deployed in a lab environment to test LaNDS has been explained in Section 5, while the final experimental evaluation has been done in Section 6 with special focus on common security threats and their detection scenarios. Finally we have concluded our paper in Section 7.

2. Related Work

Research on WSN monitoring and diagnosis of already deployed networks has not gained much attention while it is a critical issue as it has a direct influence on successful, efficient, and secure network operations. One of the reasons for this being neglected is that they are notoriously difficult to develop and debug. Most existing tools for WSN diagnosis are built on proactive approach, in which each sensor employs a debugging agent to collect the relevant status information and reports to the sink by periodically transmitting specific control messages which in turn decreases the network efficiency and can accelerate the energy depletion rate of sensor nodes. Some researchers propose to monitor sensor networks by scanning the residual energy [19] of each sensor node and collecting the aggregates of parameters of sensors where the in-network processing is leveraged. By collecting such information, the sink is aware of the network conditions. Periodic transmission of metrics from nodes to the sink is not a new idea. MintRoute [20] includes periodic transmission of neighbor tables to aid in debugging at the sink. However, it neither includes other metrics nor performs failure analysis at the sink.

Some debugging systems [21, 22] aim to detect and debug software failures in sensor nodes. For example, with

Clairvoyant [22], a source level debugger, a developer can wirelessly connect to a WSN and execute standard debugging commands including break, step, watch, and backtrace. But it is not enough since it does not cater for network and hardware issues. Sympathy [8] is an advanced debugging tool that detects and debugs the failures in a sensor network. Sympathy has selected metrics like neighbor list and traffic flow that enable efficient failure detection and includes an algorithm that root causes failures and localizes their sources in order to reduce overall failure notifications. It also applies an empirical decision tree to determine the most likely root causes for an observed exception.

The nucleus network management system (NMS) infrastructure helps sensor network applications export debugging and monitoring information [23]. Nucleus' support for exporting statistics and recording application metrics is not only easy to use but also to lightweight, but the limitation of NMS is that it does not provide infrastructure to analyze these metrics. Furthermore, these metrics consume more than double the RAM required for the rest of the stack.

A WSN, unlike an enterprise network, is featured by its hierarchical multilevel structures, which can hardly be approximated by the bipartite graph model used in most of the enterprise network monitoring tools, for example [24]. It is also impractical to maintain the network dependencies as stable inputs in highly dynamic and self-organized sensor networks.

3. Proposed Monitoring Mechanism

We have defined a new term called "Ethical Sniffing" in this paper. As the name suggests, we have incorporated a sniffing technique in order to monitor the network and in turn come up with detailed analysis, statistics, and graphs. At any given time, the coverage of our M2M enabled sniffer device will be limited to a small part of a sensor network due to the small radius of the sensor node radio signals; therefore our goal is to devise other ways to extend our monitoring range to a complete or partial (significant part) network area. We have suggested some of the ways how to implement these M2M based sniffer devices in order to get a wider monitoring scope of the network.

(a) *Crucial area coverage*: one approach is to put the sniffer devices only at particular areas in a network called crucial areas. We can place these devices at areas where there is high node density, where data delivery is crucial to delay and loss or where we anticipate an intermittent connectivity or data loss.

(b) *Network of sniffer devices*: we can also place multiple sniffing devices in the network haphazardly or separated by some distance and create a separate network of sniffer devices. Hence every device will keep an eye on its surrounding area and monitoring of the entire network can be done.

(c) *Mobile sniffing devices*: another approach can be employed using mobile sniffing devices that will sniff the data required for monitoring as it traverses through the network in a fixed predefined path or as directed by the network administrator through WiMAX downlink. The mobile sniffer devices can also be programmed to be active and monitor the network periodically.

(d) *Divide and monitor*: real-time active monitoring of entire network can be done efficiently by dividing the network into logical subnets such that each subnet has a dedicated mobile sniffing device. Figure 1 exhibits the scenario.

4. System Design and Description

In this section, we will briefly describe the software level system design and the user interface of our network monitoring tool "LaNDS". LaNDS is an all-in-one tool tailor made for WSN and its reliable, efficient, and user friendly output makes it quite different and unique from other existing tools of the same kind. M2M based sniffer devices exploit the wireless channel properties and send the sniffed packets to remote server having our LaNDS monitoring tool running. Multiple networks can be monitored remotely by a single server having multiple instances of LaNDS running.

LaNDS is a java based application and therefore can be run on a variety of operating systems including smart phones and tablets. Programmatic layout has been summarized in Figure 3. When the application runs, the interface is loaded by calling the "GUI" class. When the user clicks on the start button to initiate the sniffing process, "CaptureOptions" java class is triggered and a new window appears. In this window, the user is prompted to select the target channels from the radio button as shown in Figure 2. Since ISA100.11a runs on the principle of channel hopping, target channels here refer to the IEEE802.15.4 channels on which sniffing is to be performed. Upon pressing "Start," an object of sniffer class is made and "startSniffing()" process is executed. For every single channel to be sniffed, a separate Freescale MC1322x sniffer USB dongle is required and a separate agent (object) is created as "SnifferAgent". Each thread corresponding to its agent is parsed sequentially via the "initSerialReader()" function. This continues in a loop and every data or ACK packet received on each channel is read serially and the header parameters are arranged in a tree-like structure and later displayed in the live view tab of the main topology view tab. Within the loop boundary, another function "drawFigures()" is also called repeatedly whose sole purpose is to update the graphs and charts in the statistical view tab.

