ZigBee-based Intra-car Wireless Sensor Network

Hsin-Mu Tsai¹, Cem Saraydar², Timothy Talty², Michael Ames², Andrew Macdonald², and Ozan K. Tonguz¹
¹ Carnegie Mellon University, ECE Department, Pittsburgh, PA 15213-3890, USA
² General Motors Corporation, ECI Lab, Research and Development, Warren, MI 48092-2031, USA
Email: {hsinmut, tonguz}@ece.cmu.edu, {cem.saraydar, timothy.talty, michael.ames, andrew.macdonald}@gm.com

Abstract—Due to an increasing number of sensors deployed in cars, there is a growing interest in implementing a wireless sensor network within a car. In this paper, we report the results of packet transmission experiments using ZigBee sensor nodes within a car under various scenarios. The results of the experiments suggest that both Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI) can only be used as a threshold-based indicator to evaluate the link quality - indicating poor link quality when dropping below a certain threshold. Preliminary experimental results show that a detection algorithm developed by the authors based on RSSI/LQI/error patterns and an adaptive strategy might increase the goodput performance of the link while improving power consumption of the radio.

I. INTRODUCTION

Wireless sensor networks have been implemented in various monitoring applications such as industrial, health, environmental, security, etc. Recently, vehicular applications have entered the list of applications, mainly through tire pressure monitoring systems. More widespread use of wireless sensors in a vehicle will result from one or more of several different factors including difficulty with wired sensing and cost reduction opportunity. Fueled by the emerging interest in the industry for deploying a higher number of wireless sensors, there is a need to understand and characterize the wireless channel within a vehicle. To this end, we report a case study using wireless sensor nodes that are compliant with ZigBee. ZigBee is an industry alliance that promotes a set of rules which builds on top of the IEEE 802.15.4 standards [1]. Channel behavior under various scenarios is observed for ZigBee nodes placed throughout a midsize sedan. To the best of our knowledge this paper presents the first attempt to characterize ZigBee performance within a vehicle environment.

The rest of the paper is organized as follows. In section II, the details of the experimental setup are described. The results of the experiments and the discussions of these results are presented in section III. In section IV, we propose a set of detection algorithms and an adaptive strategy that can adjust to channel conditions for improving the error performance of the wireless channel and preliminary evaluation results are presented. Finally, the concluding remarks are given in section V.

II. EXPERIMENTAL METHOD

A. Sensor node hardware

In our experiments, we use Crossbow MPR2400 [2] as our sensor node hardware platform. The specifications are shown in Table I.

B. Experimental setup and sensor node firmware

Figure 1 shows the experimental setup. In the experiment, sensor nodes (SN) are placed in different locations in the vehicle. The base station (BS) is placed inside the instrument panel of the vehicle, next to the vent. The base station is connected to a MIB510 programmer and a RS-232 to USB converter is used to connect the laptop and MIB510.

The sensor nodes periodically retrieve sensor information from the attached sensors and send (broadcast) sensor packets to the base station. The base station acts as a bridging device between the sensor nodes and the laptop, relaying the sensor packets from sensor nodes to the laptop and the command packets from the laptop to sensor nodes, as well as logging various metrics such as Received Signal Strength Indicator (RSSI), Link Quality Indicator (LQI), CRC, etc., and appending them to each received packet. The packet logger/parser software in the laptop processes the packets sent by the base station, and saves them to a log file for further analysis. The command sender in the laptop can be used to issue commands to adjust the parameters of sensor nodes such as transmitting power, packet sending rate, etc.

The firmware of sensor nodes and the base station is based on TinyOS 1.1.15 [3]. TinyOS is an open source component-based operating system and platform targeting wireless sensor networks. Our implementations use various API and libraries provided by TinyOS.
TABLE I
CROSSBOW MPR2400 (MICAZ) SPECIFICATION

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value/Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>ATmega 128L Processor</td>
</tr>
<tr>
<td>Radio Chip</td>
<td>Chipcon CC2420 Radio</td>
</tr>
<tr>
<td>Operating Frequency</td>
<td>2.4 GHz</td>
</tr>
<tr>
<td>Effective Data Rate</td>
<td>250 Kbps</td>
</tr>
<tr>
<td>Modulation Format</td>
<td>Offset Quadrature Phase-Shift keying (OQPSK)</td>
</tr>
</tbody>
</table>

TABLE II
SENSOR NODE LOCATIONS IN THE CAR

<table>
<thead>
<tr>
<th>Node No.</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Embedded in the instrument panel, next to the vent</td>
</tr>
<tr>
<td>6</td>
<td>On the dashboard, next to the light sensor</td>
</tr>
<tr>
<td>7</td>
<td>On the right side of the trunk, next to the stability actuator</td>
</tr>
<tr>
<td>1</td>
<td>In the engine compartment, next to the fuse box</td>
</tr>
<tr>
<td>0</td>
<td>In front of the radiator, between the temperature sensor and the air quality sensor</td>
</tr>
</tbody>
</table>

C. Sensor node locations

The vehicle used in the experiment is a General Motors 2005 Cadillac STS. Figure 2 and Table II show the locations of the sensor nodes as well as the base station in the vehicle.

