

Mobility in WSNs for Critical Applications

Ricardo Silva\textsuperscript{*} \textsuperscript{1}, Zinon Zinonos\textsuperscript{†} \textsuperscript{2}, Jorge Sa Silva\textsuperscript{*} \textsuperscript{3}, Vasos Vassiliou\textsuperscript{†} \textsuperscript{4}

\textsuperscript{*}Department of Informatics Engineering
University of Coimbra, Pólo II - Pinhal de Marrocos, 3030-290 Coimbra, PORTUGAL
\textsuperscript{1}rnsilva@dei.uc.pt
\textsuperscript{3}sasilva@dei.uc.pt

\textsuperscript{†}Department of Computer Science, University of Cyprus, Nicosia, CYPRUS
\textsuperscript{2}zinonos@cs.ucy.ac.cy
\textsuperscript{4}vasosv@cs.ucy.ac.cy

Abstract—Recent critical application sectors of sensor networks like military, health care, and industry require the use of mobile sensor nodes, something that poses unique challenges in aspects like handoff delay, packet loss, and reliability. In this paper we propose a novel mobility model that handles those challenges effectively by providing on-time mobility detection and handoff triggering. In that way soft handoffs and controlled disconnections are assured. The proposed solution uses cross-layer information from the MAC and Network layers. Our solution was implemented and evaluated in an experimental testbed, in the context of the European FP7 GINSENG project.

I. INTRODUCTION

Although Wireless Sensor Networks (WSNs) are not a novelty in the scientific field, the technology is still not experiencing widespread use in industrial and other mission-critical sectors. The reason for the reluctance of these sectors to adopt WSNs is the difficulty in ensuring reliability. Reliability can be associated with consistency in operation, which in turn can be associated with the ability to execute control functions and propagate the action commands quickly and without loss. Industrial-grade control systems achieve that with proprietary bus systems and wired connectivity. There is, however, a trend to introduce more wireless embedded control sensors in many mission critical systems. These systems are invariably using wireless sensors, which form larger networks and may participate in the control of closed-loop functions. It is, therefore, essential to make sure that performance can be guaranteed, or controlled, in such environments in order to make them wired-equivalent.

One of the attractive characteristics of WSNs is their support for mobility. Transportation, healthcare, industrial, and military applications contain, and sometimes rely on, movement and mobility. Therefore, to perform effective monitoring in such scenarios it is necessary to follow the sensing object. Sensor network mobility is one of the least explored areas of the field (compared to energy consumption and data management issues). In [1] the three possible types of mobility: random, predictable, and controlled are defined. Regarding the mobile entity, there are also different interpretations. Some authors approach mobility in sensor networks by introducing the concept of Mobile Base Stations [1] or Mobile Data Collectors [2]. Others approach the problem by introducing Node Mobility, classifying it as weak if the node moves by the power of thirds or strong if it moves by itself [3]. Node mobility has been also used to increase the network coverage [4].

The approaches described above pertain more to the theoretical use and benefits of mobility. There are numerous applications and clear potential by following these proposals. Unfortunately the practical mechanisms to support mobility in WSNs are not that many. In the state of the art there are also some MAC Layer protocols designed to support mobility, such as MS-MAC [6] and MAMAC [7], which were not implement or evaluated in real platforms. In this paper we present a cross-layer mechanism to efficiently support mobility in a critical scenario. Our solution comes within the scope of the European project GINSENG, which is focused on controlling performance in WSNs. The proposed solution is evaluated and the first practical results of the on-time mobility detection and handoff triggering mechanism are presented.

The paper is organized as follows: Section II presents an overview of the GINSENG project and an explanation of the relevant protocols, algorithms, and scenarios. Section III presents our mechanism to support mobility. Section IV presents the practical evaluation as well as the obtained results and Section V concludes the paper.

II. THE GINSENG PROJECT

The GINSENG Project is a European project whose main objective is to achieve a performance controlled WSN able to provide reliable operation for use in critical scenarios, such as oil refineries.