To be more specific, LaNDS consists of five major components: (i) serial reader, (ii) signal merger, (iii) channel manager, (iv) network manager, and (v) Security manager. Detailed system design summarizing the complete process from frame sniffing to data display stage has been shown in Figure 4.

At first, a serial reader reads sniffed packets from a single or multiple K1322-Sniffer interfaces and sends special commands to sniffing protocol for resetting the sniffer CPU, setting capturing channel and capturing mode, and so forth. After that, the signal merger receives multiple data streams

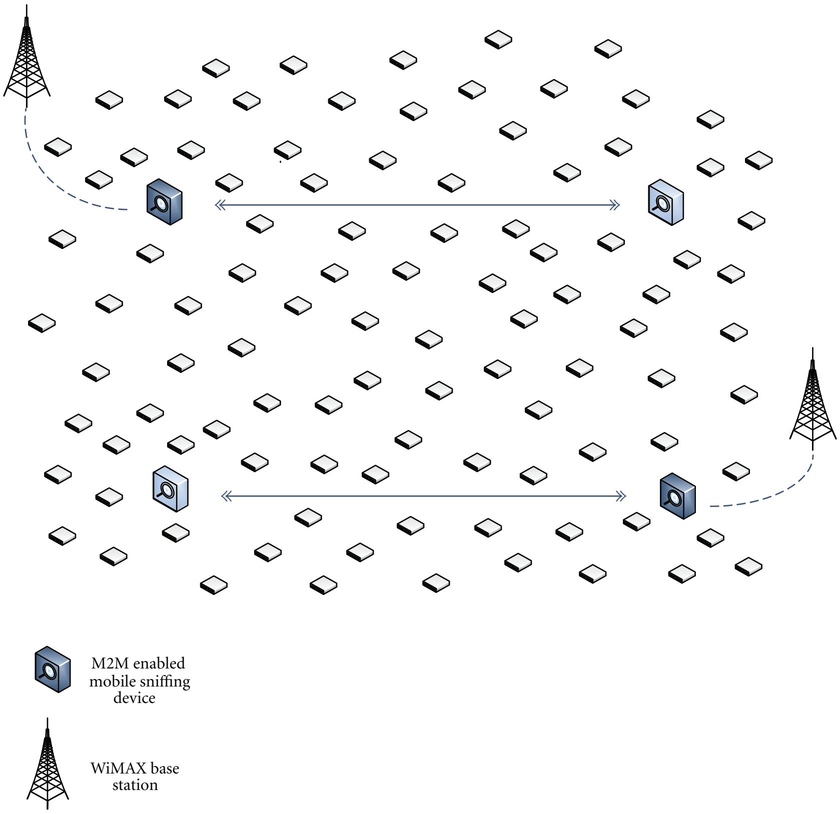
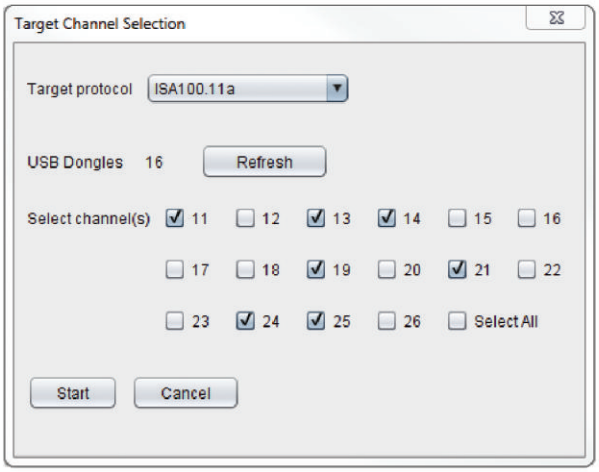


Figure 1: Divide and monitor—M2M enabled mobile sniffing devices for wide area network monitoring.



Target Channel Selection

Target protocol: ISA100.11a

USB Dongles: 16

Select channel(s): ☑ 11 □ 12 ☑ 13 ☑ 14 □ 15 □ 16 □ 17 □ 18 ☑ 19 □ 20 ☑ 21 □ 22 □ 23 ☑ 24 ☑ 25 □ 26 □ Select All

Figure 2: Target Channel Selection.

