An energy-efficient passive monitoring system for wireless sensor networks

Fernando P. Garcia[*,§], José Neuman de Souza[§,a], Rossana M. C. Andrade[§,b]

[*]Federal Education, Science and Technology Institute of Ceará
Fortaleza, Brazil
fernandoparente@ifce.edu.br

[§]Group of Networks, Software Engineering, and Systems
Department of Computer Science
Federal University of Ceará
Fortaleza, Brazil
neuman@ufc.br, rossana@ufc.br

I. INTRODUCTION

The miniaturization of electronic components and the evolution of wireless communication technologies have stimulated the development and use of Wireless Sensor Networks (WSN) in various applications, such as environmental monitoring, seismic detection, military surveillance, inventory tracking and smart spaces, among others. In general, WSNs consist of battery operated small size sensors, which use wireless short range communication. Furthermore, these networks have severe energy consumption constraints, processing power, memory and bandwidth [1].

Monitoring is important for debugging and analyzing the operation of a WSN. By using a monitoring system, a lot of information about the operation of the network can be obtained, such as topology discovery, nodes death and restart, isolated nodes, routing loops, packet loss, and network latency, among others.

In a WSN, network monitoring can be divided into active monitoring and passive monitoring. In active monitoring, code lines are inserted in the application running in sensor nodes to obtain information about the operation of the network. In this case, the monitoring packets are sent along with data packets of the network modifying the behavior and operation of the monitored network and consuming the resources of this network.

In passive monitoring, a monitor network needs to be deployed in addition to the network to be monitored, named *target network*. The monitor network captures and analyzes packets transmitted by the target network, not consuming any resources of the target network. So, when there is a need to reduce the use of target network resources, it is better to use a passive monitoring system.

In this paper, initially, we identify, analyze and compare the main passive monitoring systems proposed for WSN. During our research, we did not identify any passive monitoring system for WSN that aims to minimize the use of monitor network resources. Therefore, we propose an architecture for energy-efficient passive monitoring systems for WSN. The main goal of this architecture is to reduce energy consumption and extend the lifetime of the monitor network.

[a] Sponsored by CNPq under Productivity Scholarship D-1.
[b] Sponsored by CNPq under Productivity Scholarship DT-2.

978-3-901882-46-3

The lifetime of a WSN can be up to several years and not all problems arise during the first weeks after deployment of the network [2]. So, an energy-efficient monitoring system is important when there is a need of continuously monitoring a WSN in a real scenario (*in situ*), otherwise, the monitor network could have a much shorter lifetime than the target network. In [3], the authors describe the use of a WSN for monitoring the oceans and emphasize the importance of monitoring this network using an energy-efficient passive monitoring system.

II. PASSIVE MONITORING IN WSN

After an extensive literature review we identified five main passive monitoring systems proposed specifically for WSNs: SNTS [4], SNIF [5], Pimoto [6], LiveNet [7], and PMSW [8]. In this review, we performed a comparative analysis among these systems by analyzing the following characteristics:

- *Inference* - capacity of the system to recover packets not captured by the monitor network based on the captured packets;
- *Analysis mode* – it informs the analysis of captured packets to be performed *online* or *offline*;
- *Captured packets* - types of packets (data and/or control) captured by the monitor network;
- *Energy-efficient* – it verifies that the system is concerned to minimize the energy consumption of nodes of the monitor network;
- *Synchronization mechanism* – it describes the mechanism used by the system to insert the timestamp in captured packets;
- *Event analysis* – it informs if the monitoring system analyzes the captured packets to obtain information on the operation of the target network;
- *Visualization tool* – any kind of computational tool used to display information obtained from monitoring.