The immediate target is the Petrogal oil refinery localized in Sines, Portugal, whose operator is one of the GINSENG consortium members. Therefore, there exists a common desire in getting a really efficient and trusted solution, taking advantage of all the benefits that WSNs are capable to bring. Fast and easy deployment, portability, small size and low cost are the most obvious examples. Within the scope of the GINSENG project, three main scenarios were defined. Those scenarios are: the production monitoring scenario, the pipeline leak detection scenario, and the personnel safety scenario, which is the one where mobility is considered and will be examined within the scope of this work.
A. The Mobility Scenario

Within the oil refinery, the pollution is usually high but controlled. However, there exist some situations where employees have to work in some highly hazardous areas. Cleaning empty storage tanks is the most common hazardous activity. To perform this task, the employees must be equipped with special suits and tools to protect them from such environment. This form of protection, even though standard, is not considered adequate anymore and a constant monitoring of the personnel health status is desired. With WSNs this activity can be easily enabled and the employees can be monitored in real time, equipping their suits with sensors to measure a number of vital signals and use the network to transfer them to a monitoring station (health center). The personnel safety scenario aims to implement such real time monitoring. Moving between tanks or other type of hazardous areas can cause two types of mobility: intra-network mobility and inter-network mobility (Figure 1). Independently of the situation, our objective is to maintain the employee on-line, controlling the packet losses and latencies. Within the scope of this project, a specific MAC layer has been developed and an appropriate topology control mechanism has been defined. Therefore, to implement a mobility solution, both of the above must be taken into account.

B. Topology Control and GinMAC

The GinMAC protocol [8] is a TDMA-based protocol and assumes that data is forwarded hop-by-hop towards a sink within a tree topology consisting of $n$ nodes. The time axis is divided into fixed-length base units called epochs. Each epoch $E$ is subdivided into $k * n$ time slots for a network of at most $n$ sensor nodes. Each node is assigned $k$ exclusive slots per epoch $E$. The GINSENG topology can be modeled as a tree as shown in Figure 2. A reasonable small number of nodes can be expected ($N < 30$) of which $N$ is directly proportional to the required communications delay bound; the smaller the required delay, the smaller the $N$. Larger networks can be divided into smaller networks with additional in-field data collection stations (sinks). The maximum number of hops $H$ can be expected to be small ($H < 4$) while most nodes will be within one or two hops from the sink. Each epoch is provisioned to have a number of slots for downstream and upstream communications. The exact number of slots is determined at provisioning time and based on the tree structure. Using upstream communication the destination address is set to the unicast address of the receiver node while using downstream communication the destination address is set to the broadcast address. Dynamic topology control (DTC) uses MAC signaling (control packets) to discover the nodes’ neighbours and to dynamically create a tree-based network in a distributed fashion. Topology control is responsible for advertising empty tree positions, accepting and rejecting prospective children, perform optimizations of the node positions within the tree, and allocating the function of each slot within the next epoch i.e sleep, scan, receive and transmit.

The sink is the first node that will start sending advertisements. The advertisements are broadcast messages that advertise specific child tree position(s). The advertisements are sent in the downstream slot of the epoch. When a node (not the sink) is first switched on it initializes during the first epoch and sets all of its slots to the scan mode so that to receive advertisements. On receipt of an advertisement, the MAC will pass this packet to the topology control module running in each node, which will process the packet and select a tree address to be attached. Then the node will send a Join control packet to the advertiser node asking a confirmation to use the specific tree address. In case that a new node receives more than one advertisements it will select to join in the address with the best RSSI value and closest to the sink node (minimum number of hops from the sink). In case that two different nodes select to join the same tree address there will be a collision and the nodes will back-off and select a tree address again in the next epoch. Upon accepting a Join request, the parent node will create a Join Acknowledgement packet and send it to the child node. A child node receiving the Join Acknowledgement changes its status to attached and starts sending data upstream as well as advertising its children positions (if any). If the node is a leaf node it can not support child positions thus it is not sending advertisement. In order to keep track of the node state each node maintains a list where it keeps information about its children and free position(s). If the node does not receive any advertisements during its initialization phase it sets all the slots in idle mode and wakes up after a predefined time to listen for...
advertisement messages and repeat the above procedure. More
details about the operation of the Dynamic Topology Control
module can be found in [9] where an extended evaluation of
the DTC is presented.

The proposed mobility model, described in the next section,
takes into account the specific features and operation of the
MAC and topology implementations explained above.