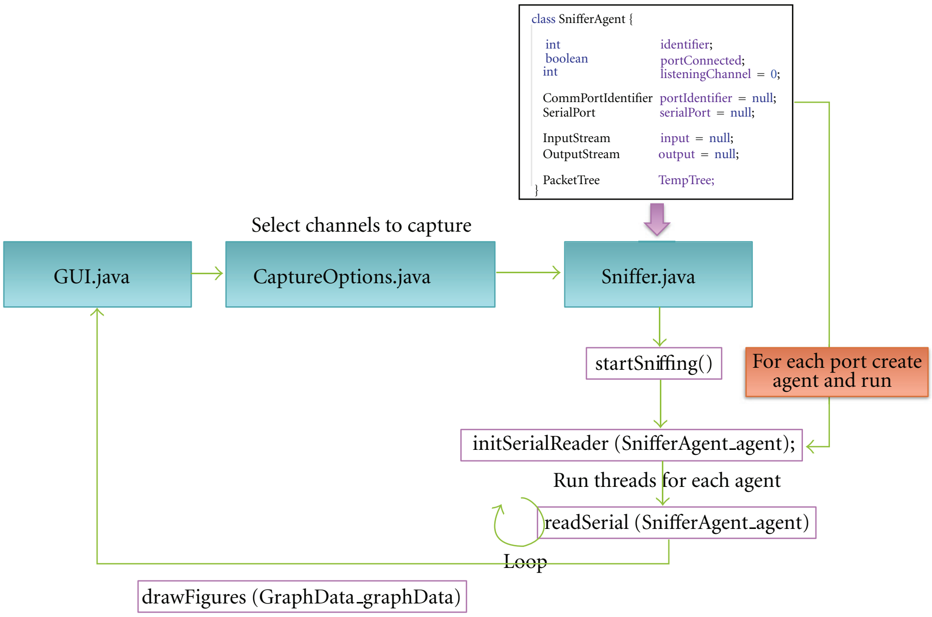
from the serial reader and combines them to serialized frames. Each frame contains captured packet from a single interface. Since K1322-Sniffer interfaces capture signals from multiple channels, correctly differentiating these signals into frames is an important task.

Network manager processes capture packets, extracts device and network information, and prepares statistical information related to network. This component also estimates efficiency of network based on collected statistical information. The parameters of network efficiency include TCP/UDP throughput, header compression efficiency, packet loss rate, packet delivery latency, and so forth. Channel manageris responsible for quantifying channel usage. ISA100.11a uses three types of hopping schemes which are (i) slotted hopping, (ii) slow hopping and (iii) hybrid hopping. Channel manager estimates how each channel is efficiently used. Channel manager also evaluates interference and intrusions, and suggests which channels should be avoided for efficient data exchange.

The purpose of Security manager is tomonitor the device abnormality and to detect various security threats. In order to provide industry-level wireless network system, ISA100.11a should guarantee robustness from security threats. Security manager in proposed framework detects denial of service attacks, channel jamming, node impersonation, and wormhole attacks and notifies the network administrator with special warning methods. After the following processes are complete, sniffed data is finally posted to the LaNDS GUI categorized as live, statistical, and network view tab. More detailed explanation along with figures is provided in Section 6.

5. Experimental Setup and Operations

For evaluation and testing purposes of our LaNDS tool, we carried out an experiment on an ISA100.11a network in



```
class SnifferAgent {

  int                  identifier;
  boolean              portConnected;
  int                  listeningChannel = 0;

  CommPortIdentifier   portIdentifier = null;
  SerialPort           serialPort = null;

  InputStream          input = null;
  OutputStream         output = null;

  PacketTree           TempTree;
}
```

Select channels to capture
GUI.java
CaptureOptions.java
Sniffer.java
startSniffing()
For each port create agent and run
initSerialReader (SnifferAgent_agent);
Run threads for each agent
readSerial (SnifferAgent_agent)
Loop
drawFigures (GraphData_graphData)

Figure 3: Sniffer class Java code Model.

a local lab environment. A group of USB dongle sniffers without M2M interface were used as sniffer device. Figure 5 shows the basic network and sniffing model. Multiple sensor nodes were connected to a gateway, while, on the other hand, a set of sniffing devices (USB dongles) each for a particular IEEE 802.15.4 channel sniffed the ongoing wireless communication between the neighboring ISA100.11a based sensor nodes. Laptop running our tool LaNDS interpreted and prepared packets after receiving the sniffed data sequentially from each sniffer data stream. Finally the network statistics were viewed on the laptop screen in three different tabs as explained in the previous section.

On the network side we used twelve ISA100.11a based sensor nodes and a gateway as shown in Figure 6. These gateway controlled nodes transmit data packets haphazardly to each other using the frequency hopping technique as per ISA100.11a standard. In our network scenario

(a) at IEEE802.15.4 layer, 16-bit short addressing has been used with PAN ID compression and security disabled;

(b) in ISA100.11a protocol;

  (i) DHDR subheader specifies that slow hopping offset and DAUX subheader are included whereas signal quality in ACK is excluded.

  (ii) ISA100.11a compress feature has been set to null as per DROUT sub-header;

(c) IPHC has been used in our network with IPNHC bit set, from which layer 3 header and layer 4 header are extracted.

On the sniffer side, in order to capture all channels of IEEE802.15.4, we used 16 Freescale MC1322x USB dongles mounted on a 16-port USB hub as shown in Figure 6. The USB hub was connected to a PC running LaNDS and it captured the live packets as they were being sent by the target nodes.