Table I and Table II show the comparative analysis of these monitoring systems. The main considerations obtained by observing these tables are:

- None of these systems is energy-efficient;
- Only PMSW has the capacity to infer packets not captured;
- None of these systems implements synchronization mechanisms in the sniffers (nodes of the monitor network). LiveNet synchronizes sniffers only on

network deployment. SNTS, Pimoto, and PMSW use a strategy to adjust the clock of the sniffers, but this strategy is not as precise as the synchronization of sniffers;

- SNTS, SNIF, and PWSW analyze the captured data in order to detect failure events or performance of the target network, while Pimoto and Livenet only display the traces of captured packets; and
- Only Pimoto displays the monitoring information in a network management tool used by the community (i.e., wireshark).

TABLE I. CHARACTERISTICS OF MONITORING SYSTEMS – PART 1

System	Inference	Analysis mode	Captured packets	Energy efficient
SNTS	No	Offline	Data + Control	No
SNIF	No	Online	Data + Control	No
Pimoto	No	Online	Data + Control	No
LiveNet	No	Offline	Data	No
PMSW	Yes	Online	Data + ACK	No

TABLE II. CHARACTERISTICS OF MONITORING SYSTEMS – PART 2

System	Synchronization mechanism	Event analysis	Visualization tool
SNTS	Clock adjustment	Yes	Implemented by the authors
SNIF	No	Yes	Implemented by the authors
Pimoto	Clock adjustment	No	Wireshark
LiveNet	Sniffers (deploy)	No	Implemented by the authors
PMSW	Clock adjustment	Yes	Implemented by the authors

III. THE PROPOSED MONITORING SYSTEM

From the considerations described in section II, we propose an architecture for an energy-efficient passive monitoring system for WSNs that reduces the energy consumption of the monitor network. This proposed system implements a synchronization mechanism in the sniffers and provides monitoring information using a SNMP (*Simple Network Management Protocol*) agent.

As mentioned in section I, an energy-efficient passive monitoring system is important when there is a need of continuously monitoring a WSN in a real scenario, otherwise, the monitor network could have a much shorter lifetime than the target network, due to misuse of the energy of sniffers nodes. The synchronization of sniffers is important to ensure the accuracy of the captured packets timestamps, for it allows a more accurate data analysis. By providing the information obtained from monitoring through a SNMP agent, it allows integrating the proposed system with management tools that support the SNMP protocol, such as Nagios and Net-SNMP.

Figure 1 shows the topology of the proposed passive monitoring system, where a monitor network is deployed together with the target network. A monitor network node (*sniffer*) captures in a promiscuous mode the packets sent by one or more nodes of the target network, it inserts a timestamp for each captured packet, it aggregates the headers of several packets in a monitoring message and sends this message, using the monitor network, to the local monitor. The local monitor receives monitoring messages from multiple sniffers, then, it generates a trace file (local trace) with information of captured packets sorted by time and sends the local trace, using an IP network, to the global monitor. The global monitor receives the trace sent by one or more local monitors and it generates a single trace (global trace), which is analyzed to obtain a lot of information about the target network.

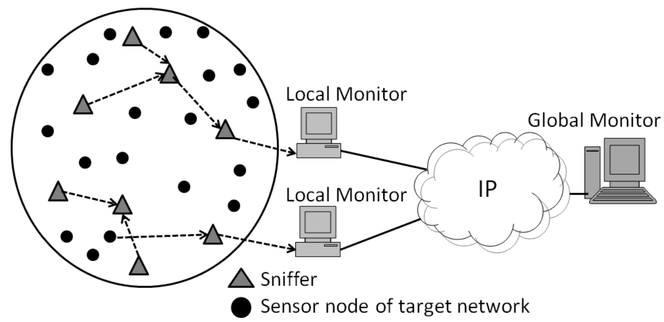

Figure 1. Topology of the monitoring system.

Figure 2 shows the architecture of the proposed monitoring system. In this system, the packets of a given node of the target network are captured by only one sniffer in order to avoid duplicate transmission of packets and, hence, reduce the energy consumption of the monitor network. Therefore, it is necessary to implement a mechanism (**Election Sniffer**) in each sniffer for electing which nodes of the target network will have their packets captured by which sniffers. This election mechanism should be performed by the sniffers using the concept of neighbor semantics discussed in [9], and taking into account some parameters of the monitor network, such as RSSI (received signal strength indicator), battery level, and number of nodes monitored by each sniffer.