III. PROPOSED MOBILITY MODEL

In the GINSENG project, we aim to maintain good links
between tree nodes which means that we need to monitor a
set of parameters in real time such as the Received Signal
Strength Indicator (RSSI) values, the Packet Reception Rate
(PRR) and any transmission errors. In the presence of mobility,
at Layer 2 we implement mechanisms in order to assure soft
handoff (when possible) or controlled disconnections. During
movement the node may be in a state where it is actively
communicating or it might be completely silent (radio off).
In the first case the application allows us to detect the movement
of the node without using any extra features. However, in the
second case, we need to implement additional mechanisms to
make sure that mobility is detected on time and the handoff
disconnection time remains under a predefined maximum
value.

A. Movement detection

To detect the movement we use the RSSI values obtained
from the metadata field of IEEE 802.15.4 protocol. Therefore,
if nodes are silent they can move without being detected and
future communication packets may be lost. To overcome this
problem we aim to use the KEEP_ALIVE and NODE_ALIVE
messages directly from the GINSENG DTC module. The
KEEP_ALIVE message is sent by the parent every $x$ seconds
where $x \in \mathbb{N}$. Consequently, the MN is aware of the time
to receive the KEEP_ALIVE message. If it does not receive
the message on time then it transmits a NODE_ALIVE message
and waits for the acknowledgement. If the acknowledgement
is not received, then the MN sets the mode to scan.

With this solution, we can assure that it is possible to control
the maximum disconnection time during handoff. This dis-
connection time will be at worst equal to two MAC epochs.
However, we can also improve the handoff performance by
following our new handoff trigger method. Let us take for
example the case of a mobile node movement towards a new
connection point, while the node has its radio on. During that
movement, the node receives a KEEP_ALIVE message from
the prospective parent. Looking at Figure 3, we can recognize
that when we receive this first KEEP_ALIVE message our
current RSSI value was in the critical zone. The critical zone,
as defined in [10], is the interval between a threshold value
we set to recognize the deterioration of our communication
link (Threshold) and the point where the link is no longer
usable (Rupture point). In [10] this zone is obtained based
on the nodes’ direction, velocity and noise. However, for
GINSENG we follow a different approach, where the critical
zone becomes a static point based on the desired performance.
Once the current RSSI is within the critical zone, nodes are
automatically entering the scan mode searching for a new
attachment point (parent). Topology control will be responsible
to find a new attachment point for the MN, as is described in
Section II-B.

Another possible situation is shown in Figure 4. While re-
ceiving the KEEP_ALIVE message, the RSSI is not in the
critical zone, but with high probability it will enter the
critical zone soon. Since no further message is sent in the
next seconds, the MN will be disconnected and it will not
be possible to detect this, neither to localize it. Therefore,
we included a NODE_ALIVE message to be sent after the
expected KEEP_ALIVE message. For instance, if we set the
KEEP_ALIVE interval to 30 seconds, at the end of the inter-
val, the MN will send a NODE_ALIVE message if no further
messages have been received. After sending NODE_ALIVE,
the MN must receive an acknowledgement. If not, it will enter
the scan mode searching for a new attachment point. The
MN will then switch to scan mode in order to receive an
advertisement of a new position. As explained at the beginning
of this section, topology control will take care of that.

The MN, in order to be able to send any messages, must first
establish a link connection and obtain a tree address. The MN
will acquire a tree address following the procedure outlined
in Section II-B. Upon joining a tree the node will obtain a
16-bit unique tree address that is used for routing the packets
inside the tree. After obtaining the tree position it checks the
frame to identify whether it has moved to a new Personal Area
Network (PAN). If so, Layer 3 mechanisms start to deal with
the registration update process.

The physical location of the node affects the tree position
that the node will acquire. In the mobility aware scenario,
topology control must ensure that the mobile node will always
have available tree positions to be attached. Hence, a mobile
worker will always have the ability to connect to a different
tree position during the intra- or inter-network movement.
Based on GinMAC a node can be attached to more than one
tree positions inside the same tree, with DTC responsible for
deciding which is the best tree address to attach the MN to.

This option is helpful in case we have intra-network mobility where mobility occurs within the same tree. In that case when the RSSI value is inside the critical zone the MN uses its unused slots to scan and receive messages (advertisements) from other nodes. By evaluating the received advertisements, the topology control decides which is the best tree address for the MN to be attached to.

The MN releases the first tree position after it is attached to the second tree position, thus ensuring zero down time during intra-network movement.

The same procedure is followed in the case of inter-network movement but with some assumptions:

- The tree structure is the same; meaning that the networks that may be visited by MNs should have the same predefined tree structure.
- Since individual networks are using different radio frequencies we accept brief disconnections in order to switch the radio.
- Global time synchronization between sinks is expected.