6. Network Diagnosis and Anomaly Detection

This section shows the output of our developed tool LaNDS when it is in the middle of the live traffic the monitoring stage. As discussed in the previous sections that, at an instance, the tool monitors a part of an already deployed WSN using an ethical sniffing technique by exploiting the intrinsic characteristics of wireless medium and, unlike other existing diagnosis and monitoring tool, it does not result in rapid energy depletion or additional control packet dissemination through the network.

The output of LaNDS is divided into three tabs each showing different representation of sniffed data. The first tab shows live packets sniffed in a tree form. Every packet is associated with the packet number and the channel number on which it was captured as shown in Figure 7. In this view, we can have a deep packet inspection by expanding the target layer header and can view the parameters and addresses present there.

The other two tabs shown in Figures 8 and 9 are more of a graphical representation of data. In the statistical view tab, there are four graphs showing network statistics (Figure 8). The first graph shows the channel utilization. A bar chart shows the number of packets sent through each channel. The second graph is a line graph showing the network efficiency; it shows the number of packets transmitted as the time progresses. Another graph on the bottom left classifies the packet according to the packet type. A pie chart shows the distribution of broadcast, multicast, and unicast packets.

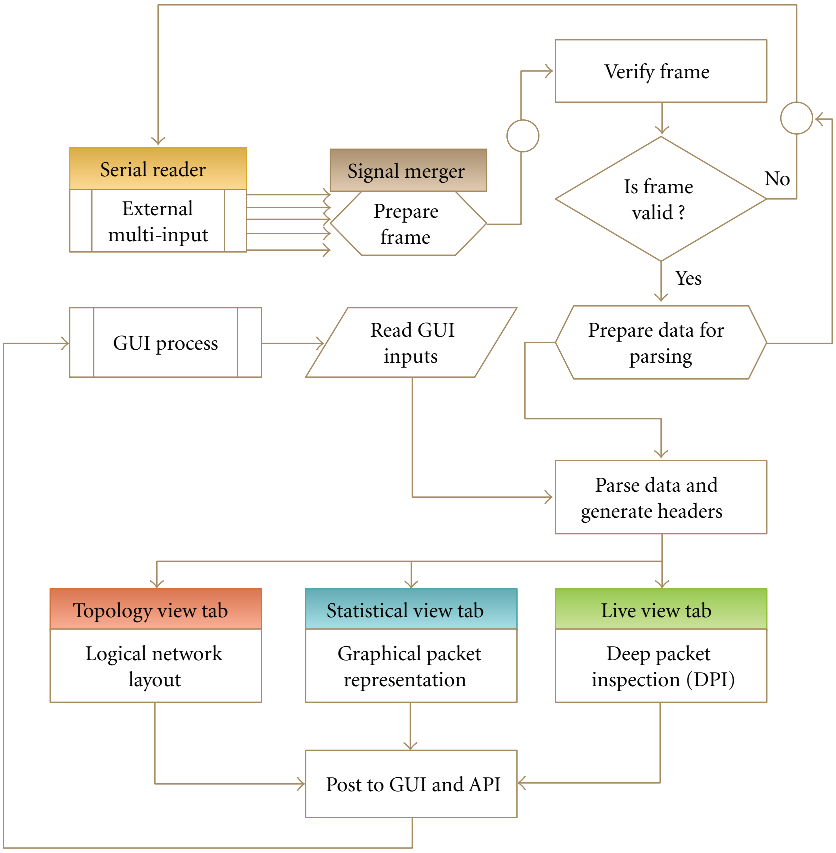


Figure 4: Flowchart of system design.