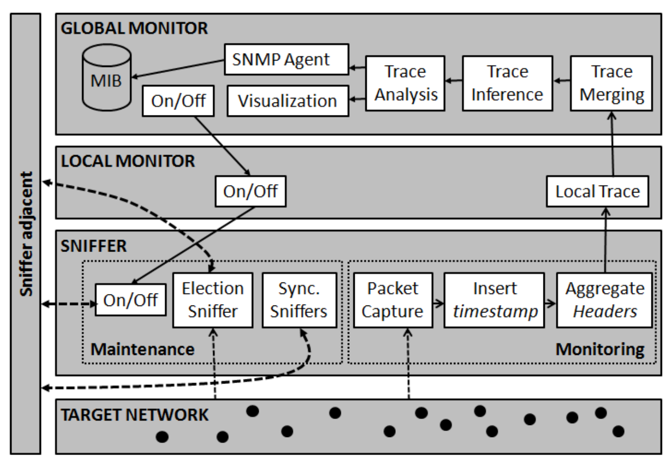

Figure 2. Architecture of the monitoring system.

When capturing a packet of the target network, the sniffer inserts a timestamp in this packet. To ensure the accuracy of the timestamps, it is necessary to use a mechanism for time synchronization of the sniffers (**Sync Sniffers**). The authors of [10] and [11] propose efficient synchronization mechanisms for WSN that can be used in implementing this architecture.

After capturing some packets, the sniffer may aggregate the headers (**Aggregate Headers**) of these packets in a monitoring message to send to the local monitor. The aggregation of headers aims to reduce the amount of data sent through the monitor network, and consequently reduce the energy consumption of this network. In this case, only the information present in the headers of the packets sent by the target network is monitored. However, when there is a need to monitor the data sent by the target network, the system should not use this module aggregation.

The local monitor receives the monitoring messages sent by sniffers, generates a trace file (**local trace**) with information of captured packets sorted by time and sends the local trace to the global monitor. In some scenarios, it is necessary to deploy local monitors in different parts of the network due to the limited range of the radio of sniffers. In a more restricted scenario, only one local monitor can be used and, in this case, the same computer can act as local monitor and global monitor.

The global monitor receives the traces sent by the local monitors and merges these traces (**Tracing Merging**) to generate the global trace. From the global trace, the global monitor uses inference mechanisms (**Trace Inference**) to infer some packets not captured by the sniffers. The authors of [8] propose algorithms to infer packets not captured by sniffers that can be used in implementing this architecture.

After this, the global trace is analyzed (**Trace Analysis**) to obtain information about the target network (topology discovery, nodes death and restart, isolated nodes, routing loops, packet loss, among others). The obtained information is stored in a MIB (*Management Information Base*) by a SNMP agent. Thus, any management tool based on SNMP can display the information obtained from monitoring the target network.

The proposed system also has a mechanism to turn **on** or **off** (**on/off**) the monitoring in order to reduce the energy consumption of sniffers. Thus, the user can program in the global monitor the time intervals in which the target network will be monitored. So, the global monitor sends a command to each local monitor to turn **on** or turn **off** the monitoring, and the local monitors send this command to their sniffers. When receiving the command **off**, the sniffer switches to the *sleep mode*, thereby reducing the energy consumption by up to 95% [12]. Periodically, sniffer switches to the *reception mode* to check if the local monitor sent a command **on**. When sniffer receives a command **on**, it remains in reception mode and restarts the packet capture of the target network.

It is expected to increase the lifetime of the monitor network with the implementation of the proposed architecture in this paper, and thereby to enable the monitoring of WSNs *in situ* during a long period or even during their whole lifetime.