B. S-GinMob

Since the mobile node has to maintain its attachment position to the tree topology, our proposal, named S-GinMob, is specifically designed to operate on top of GinMAC and in conjunction with the DTC implementation. We implemented our mobility protocol in such way to exploit the idle slots of the MN. Following a predefined schedule, the MN will set its idle slots to scan mode in order to receive any advertisements send by neighboring nodes and possibly find better attachment point. When it receives a new advertisement it will compare the received RSSI value with the current parent RSSI values. In case that the new RSSI is better, it will send the join request to the new parent. The better RSSI decision is obtained taking into account the following formula:

\[
\text{RSSI}_{\text{current}} + s < \text{RSSI}_{\text{received}}
\]  

where \( s \) is a constant value obtained by experiments in field.

Upon receiving the Join Ack message from the new parent the MN will switch to the new address. The signaling of the proposed solution is depicted in Figure 5.

The critical zone calculation based on the RSSI threshold, the energy cost of the periodic KEEP_ALIVE messages and the S-GinMob are evaluated in the next section.

IV. EXPERIMENTAL EVALUATION AND PERFORMANCE ANALYSIS

A. Assuring on-time handoffs

Our solution intends to guarantee the reliability of links between MN and parents by monitoring in real time the connection RSSI. Although, RSSI was considered a non-reliable metric in conventional networks, in WSNs it is widely used and scientifically supported [11].

In the presence of mobility, link quality metrics are crucial to control the mobile process and to assure a reliable connectivity across the path. Based on them, MNs or parents must recognize what is a good link and what is not. Furthermore, both must recognize when the handoff must take place and how fast it must be. In our evaluation we aimed to achieve a relation between PRR and RSSI, capable to assure a certain percentage of packet delivery at a certain RSSI threshold.

To achieve such, we run an experimental setup in which two TelosB nodes programmed with ContikiOS were used. One node was configured as the mobile node (MN) and sent 200 packets with an inter-packet interval of 100ms to the Sink Node. At each evaluation we increased the distance 1 meter between the sending and receiving nodes. This test was run in a controlled environment with both nodes in line-of-sight communication which assured us the best possible performance.

Figure 6 depicts the obtained results. The RSSI values are the average of the received packets. As we can observe it is possible to assure a 90% packet delivery until the -80dBm. We can also observe that the number of packet losses was stable until the -70dBm range with one or two losses caused by transient interference. Those losses slightly increased until the -80dBm and started dropping after that. In order to establish a relationship between the desired PRR and the RSSI threshold, the following 6th order polynomial formula, was obtained:

\[
y = p1 \ast z^6 + p2 \ast z^5 + p3 \ast z^4 + p4 \ast z^3 + p5 \ast z^2 + p6 \ast z + p7
\]  

In equation 2 \( z \) is centered and scaled with \( z = (x - \mu)/\sigma \), where \( \mu = -69.258 \) and \( \sigma = 10.898 \).

The norm of residuals is equal to 0.58771 and the obtained coefficients are: \( p1 = -0.0014668 \), \( p2 = 0.012672 \), \( p3 = -0.037544 \), \( p4 = 0.037544 \), \( p5 = -0.003296 \), \( p6 = -0.0046873 \), and \( p7 = 0.99419 \).

With this formula it is possible to determine the best RSSI threshold for the desired PRR, between the given interval, i.e., [-30; -90]. Hence, if for instance a critical application requires a PRR of 0.95 the chosen threshold should be approximately -78 dBm.
of the order of 1.5%, while at the longest period the energy consumption difference is only 7 μJ or 0.2%.

Additionally, we measured how the active duty cycle is affected by sending the KEEP_ALIVE message. The Duty Cycle is calculated by dividing the number of active slots by the number of total slots (see eq. 4).

\[
\text{DutyCycle(\%)} = \frac{(N_{\text{listen}} + N_{\text{transmit}})}{(\text{Total slots} \times 82)}
\] (4)

The duty cycle that the node is on follows the same pattern as the energy consumption. In Table I we observe that the biggest overhead to active duty cycle is observed when the node sends the KEEP_ALIVE message every one second. In that case the node is on for 0.07% more time. As we increase the KEEP_ALIVE interval we observe that the overhead to the active duty cycle is decreasing and tends again to be very small when the 30 seconds interval is used. Bearing in mind that the sensor nodes are attached to mobile workers, the speed of workers is not expected to be high. Thus, the use of a small interval for the KEEP_ALIVE is not necessary. In addition, since the energy overhead for 30 seconds interval in both energy consumption and duty cycle is not considerable (almost zero) we set that interval as a default value of sending the KEEP_ALIVE messages and detecting the workers mobility on time.