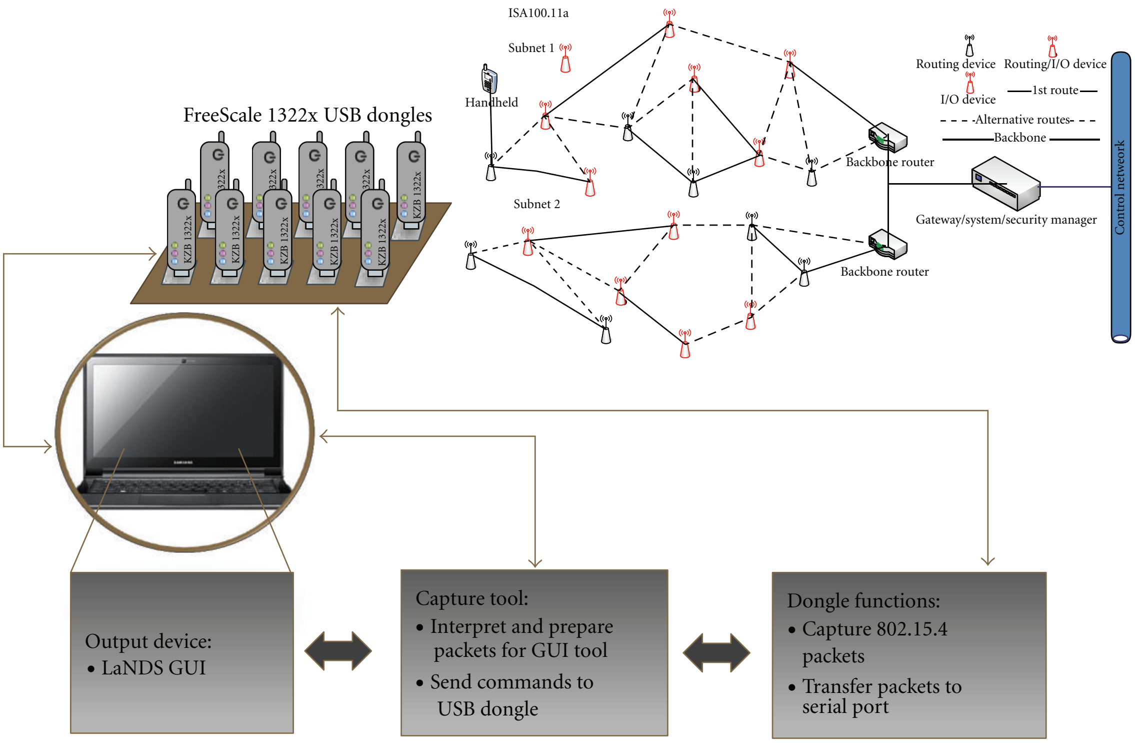


Figure 5: ISA100.11a network based monitoring scenario.

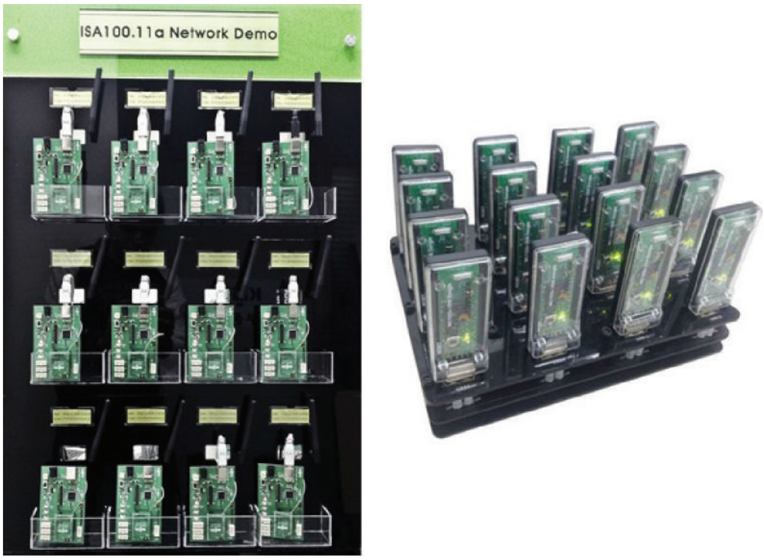
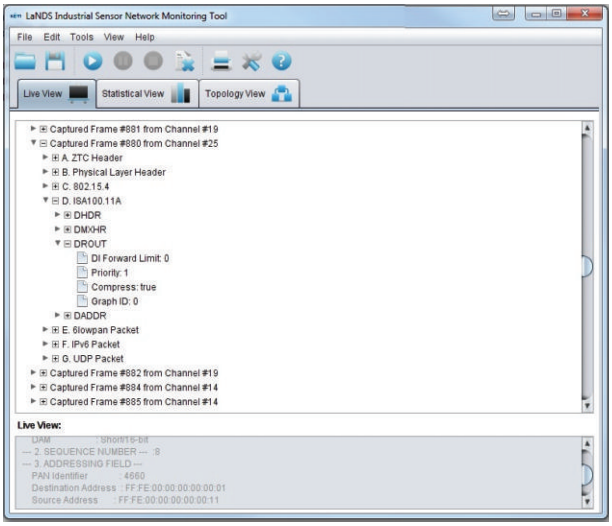


Figure 6: (a) Experimental network with eight ISA100.11a sensor nodes (b) Freescale MC1322X USB dongle hub.



```
▸ ⊞ Captured Frame #881 from Channel #19
▾ ⊟ Captured Frame #880 from Channel #25
    ▸ ⊞ A. ZTC Header
    ▸ ⊞ B. Physical Layer Header
    ▸ ⊞ C. 802.15.4
    ▾ ⊟ D. ISA100.11A
        ▸ ⊞ DHDR
        ▸ ⊞ DMXHR
        ▾ ⊟ DROUT
              DI Forward Limit: 0
              Priority: 1
              Compress: true
              Graph ID: 0
        ▸ ⊞ DADDR
    ▸ ⊞ E. 6lowpan Packet
    ▸ ⊞ F. IPv6 Packet
    ▸ ⊞ G. UDP Packet
▸ ⊞ Captured Frame #882 from Channel #19
▸ ⊞ Captured Frame #884 from Channel #14
▸ ⊞ Captured Frame #885 from Channel #14
```

```
  DAM                 : Short/16-bit
--- 2. SEQUENCE NUMBER ---  :8
--- 3. ADDRESSING FIELD ---
  PAN Identifier          : 4660
  Destination Address : : FF:FE:00:00:00:00:00:01
  Source Address      : FF:FE:00:00:00:00:00:11
```

Figure 7: LaNDS Deep Packet Inspection.