D. Experimental scenarios

We performed different experiments under various scenarios shown in Table III. The details of these scenarios are discussed in the following.

1) Location
   a) Maintenance garage This is similar to the service depot of a regular car dealer. Technicians walk by frequently and several other cars are parked nearby. There are a lot of service equipments in the garage.
   b) Corporate parking lot This is a regular corporate parking lot. The test vehicle was parked in one of the parking space and surrounded by other cars. Pedestrians passed by the test vehicle sometimes.

2) Driver In “driver present” scenarios, the driver was sitting in the car and has frequent movements, such as operating A/C, radio, steering wheel, etc. In “driver not present” scenarios, the driver’s seat was empty.

3) Engine In “engine on” scenarios, the engine was started and kept running throughout the whole measurement. The air conditioner and the radio in the vehicle were also turned on. In “engine off” scenarios, the engine was turned off (not in the accessory mode) and the key was removed from the vehicle.

E. Communication parameters

- Transmitting power: In the experiment, we set the transmitting power of the sensor nodes to be at 5 different levels: 0, -5, -10, -15, and -25 dBm. The transmitting power of the base station was fixed at 0 dBm (The base station only transmits when sending commands to sensor nodes).
- Packet sending rate: We configured the sensor node to send a sensor packet every 100 ms. This sending rate is sufficient for non-safety sensors in the vehicle.
- Channel selection: The physical layer standard of MICAz nodes follows IEEE 802.15.4 standard [4]. Since 802.11b/g devices are the most common devices in 2.4 GHz ISM band and are likely to create a lot of interference, we select a channel that is away from the bandwidth occupied by 802.11b/g standard. The bandwidth used by 802.11b/g and 802.15.4 devices are 3 MHz and 22 MHz, respectively. In our experiment, we configured the sensor nodes to use channel 26 (2480 MHz) to avoid interference from 802.11b/g devices. In this case, the closest 802.11b/g channel is channel 11 (2462 MHz) and does not overlap with our 802.15.4 channel.
- Packet format: Figure 3 shows the sensor packet format used in the experiment. The total size of MAC Protocol Data Unit (MPDU) plus the frame length field is 31 bytes. Note that most of the fields in the application level payload are used to record the information for the experiments. For example, we used 12 bytes to record the sensor information, as well as 1 byte each to record transmitting power and version number of the firmware. The size of the sensor packet can be reduced by removing

Nearby vehicles sometimes moved in or out of the parking lot.

c) Road This is the driving scenario. The car was driven on the highway most of the time and sometimes on large (multiple-lane) local roads.

Fig. 2. Sensor node locations in the car. The numbers in the circles are showing the number of the sensor node. See Table II for descriptions.
unnecessary fields and results in a lower packet error rate. Depending on applications, the size of the sensor information field can also be reduced.

- **Node transmissions**: In the experiments conducted, only one of the sensor nodes transmitted at a time. We used this setting to avoid interference from other sensor nodes and focus on measuring the link quality.

- **MAC related parameters**: In the experiment, we disabled the automatic ACK feature, as well as the retransmissions. The sensor nodes use a MAC protocol similar to the Carrier Sense Multiple Access (CSMA) used in 802.11b/g, in which it will wait until the channel is clear (perform clear channel assessment) and then start transmitting.

- **Data collection**: For each scenario/transmitting power/sensor node, we configured the sensor node to transmit 6000 sensor packets, which took 10 minutes. The total time to complete the data collecting process for each scenario was around 200 minutes.

### F. Observable entities

The following describes various observable entities recorded by the base station.

- **Link Quality Indicator (LQI)** LQI is calculated by Chipcon CC2420 radio chip and is actually Chip Correlation Indicator (CCI). It is related to the chip error rate. LQI ranges from 50 to 110 and is calculated over 8 bits following the start frame delimiter.

- **Received Signal Strength Indicator (RSSI)** RSSI is measured by Chipcon CC2420 radio chip and represents the amount of energy received by the sensor node. According to [5], RSSI has a range from -100 dBm to 0 dBm and the maximum error (accuracy) is 6 dB. The RSSI is calculated over 8 symbol periods.