C. S-GinMob Evaluation

To evaluate the proposed soft handoff mechanism a number of tests were conducted on a real testbed, deployed inside our lab environment. We used the dynamic topology control module to build a tree topology as depicted in Figure 2. We assume less than 16 nodes since we need to have available positions to support the handoff procedure. The sensor nodes were configured to send their readings to the sink every one second, through the constructed tree topology. The tests were mainly performed to evaluate the downtime and packet losses during the handoff procedure. The MN was performing a random walk between two nodes, node 0 and node 1 (as shown in Figure 2). During this walk, we monitored the data that the MN was sending to the sink node. In addition, we monitor the slot numbers where the mobile node was disconnected and connected again to the tree. The received
sequence number and the handoff events are shown in Figure 8. Based on the results, the disconnection time and the packet losses during the handoff procedure are equal to zero. Figure 8 also shows, in zoom, the packet sequence numbers during the two handoff events where it is clearly indicated that there is no packet loss during the handoff events.

V. Conclusion

In this paper, we proposed and evaluated a mobility model that assures soft handoffs and controlled disconnections. The model is based on the evaluation and prediction of the RSSI parameter, the use of two new messages, the KEEP_ALIVE and NODE_ALIVE, and the reception of new address before the disconnection time. We used experimental testbed evaluations to discover if the RSSI is the right metric to be used for the handoff decisions in mobility aware scenarios, and following a confirmation, we have found the method to define the specific RSSI value to be used. In addition, the energy overhead for the new DTC messages was evaluated. From our results, we concluded that the overhead of using topology control messages to assist in mobility operations is not considerable and that the effect to the sensor node power consumption is small. In addition, we managed to support soft handoffs with zero packet losses and zero downtime. In the future, we aim to test our model inside the Sines refinery environment using both intra- and inter-network mobility cases.

ACKNOWLEDGMENTS

The work presented in this paper was partially financed by the ICT FP7 0384239 GINSENG - Performance Control in Wireless Sensor Networks project of the EU.

REFERENCES


### TABLE I

<table>
<thead>
<tr>
<th>Interval</th>
<th>Tx ticks</th>
<th>Rx ticks</th>
<th>Total ticks</th>
<th>Duty cycle (%)</th>
<th>Energy (% of consumption)</th>
<th>Energy(mJ)</th>
<th>Duty Cycle (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>720</td>
<td>47285</td>
<td>48005</td>
<td>4.885</td>
<td>2.927</td>
<td>Energy</td>
<td>0.044</td>
</tr>
<tr>
<td>1</td>
<td>960</td>
<td>47779</td>
<td>48739</td>
<td>4.947</td>
<td>2.971</td>
<td>1.500</td>
<td>0.032</td>
</tr>
<tr>
<td>5</td>
<td>768</td>
<td>47555</td>
<td>48323</td>
<td>4.905</td>
<td>2.947</td>
<td>0.657</td>
<td>0.019</td>
</tr>
<tr>
<td>10</td>
<td>744</td>
<td>47491</td>
<td>48235</td>
<td>4.896</td>
<td>2.941</td>
<td>0.477</td>
<td>0.014</td>
</tr>
<tr>
<td>15</td>
<td>736</td>
<td>47441</td>
<td>48181</td>
<td>4.891</td>
<td>2.938</td>
<td>0.365</td>
<td>0.011</td>
</tr>
<tr>
<td>20</td>
<td>732</td>
<td>47440</td>
<td>48172</td>
<td>4.890</td>
<td>2.938</td>
<td>0.247</td>
<td>0.007</td>
</tr>
<tr>
<td>25</td>
<td>728</td>
<td>47396</td>
<td>48124</td>
<td>4.885</td>
<td>2.935</td>
<td>0.222</td>
<td>0.007</td>
</tr>
<tr>
<td>30</td>
<td>728</td>
<td>47384</td>
<td>48112</td>
<td>4.884</td>
<td>2.934</td>
<td>0.209</td>
<td>0.007</td>
</tr>
</tbody>
</table>

Fig. 8. Soft Handoff Sequence number.