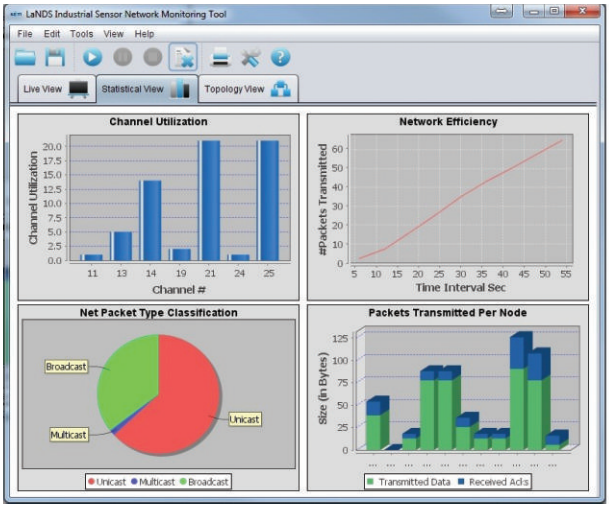


Figure 8: LaNDS Statistical View Tab.

The last graph shows the packet size; each node has transmitted since the sniffing by LaNDS began.

The third tab, the network topology view tab, shows the topology view of the surrounding network. Sensor nodes sending data to each other will be interconnected with each other as shown in Figure 9. In this exemplary view, we can easily infer that the middle node is broadcasting packets to the surrounding nodes.

By studying typical protocols used in sensor networks, we found that a great deal of information about the state of the sensor network can be inferred from a message trace. For example, we can detect node failures and node reboots without modifying the protocols used in the sensor network. We can even infer routing topologies or detect the existence of network partitions without touching the sensor network.

LaNDS was originally designed for diagnosis and monitoring of WSN, but it also provides some additional benefits against detection and elimination of active security attack. Generally, if an attack is successful on a WSN, it is very difficult to identify and to eliminate the attack, but, through LaNDS, we cannot only identify the threat but also reach the source of the attack in a short time. From the security point of view, some of the protection scenarios are mentioned below.

6.1. Protection from External Malicious Nodes. Inspired by the approach of some cellular operators to maintain a list of legitimate users and implicitly prevent unauthoritative access, we have proposed to maintain a whitelist containing addresses of all devices deployed in our network, and if during sniffing, we encounter any external node (node

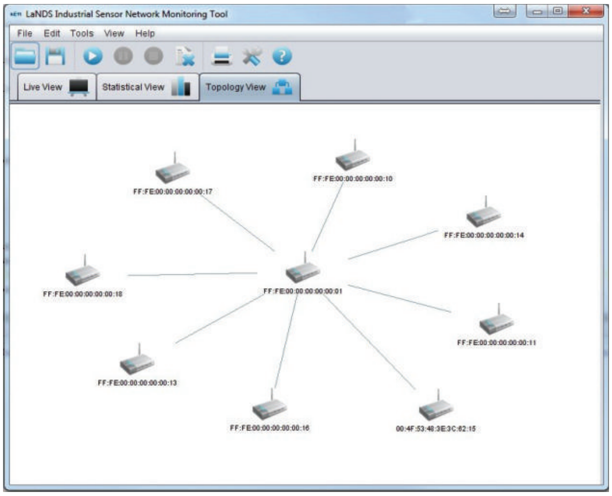


Figure 9: LaNDS Topological View Tab.

address not mentioned in the whitelist), it gets highlighted in the topology view and the next step would be to eliminate it from the network. Similarly, we have also maintained a blacklist through which nodes having a particular MAC address (known attackers) can be identified.

6.2. Protection against Internal Attack. The statistical view tab not only shows the graphical representation of the network, nodes, channels, and packet types, but we can also exploit it for detecting successful security attacks from inside nodes. Security based monitoring through LaNDS can provide the following:

(i) detecting a security attack,

(ii) identifying security attack type,

(iii) pin pointing source of attack. (graphically and via MAC address of malicious node).

Here we have shown the identification of the most common security attacks (Sinkhole attack and DoS attack). However other security attacks like selective forwarding attack, acknowledgment attack, and wormhole attack can also be identified after an acute study.

6.2.1. Blackhole/Sinkhole Attack Identification. Since Blackhole attack attracts all the traffic in the sensor network towards a single node, identification can be done in the following manner:

> *if number of packets with same destination/time period* $\gg$ *normal packet rate*
>
> **OR**
>
> *If number of packets with same destination/time period* $\geq$ *threshold.*

threshold and time period can be set by the administrator and it depends on the type of network; threshold for multimedia based sensor network will be high than for temperature monitoring based sensor network.