IV. CONCLUSIONS AND FUTURE WORK

In this paper we analyzed the main passive monitoring systems proposed for WSN and we found that: none of these systems is energy-efficient; none of these systems implements synchronization mechanisms in the sniffers; only one system (Pimoto) displays the monitoring information in a free network management tool (wireshark).

Thus, we propose an architecture for an energy-efficient passive monitoring system for WSNs that can reduce the energy consumption of the monitor network, and which implements a synchronization mechanism in the sniffers and provides monitoring information using a SNMP agent.

As future work, we will implement and evaluate the proposed monitoring architecture. This architecture will be implemented using sensor nodes of the platform MicaZ (Crossbow Technology) with TinyOS operating system. The software of the local monitor and global monitor will be developed using the Java programming language.

ACKNOWLEDGMENT

This work is a partial result of the UbiStructure project supported by CNPq (MCT/CNPq 14/2011 - Universal) under grant number 481417/2011-7.

REFERENCES

[1] Borges Neto, J. B., Ribeiro Neto, P. F. and Andrade, R. M. C. (2010) “Wireless Sensor Networks Advances for Ubiquitous Computing” In: Designing Solutions-Based Ubiquitous and Pervasive Computing: New Issues and Trends. Hershey-PA: IGI Global, pp. 175-189.

[2] Hanninen, M., Suhonen, J., Hamalainen, T. D. And Hannikainen, M. (2011) “Practical monitoring and analysis tool for WSN testing” In: IEEE Conference on Design and Architectures for Signal and Image Processing (DASIP), pp. 1-8.

[3] Liu, Y., Liu, K. and Li, M. (2010) “Passive Diagnosis for Wireless Sensor Networks” In: IEEE ACM Transactions On Networking, Vol. 18, No. 4, pp. 1132-1144.

[4] Khan. M. M. H., Luo, L., Huang, C. and Abdelzaher, T. (2007) “SNTS: sensor network troubleshooting suite” In: Proceedings of the 3rd IEEE International Conference on Distributed Computing in Sensor Systems, Springer, Berlin.

[5] Ringwald, M. and Romer, K. (2007) “Deployment of sensor networks: problems and passive Inspection” In: 5th Workshop on Intelligent Solutions in Embedded Systems, IEEE, New York.

[6] Awad, A., Nebel, R., German, R. and Dressler, F. (2008) “On the need for passive monitoring in sensor networks” In: IEEE Euromicro Conf. on Digital System Design Architectures, Methods and Tools.

[7] Chen, B. R., Peterson, G., Mainland, G. and Welsh, M. (2008) “LiveNet: using passive monitoring to reconstruct sensor network dynamics” In: Distributed Computing in Sensor Systems, pp. 79-98.

[8] Xu, X., Wan, J., Zhang, W., Tong, C. and Wu C. (2011) “PMSW: a passive monitoring system in wireless sensor networks” In: International Journal of Network Management, vol. 21, pp. 300-325.

[9] Rocha, A. R., Santos, I. L., Pirmez, L., Delicato, F. C., Gomes, D. G., Souza, J. N. (2010) “Semantic Clustering in Wireless Sensor Networks“ In: 3rd IFIP International Conference on Wireless Communications and Information Technology in Developing Countries (WCITD 2010), IFIP, Brisbane.

[10] Swain, A.R. and Hansdah, R.C. (2010) “An energy efficient and Fault-Tolerant Clock Synchronization Protocol for wireless sensor networks” In: Second International Conference on Communication Systems and Networks (COMSNETS), pp. 1-10.

[11] Tackett, N., Jovanov, E. and Milenkovic, A. (2011) “An implementation of time synchronization in low-power wireless sensor networks” In: IEEE 43rd Southeastern Symposium on System Theory (SSST), pp. 61-66.

[12] Jurdak, R., Ruzzelli, A. G. and O'Hare, G.M.P. (2010) “Radio Sleep Mode Optimization in Wireless Sensor Networks” In: IEEE Transactions on Mobile Computing.