- **Sequence Number** In the sensor data packet, there is an application-level sequence number field which will be increased each time the sensor node sends out a packet. This can be used by the base station to detect a lost packet.

- **Cyclic Redundancy Check (CRC) field** Chipcon CC2420 radio chip has automatic CRC checking capability and TinyOS has a CRC field in its radio packet indicating whether the packet received pass the CRC checking. The CRC scheme used in CC2420 is CRC-16 (ITU-T).

### G. Definitions of metrics

In this sub-section, we define the metrics used later. First we define the following variables:

- **G**: The number of packets received by the base station and passed the CRC check.
- **LE**: The number of packets received by the base station and either the length of the packet or the type of the packet (indicated by the type field) was not correct.
- **CE**: The number of packets received by the base station and failed the CRC check.
- **A**: The total number of packets transmitted

Note that our packet parser will first detect length/type errors. If the length/type of the packet is not correct, it will be put into LE category. The CRC field of these packets might indicate that it is in error, but these will not be included in CE.

Now we define the following error-related performance metrics using the above variables:

- **Packet Reception Rate (PRR):**
  \[
  PRR = \frac{G + LE + CE}{A} \tag{1}
  \]

- **Packet Error Rate (PER):**
  \[
  PER = \frac{LE + CE}{G + LE + CE} \tag{2}
  \]

- **Goodput:**
  \[
  Goodput = \frac{G}{A} \tag{3}
  \]

### H. Experiment for understanding the impact of Bluetooth

To study how the existence of an interference source can impact the performance of the ZigBee sensor nodes, we used the integrated Bluetooth hands-free in the Cadillac and a Motorola RAZR V3 cell phone to create interference. We performed the experiment with and without the Bluetooth interference in scenario no. 3 in Table III (with limited Bluetooth data set; each node transmitted using only one transmitting power setting). In the experiment with Bluetooth interference, the cell phone was used to place a phone call and maintain a Bluetooth connection with the hands-free during the whole experiment period.

The Bluetooth protocol uses a Frequency Hopping Spread Spectrum (FHSS) mechanism. It hops to one of the available channels every 0.625 ms according to a hopping sequence specified by the master node. The Bluetooth standard used in U.S. has 79 1-MHz-wide channels spread from 2402 MHz to 2480 MHz. Hence, the last two channels will overlap with the 802.15.4 channel (2479 MHz) used in our experiment and will create interference to the sensor nodes.
channel loss packets, the average channel loss is higher than the expected value. Observe from Figure 4 that insufficient amount of data and statistical effects result in some variations of the channel loss curves, as opposed to perfectly flat lines.

B. Error metrics and RSSI profiles

The receive sensitivity of the radio chip in the sensor nodes is -95 dBm (typical) and -90 dBm (minimum), as specified by [5]. The sensitivity corresponds to the minimum received signal strength beyond which the packet error rate exceeds 1%, as defined in [4].

To study the relation between various error metrics and RSSI, we computed the plots as follows. Each 6000-packet sequence of one setting was split to segments of 50 consecutive packets. For each of the segments, error metrics were calculated over these 50 packets, which is represented by y. The mean of the RSSIs of these 50 packets were also calculated and represented by \( \mu_x \). Then we plot the point with coordinate \((\mu_x, y)\) on the figure to represent this segment, and repeat this procedure with all segments in this setting, and all the data of other settings. Figure 5(a)-(e) shows the profiles of PRR versus RSSI for each of the scenarios. Figure 6 and Figure 7 show the profiles of 1-PER and Goodput versus RSSI, respectively.

In Figure 5, one can observe that, in agreement with the specified receive sensitivity, the PRR drops from 1 to 0 within the range -91 to -94 dBm. The outliers that violate this general observation are due to external effects such as driver movement within the cabin, interference from other wireless devices, etc. For instance, 802.11b/g access points are deployed in the maintenance garage in scenario no. 1 and 2, which is configured to operate on channel 11, which is rather close to the frequency band the wireless nodes operate at. As a result of such effects, the receive sensitivity boundary experiences slight shifts to the right of the figure, representing a less “friendly” propagation environment. One can also observe that the trunk data manifests a higher level of fluctuations (more outliers), possibly due to the rich multipath environment caused by the presence and the motion of the passenger along the direct path that lies between the sensor node and the base station node.

In Figure 6, we observe that in scenarios where the engine is on or the driver is present, there is a higher noise level which leads to poorer PER performance. We also observe that there are fewer outliers as compared to the results in Figure 5, which could be explained by a relatively lower impact of the driver or the engine noise on the correlation between PER and RSSI. In Figure 7, one can also observe that the goodput performance is good only when RSSI is much larger than the received sensitivity boundary.

C. Error metrics and LQI profiles

In Figure 8, we calculate the mean \((\mu_x)\) and the standard deviation \((\sigma_x)\) of the LQIs of the 50 packets in a segment. Then we plot the points \((\mu_x - \sigma_x, y)\) and \((\mu_x + \sigma_x, y)\), and connect these two points with a line to represent this segment and repeat the procedure for all other segments.
As shown in Figure 8, one can observe that the variance of each segment (50 packets, which is received in a period of 5 seconds) in these plots is too large, and hence we conclude that it is difficult to estimate link quality based on the short-term averages that our data points represent [6]. It should be noted that we observe a relatively high correlation between LQI and Goodput. Based on these observations, we can develop a rule of thumb which says that the Goodput could be used as an upper bound: If LQI is smaller than a certain value; the Goodput cannot be higher than a certain value. Closer look at Figure 8 shows an example of such a curve (black dashed curve) which could be used as this bounding function.

D. Bluetooth’s impact

Figure 9 compares the goodput performance with and without the Bluetooth interference. As expected, Bluetooth interference has a big impact on the goodput performance to all nodes and the goodputs decrease by 3% - 40%, depending on the individual node and received power. One can also observe that the impact on goodput is larger on nodes which have poorer channel quality than others.

IV. DETECTION ALGORITHM AND ADAPTIVE STRATEGY

Existing studies in the open literature of wireless sensor network usually concentrate on how to use various observable entities such as RSSI to evaluate the link quality and choose one of the available routes or links based on the link quality evaluation. In our experiment, we assume that star topology is used for the wireless sensor network in cars - each node has only one available route/link to the base station. Hence, instead of choosing a better route/link, the sensor node need
to improve the goodput performance of the link.

A. Detection algorithm

Based on the experimental results, we identified 3 different problems which would result in low goodput performance of the link:

- Fading (“Long-term” problem): e.g., passenger causing channel fading
- Interference (“Short-term” problem): e.g., frequency hopping interference
- Low received signal strength

One can also observe that these 3 problems have different RSSI/LQI/Error patterns (See Figure 10, 11, and 12):

- Fading
  - Deep dropping of RSSI/LQI/Error for a long period
  - Consecutive low LQI points during fading
- Interference
  - RSSI outliers (mostly bigger RSSI samples)
  - Random RSSI/LQI/Error outliers
- Low received signal strength
  - Low RSSI
We developed a set of detection algorithms to identify and detect these 3 problems in real-time, based on the patterns of RSSI, LQI, and error indicator inputs. The experimental data are used to fine-tune various parameters in the algorithm. The algorithms are not shown in this paper due to lack of space.

B. Adaptive strategy

We choose to use detector-in-base-station model (as opposed to detector-in-sensor-node model, both shown in Figure 13). In this model, the detection algorithms are executed at the base station side. When detecting either one of the 3 problems, the base station will send out a command to the corresponding sensor node. Based on the problem detected, the sensor node will take the actions described as follows:

- **Fading** Increase transmitting power and observe if the additional amount of power can overcome the increased channel loss due to fading.
- **Interference** Use a more powerful transmission scheme (e.g. repetition code, retransmission, etc.).
- **Low Received Signal Strength** Increase transmitting power and bring RSSI away from the received sensitivity boundary (-90 dBm)

The detector-in-base-station model has many advantages. The detection algorithms do not consume additional computational power on the sensor nodes, and do not require additional control message to detect the problems. The detection is more accurate than the detector-in-sensor-node model as it does not require the assumption of symmetric wireless channel.

C. Preliminary results

We implemented the fading and low power detection algorithms and the adaptive strategy described in the previous subsections and performed experiments with two fixed transmitting power settings and another one with the adaptive strategy. We performed the experiment with only node 7 (trunk) since it is the most likely node to experience fading.

Table IV shows the experimental results. The results suggest that using a simple strategy might be sufficient to increase the link quality while not consuming too much radio transmitting power.

V. Conclusions

In this paper, we report the use of ZigBee sensor nodes to perform packet transmission experiments in a car environment. Results suggest that the change in link quality with respect to the locations of the nodes in the car is significant. Engine noise can increase the PRR/PER/Goodput received sensitivity threshold by 2 to 4 dB. Bluetooth interference can decrease the goodput performance by 3% to 40%. RSSI and LQI can be used to evaluate the link quality by indicating poor link quality when dropping below a certain threshold. Preliminary results of the implementation show that the detection algorithms we have designed based on different RSSI/LQI/error patterns and an adaptive strategy might improve the goodput performance of the link while optimizing the transmitting power of the sensor node’s radio.

REFERENCES