6.2.2. DoS Attack Identification. DoS attack has severe impact on network as it tries to exhaust the resources available to the victim node by sending extra unnecessary packets and thus preventing legitimate network users from accessing services or resources. Its identification can be done through a simple approach:

> *if number of packets with same sender address/time period* $\gg$ *normal packet rate*
>
> **OR**
>
> *if number of packets with same sender address/time period* $\geq$ *threshold.*

Here too, threshold and time period can be set by administrator and it depends on type of network; threshold for multimedia based sensor network will be high for temperature monitoring based sensor network.

7. Conclusions

Although much research has been done in various aspects of wireless sensor networks, little work has been done towards a diagnosis tool for monitoring the statuses of operational systems in the field. This paper proposes a new technique used for remote monitoring of already deployed industrial ISA100.11a based sensor network using LaNDS. LaNDS receives the data through the WiMAX link from M2M enabled sniffer devices and later processes it to give a detailed insight of the network. LaNDS system can be thought as a passive diagnosis tool that exploits the wireless medium characteristics, but it can be efficiently applied to any already deployed industrial sensor network system giving live traffic analysis and statistics in the form of text and figures.

What makes LaNDS system unique is that unlike other tools it does not periodically pull the status and other predefined parameters from the nodes; thus it does not result in early depletion of already scarce available energy of sensor nodes. Moreover, it does not disseminate additional control packet throughout the network, and hence it does not have adverse effects on network efficiency either. The only limitation is that it provides monitoring only to a part of a network at a time and, in order to get an overview of the entire network, we have to install multiple mobile devices which add to cost and complexity.

LaNDS is quite a user friendly tool and the output is presented in three different categories separated by tabs. We cannot only see parameters of each header in the form of tree, but different graphs in the statistical view tab tells us different aspects and hidden anomalies present in the network. Moreover it also shows a network topology view which can further assist us to monitor the live network quite efficiently.

Network monitoring using LaNDS can be further enhanced to detect active malicious attacks and to pin point the source at node level. As a result, the network once attacked can be quickly brought back to normal routine and the sensor network can be protected from prolonged internal and external attacks.

Acknowledgments

This work is supported in part by the IT R&D program of MKE/KETI (number 2011-10040909) and the Convergence ITRC (Convergence Information Technology Research Center) support program (NIPA-2012-H0401-12-1004) of MKE/NIPA.

References

[1] T. Gao, D. Greenspan, M. Welesh, R. R. Juang, and A. Alm, "Vital signs monitoring and patient tracking over a wireless network," in *Proceedings of the 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '05)*, September 2005.

[2] L. Gu, D. Jia, P. Vicaire et al., "Light weight detection and classification for wireless sensornetworks in realistic environments," in *Proceedings of the 3rd ACM International Conference on Embedded Networked Sensor Systems*, November 2005.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[4] J. Kahn, R. Katz, and K. Pister, "Next century challenges: mobile networking for smart dust," in *Proceedings of the Annual ACM International Conference on Mobile Computing MobiCom (MobiCom '99)*, August 1999.

[5] X. Zhang, M. Wei, P. Wang, and Y. Kim, "Research and implementation of security mechanism in ISA100.11a networks," in *Proceedings of the 9th International Conference on Electronic Measurement and Instruments (ICEMI '09)*, pp. 4716–4721, August 2009.

[6] M. Subramanian, *Network Management: An Introduction To Principles and Practice*, Addison-Wesley Longman Publishing Co., Inc., Boston. Mass, USA, 1999.

[7] R. Jurdak, R. Wang, O. Obst, and P. Valencia, "Wireless sensor network anomalies: diagnosis and detection strategies," in *Intelligence-Based Systems Engineering, ISRL*, vol. 10, pp. 309–325, Springer-Verlag, Berlin, Germany, 2011.

[8] N. Ramanathan, K. Chang, L. Girod, R. Kapur, E. Kohler, and D. Estrin, "Sympathy for the sensor network debugger," in *Proceedings of the ACM Conference on Networked Sensor Systems (SenSys '05)*, pp. 255–267, 2005.

[9] J. Zhao, R. Govindan, and D. Estrin, "Residual energy scan for monitoring sensor networks," in *Proceedings of the IEEE Wireless Communication and Networking Conference (WCNC '02)*, vol. 1, pp. 356–362, 2002.

[10] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly detection in wireless sensor networks," *IEEE Wireless Communications*, vol. 15, no. 4, pp. 34–40, 2008.

[11] M. Prokopenko, P. Wang, M. Foreman, P. Valencia, D. Price, and G. Poulton, "On connectivity of reconfigurable impact networks in ageless aerospace vehicles," *Robotics and Autonomous Systems*, vol. 53, no. 1, pp. 36–58, 2005.

[12] S. Rost and H. Balakrishnan, "Memento: a health monitoring system for wireless sensor networks," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad hoc Communications and Networks (Secon '06)*, pp. 575–584, Reston, Va, USA, September 2006.

[13] M. Wälchli and T. Braun, "Efficient signal processing and anomaly detection in wireless sensor networks," in *Proceedings of the European WorkShops on Applications of Evolutionary Computation (EvoWorkshops '09). LNCS*, M. Giacobini, A. Brabazon, S. Cagnoni et al., Eds., vol. 5484, pp. 81–86, Springer, Heidelberg, Germany, 2009.

[14] B. Krishnamachari and S. Iyengar, "Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor networks," *IEEE Transactions on Computers*, vol. 53, no. 3, pp. 241–250, 2004.

[15] N. Ramanathan, L. Balzano, M. Burt et al., "Rapid deployment with confidence: calibration and fault detection in environmental sensor networks," Technical Report, UCLA CENS, Los Angeles, Calif, USA, 2006.

[16] J. Chen, S. Kher, and A. Somani, "Distributed fault detection of wireless sensor networks," in *Proceedings of the Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS '06)*, pp. 65–71, New York, NY, USA, September 2006.

[17] T. Petersen and S. Carlsen, "WirelessHART versus ISA100.11a: the format war hits the factory floor," *IEEE Industrial Electronics Magazine*, vol. 5, no. 4, pp. 23–34, 2011.

[18] Y. Wang, G. Attebury, and B. Ramamurthy:, "A survey of security issues in wireless sensor networks," IEEE Communications Survey, 2006.

[19] J. Zhao, R. Govindan, and D. Estrin, "Residual energy scan for monitoring sensor networks," in *Proceedings of the IEEE Wireless Communication and Networking Conference (WCNC '02)*, vol. 1, pp. 356–362, 2002.

[20] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," in *Proceedings of the IEEE ICC Workshop on Sensor Network Protocols and Applications*, Anchorage, AK, 2003.

[21] N. Ramanathan, K. Chang, L. Girod, R. Kapur, E. Kohler, and D. Estrin, "Sympathy for the sensor network debugger," in *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. 255–267, 2005.

[22] J. Yang, M. L. Soffa, L. Selavo, and K. Whitehouse, "Clairvoyant: a comprehensive source-level debugger for wireless sensor networks," in *Proceedings of the 5th ACM International Conference on Embedded Networked Sensor Systems (SenSys '07)*, pp. 189–203, November 2007.

[23] G. Tolle and D. Culler, "SNMS: application-cooperative management for wireless sensor networks," in *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys '04)*, 2004.

[24] S. Kandula, D. Katabi, and J. P. Vasseur, "Shrink: a tool for failure diagnosis in IP networks," in *Proceedings of the ACM Workshops: Conference on Computer Communications (SIGCOMM '05)*, pp. 173–178, August 2005.

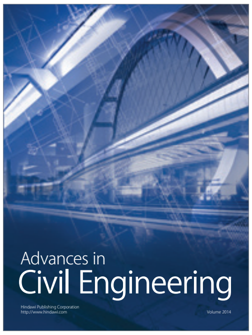
Journal of
Engineering

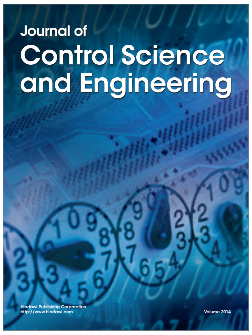
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

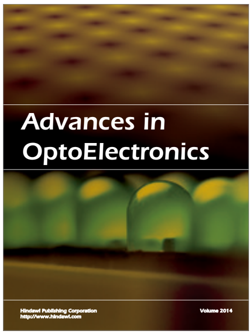
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

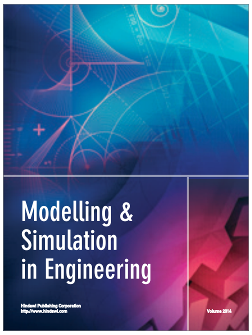
Journal of
Robotics

Journal of
Electrical and Computer Engineering

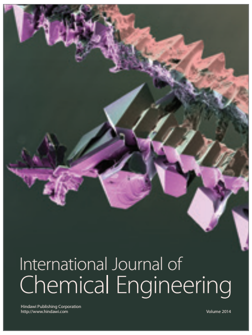

Advances in
OptoElectronics

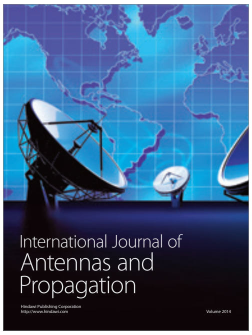
VLSI Design

International Journal of
Navigation and Observation

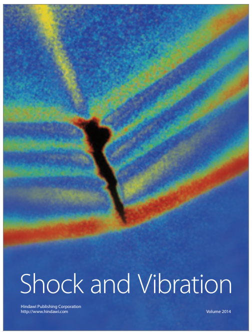

Modelling &
Simulation
in Engineering

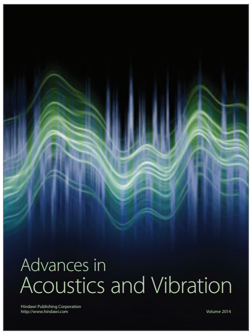
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration