

USING HIGH-POWERED LONG-RANGE ZIGBEE DEVICES FOR COMMUNICATION DURING AMATEUR CAR RACING EVENTS

James Wakemen, Matthew Hodson, Philip Shafer, and Vasil Hnatyshin
Department of Computer Science
Rowan University
Glassboro, NJ 08028

Abstract—In the world of amateur motorsports the racers are always looking to improve their skills. This has led to the development of various monitoring and data logging systems, often coupled with "action sports" cameras which capture the car performance in enough detail to allow the driver to study the race after completion. In this paper we investigate the possibility of the using data logging systems together with IEEE 802.15.4 high power devices for race car – pit crew communication during the race. This will allow the pit crew to monitor the car and get the driver to stop before a catastrophic failure occurs. We used OPNET IT Guru ver. 17.0 network software package to conduct our simulation study. Specifically, we focus on such aspects of 802.15.4 protocol as communication range, reliable data delivery, achievable throughput, and end-to-end delay experiences by the application during the car race.

Keywords—802.15.4; ZigBee; XBee; race car communication; vehicular communication;

I. INTRODUCTION

In the world of amateur motorsports the racers are always looking to improve their skills. This has led to the development of various monitoring and data logging systems, often coupled with "action sports" cameras which capture the car performance in enough detail to allow the driver to study the race after completion.

Professional motorsports teams use sophisticated monitoring systems to tune and monitor the car during a race. These systems are usually sold as complete packages that are not available as generic "off the shelf" systems. The manufacturers of these systems do not typically open their system specifications to the public. The data used by the expensive monitoring systems and that which is logged by the data loggers is similar. An amateur race team would be able to advise the driver during the race to be a better driver or perhaps bring the car off the track and back to the pit crew for repairs during the race if they could have access to the data contained in the data logging system while the car is actively racing.

Recently, inexpensive amateur endurance racing became very popular. Such racing events are governed by various organizations that often mandate strict spending limits on the racecar improvements. The cars are limited to what can be reasonably built on a very tight budget, which often leads to some of these cars becoming very unreliable. Closely monitoring the performance of the engine, transmission, and other components is the key to preventing catastrophic failures during the race. That is why many teams install a large numbers of gauges and lights which allow the driver to observe

the status of the main car components during the race. Bringing the car in for repairs well before failures occur is in best interest of the entire team.

However, during the race driver may not closely monitor the car performance which is why the teams often search for inexpensive methods to monitor the race car remotely. This allows the team members in the pit or paddock area to recognize that the car is beginning to fail and signal the driver via radio or pit board to bring the car in. The radio and pit board signals are harder for the driver to ignore. This should lead to a better outcome for the entire team.

The overall goal of this project is to develop a system for monitoring the performance of a small number of cars during an endurance race and providing this information to the team's pit crew. The main requirements for such a system are reliability, security, low cost, communication in a range of up to 2 miles, ability support for communication between very fast moving nodes, and most importantly easy and quick deployment in the environment without existing communication infrastructure. Typically, the car status information can be encoded within few bytes of data. Therefore, channel throughput is not of the highest importance. Although in the race were the pit crew team needs to remotely monitor several of its cars, the channel throughput may become an issue. Low energy consumption by communication device is preferable but not highly important since the device could be connected to the car's power source.

The most obvious solution for communication problem is to use IEEE 802.11a/b/g/n (Wi-Fi) hardware together with the data logging software both of which are readily available on most smart phone devices. Wide-spread availability of such devices and easy of deployment makes this approach very attractive. However, anecdotal evidence provided by the endurance racers who have attempted to communicate using Wi-Fi enabled smart phones suggest that this approach does not work due to very short communication range of 802.11 standard (i.e., up-to 250 meters), which is not nearly enough to cover most of the racetrack areas.

Next we decided to examine the possibility of using IEEE 802.15.4 (ZigBee) devices for exchanging information between the racecar and the pit crew. Such devices typically are inexpensive, allow for easy deployment, support data encryption, and high power ZigBee modules claim to have line-of-sight communication range of few miles. We were not

able to locate and study any previous research using the higher power ZigBee modules in similar applications.

In this paper we investigate the possibility of the using data logging systems together with IEEE 802.15.4 high power devices for race car – pit crew communication during the race. We used OPNET IT Guru version 17.0 package [4] to conduct our simulation study. Specifically, we focus on the aspects of the 802.15.4 protocol that are most important to the applications such as communication range, reliable data delivery, achievable throughput, and end-to-end delay experienced by the application during the car race.

The rest of the paper is organized as follows. In Section II we provide a brief overview of the 802.15.4 standard followed by description of our preliminary study of communication range in high power 802.15.4 devices in Section III. Sections IV and V introduce the simulation model of the car race we used in our study and provide analysis of the simulation results. The paper concludes in Section VI, which provides summary and plans for future work.

II. OVERVIEW OF 802.15.4 STANDARD

The IEEE 802.15.4 standard [2], commonly referred to as ZigBee, was created for wireless personal area networks (WPAN), which typically consist of several smart devices communicating with one another over short distances in the environment with very little or no existing communication infrastructure. ZigBee devices often are relatively small, portable, inexpensive, easy to deploy, have low power consumption, and support low data rates.

The IEEE 802.15.4 standard was designed to support sensor network applications for control and monitoring of various smart devices within a fairly short range. However, high power ZigBee devices support much longer communication ranges. For example, XBee-Pro® 900HP [1] modules, boast an outdoor line-of-sight range of up-to 28 miles with high-gain antennas and up-to 9 miles with dipole antennas. ZigBee devices typically support data rates of up-to 250 Kbps and operate in license free 915 MHz (US) and 2.4 GHz (worldwide) radio bands [2].

ZigBee devices could be classified into three categories:

- Coordinator nodes that provide synchronization services to other nodes in the personal area network (PAN) and may also serve as a bridge to other PANs. There is only one coordinator per PAN but it is possible to have PANs without a coordinator.
- Router nodes in addition to running application services can forward data to other nodes in the PAN.
- End devices are simple nodes that only support functionality for relaying data to coordinator or a router.

The IEEE 802.15.4 standard refers to coordinator and router nodes as full-function devices (FFD) while end nodes are referred to as reduced function devices (RFD). A ZigBee WPANs can be organized using star or peer-to-peer pattern. In star topology, all communication is conducted via a central node (coordinator), while in peer-to-peer topology nodes can communicate with one another directly. The racetrack communication environment fits well into the star pattern,

shown in Figure 1, where the pit crew serves as a central coordinator node while the racecars are the FFD peers.

The 802.15.4 standard also provides support for such security services (implemented in MAC layer) as data confidentiality, authentication, and replay protection. Additional security features can be added at the application layer as desired.

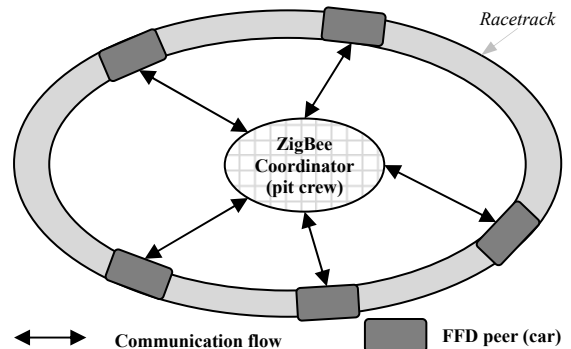


Figure 1: Typical racetrack topology

III. ZIGBEE COMMUNICATION RANGE

During the race we plan to use XBee-Pro® ZB module [3] which is advertised to have the outdoor communication range of 2 miles. The cost of such a device is about \$30, and should be well within the required budget of a typical amateur endurance race. Typically, the race tracks have an oval-like shape of length up-to 3 miles. The pit crew and paddock areas are located with the racetrack oval (Figure 1), which means that the distance between the pit crew and the racecar on the track will not exceed the half of the track length or 1.5 mile. Thus, the communication range of XBee-Pro® ZB module should be long enough to cover a typical racetrack. To verify this fact we conducted a simple simulation study that examines communication range of 802.15.4 devices.

We conducted a simulation study using the OPNET IT Guru ver. 17.0 software package [4]. In our study we placed a ZigBee coordinator and FFD peer next to one another and had the FFD peer move away from the coordinator in a straight line. The FFD peer travels with the constant speed of 20 meters per second while continually sending data to coordinator with the constant rate of one 1024 byte packet every second. We estimated the range of ZigBee device based on the time when the coordinator node stops receiving the data from the peer. XBee-Pro® ZB module's [3] transmit power = 63 mW, receiver sensitivity = -102 dBm, and the data rate = 250 Kbps. In the simulation we set the data rate to 250 Kbps but we varied the transmission power and the receiver sensitivity of the ZigBee devices to get a better picture. Summary of collected results is presented in Table 1. As expected, the simulation results suggest that with the increase in transmit power and receiver sensitivity the communication range of ZigBee devices also increases. For the XBee-Pro® ZB model [3], the OPNET simulation reported communication range of 7120 meters or 4.45 miles, which should entirely cover any typical racetrack.

TABLE 1. SUMMARY OF ZIGBEE DEVICE COMMUNICATION RANGES

Transmit Power (W)	Receiver Sensitivity (dBm)	Communication range (m)
0.1	-85	1700
	-95	5400
	-102	7120
0.063	-85	1340
	-95	4280
	-102	7120
0.05	-85	1200
	-95	3800
	-102	7120
0.01	-85	520
	-95	1700
	-102	3820

IV. SIMULATION MODEL

To obtain an accurate model of the racetrack event we collected the velocity and GPS coordinates of the cars competing in the budget endurance racing event held in New Hampshire Motor Speedway [6], Loudon, NH in October 2012. The event consisted of 14.5 hours of racing divided into two race sessions of 7 hours and 7.5 hours. Our race team fielded two cars for the race. Both cars were equipped with the Android-based Trackmaster [5] data logger software which recorded the location coordinates and traveling velocities of the cars using internal GPS and accelerometers. We exported collected data into RaceChrono [7] video rendering software format. The resulting comma separated text file contains a set of entries, recorded one per line, each consisting of the time and GPS location (i.e., latitude, longitude, and altitude triple) of the car during the race. For our simulation study we selected the data file collected for that car that finished the race in the higher overall position.

We used OPNET IT Guru 17.0 [4] to create a simulation study of the race. OPNET software models node movement using trajectories which are also stored in a comma-separated file. The OPNET’s trajectory file records node movements as a set of entries, each entry consisting of the following information: X position, Y position, Altitude, Traverse Time, Wait Time, Pitch and Yaw. We wrote a java program that converts the RaceChrono data file to OPNET’s trajectory format. The conversion program worked as follows. We recorded OPNET’s X position, Y position, and Altitude in the units of degrees, which allowed us to directly copy the corresponding values of latitude, longitude, and altitude from the RaceChrono data file. We computed the Traverse time as the difference between adjacent timestamp values. While, the Wait Time was set to 0 seconds, and Pitch and Yaw were set to *Autocomputed* value. Since, during a single day race the car travels around the racetrack several hundred times, we converted the RaceChrono data file into 23 different OPNET trajectory files, each consisting of 1200 (time, location) data points.

Each of the simulated race cars had a different trajectory file associated with it. We placed individual cars at various points along the race track, while the Base Station was placed in the paddock area in the middle of the racetrack, approximately the same location as during the race. We used a

satellite image of the New Hampshire Motor Speedway as a background for our simulation scenario. Figure 2 illustrates the network topology used in our OPNET study: the cell phone icons represent individual racecars; a rectangle with the cell tower, located in the middle of the race track, represents the Base Station; while the grey lines around the race track show trajectories of the racecar movement.

The Base Station was configured to act as a ZigBee coordinator, while individual racecars were set-up as mobile ZigBee Routers. Each ZigBee node was configured with transmit power of 63 mW, receiver sensitivity of -102 dBm, and the data rate of 250 Kbps (i.e., 2.4 GHz band). The base station was configured to send periodic status update messages to the race cars, i.e., a 200 bit message was sent to a randomly selected racecar every 15 seconds. The race cars were configured to transmit data log messages to the base station at regular intervals. However, depending on the logging software and the data being collected, the frequency of racecar updates and their sizes could vary. For example, in addition to time, GPS location, and velocity the data loggers could record such information as RPM, throttle position, brake position, oil temperature, oil pressure, coolant temperature, and fuel level, etc with recording frequencies ranging from 1 to 10Hz. We estimated that in the worst case scenario each racecar will transmit logged information at a rate of 12 Kbits per second. We conducted a simulation study to identify the best configuration for transmitting logged data by ZigBee’s application layer. In our study all the racecars were transmitting data at 12Kbps, but in each scenario we changed the size of the data packet and the packet inter-arrival time as shown in Table 2.

TABLE 2. ZIGBEE’S DATA TRANSMISSION SETTING

Packet Size (bits)	Packet Inter-Arrival Time (seconds)	Transmission Rate (bps)
1200	0.1	12,000
6000	0.5	12,000
12000	1.0	12,000

Finally, we also varied the number of racecars on the track. Specifically, we conducted experiments with 1, 2, 5, 10 and 20 cars participating in a race. Please note that in our study, all the cars were traveling around the 1.6 mile racetrack with speeds up-to 90 miles per hour.



Figure 2: Racetrack topology

V. ANALYSIS OF RESULTS

Due to huge amounts of simulation data (~400 MB), the results presented in this section have been averaged to simplify the description and analysis of collected data. In this study we tried to answer the following two questions:

- (1) How many racecars can communicate effectively during the race using ZigBee technology?
- (2) How should the application transmit logged data to maximize the overall performance?

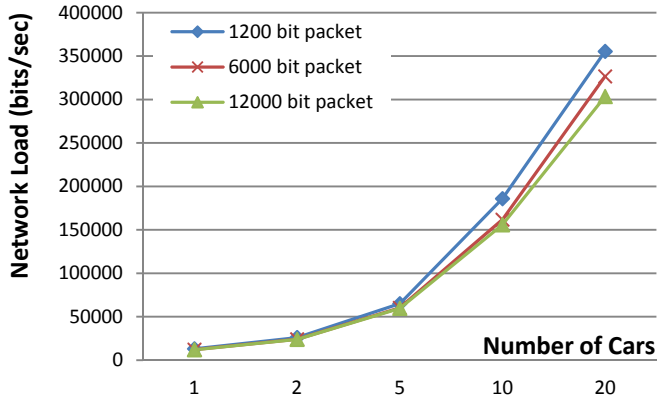


Figure 3: Network Load (bits/sec)

While all the cars transmitted logged data at the same rate in bits per second, the data packet size and the packet generation frequency varied (Table 2). First, we examined the effects of packet size on the network load (Figure 3). As expected, when there were more racecars communicating during the race, the overall network load (the amount of data sent by an application to 802.15.4 MAC) was higher. Also, when the data was sent using small packet sizes the overall network load was higher than when the data was sent using larger packet sizes. This is attributed to the fact that each data packet has lower layer headers associated with it. Since more small packets are needed to transmit the same amount of information, there will be more header information sent into the network, resulting in the higher network load.

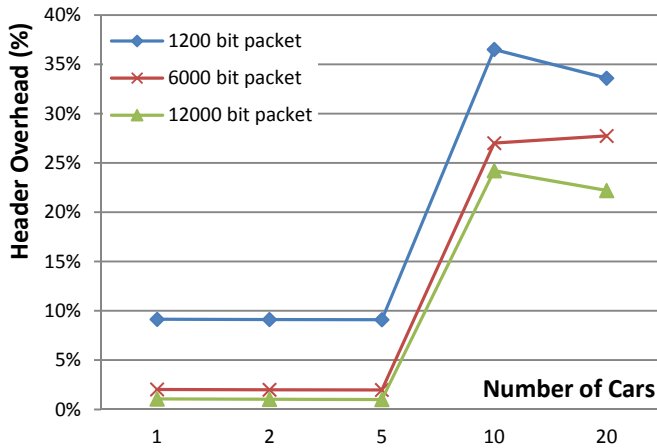


Figure 4: Header overhead (bits/sec)

Next we examined the header overhead generated by the racecar data traffic in our study. Each value was computed by subtracting the amount of application data generated by the racecars from the network load and dividing the result by the network load amount (Figure 4). In the simulated system with

1, 2, and 5 cars, the overhead for 1200 bit, 6000 bit, and 12000 bit packets was constant at 9.1%, 2.0% and 1.0%, respectively. However, with 10 and 20 cars transmitting the overhead increases dramatically: for 1200 bit packets the overhead was 36.5% and 33.6% (~400% increase), for 6000 bit packet the overhead was 27.0% and 27.7% (~1300% increase), while for 12000 bit packets the overhead was 24.2% and 22.2% (~2300% increase). Such drastic change in the amount of the overhead is attributed to the packet loss and subsequent retransmissions which resulted in more data sent into the network.

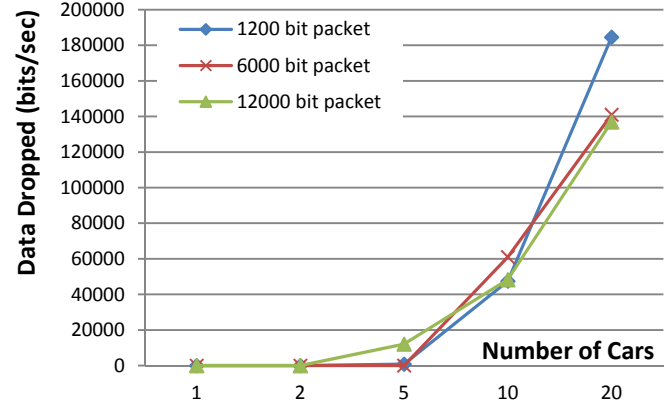


Figure 5: Data Loss (bits/sec)

As shown in Figure 5, the data loss, due to retry threshold exceeded, when only one or two cars are transmitting is close to 0. When there are five cars transmitting the data loss for 1200 bit and 600 bit packets is still close to 0 but the data loss for 12000 bit packets is significant. This phenomenon can be attributed to fact that the large packets occupy the channel for a longer time, which results in more contention between the transmitting cars for available resources. When there are more than five cars in the system, the network is completely overloaded and there is significant data loss.

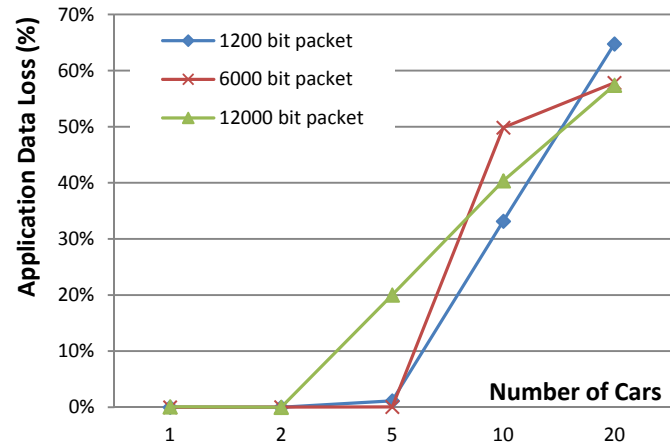


Figure 6: Application Data Loss (%)

Figure 6 illustrates the application-layer data loss in the system. The values in the graph were computed as the difference between the application data sent and received divided by the amount of application data sent. The results presented in this graph are consistent with the results for the data loss at the MAC layer due to exceeding the retransmit threshold: in the system with 1 and 2 cars the data loss is close to 0%; in the system with 5 cars the data loss for 1200 and 600

bit packets is also close to 0% while for 12000 bit packets the data loss is around 20%; in the system with 10 and 20 cars the data loss exceeds acceptable values and ranges between 33 and 65%. Based on these results, we believe that the race team that uses ZigBee devices for communication between the racecars and the Base Station should be limited to up to five cars.

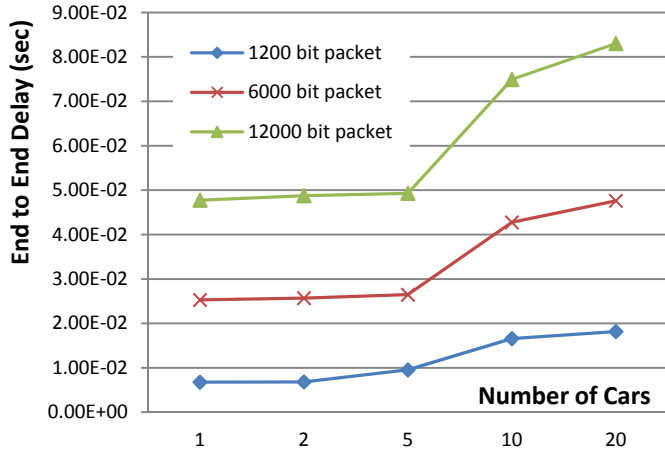


Figure 7: Application end-to-end delay (sec)

Finally, we examined the application-layer end-to-end delay experienced in each of the simulation scenarios. Monitoring the “health” of the racecar system is effective only when the delay between retrieving the car data and delivering it to the pit crew is small. Figure 7 illustrates the summary of results. As expected, the end-to-end delay for communication system that uses small packets is lower than that with the large packet sizes. This is due to the fact that the small packets are processed (i.e., transmitted, sent out of the queue) faster and thus arrive at the destination earlier than the large packets. We also observed that as the number of cars on the racetrack increases, so does the end-to-end delay. This occurs because with more cars there are more packets competing for available system resources, resulting in higher queue occupancy and thus longer queuing delays. However, the overall end-to-end delay never exceed 0.1 second, even in the worst case, which should be more than adequate for delivering logged data from the racecar to the pit crew team.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we investigated the possibility of the using IEEE 802.15.4 high power devices for communication during the race between the racecars and the pit crew. The results collected with OPNET IT Guru version 17.0 [4] network simulation software suggest that while a ZigBee network can support a reasonable amount of telemetry data sent from the racecars to the pit crew, a single ZigBee WPAN is only enough to support a team of up to five racecars. Once there are more than five racecars competing for available ZigBee resources, the network performance degrades to unacceptable levels. While ZigBee’s limit of five cars per team may seem like a big obstacle to our goal of creating a communication system between racecars and a pit crew, in practice this is not a problem. Typically, the amateur endurance racing teams field at most 2-3 cars per team. Only in very rare cases the race team will bring five or more cars to competition.

We also observed that the ZigBee application should not attempt to transmit big data packets infrequently because this

appears to result in high levels of data loss. On the other hand, frequently sending a lot of small packets results in unnecessary header overhead. The simulation results suggest that the ZigBee application should send moderate amounts of logged data (i.e., 6000 bit packets) about every half second. Overall, this study suggests that a single team equipped with communication device built with XBee-Pro® ZB module will be able to transmit in real time, using a single ZigBee WPAN, all collected telemetry data, for an entire race team consisting of up to five racecars.

The next step of our project is to develop an Android-based software which will provide an interface between the stand-alone data logging device and with XBee-Pro® ZB module. The racecar portion of the software will retrieve the data from the logger, encode it, and will send it to the pit crew via the XBee module. At the base station the software will receive the data, decode it, and display it to the pit crew in a meaningful format, allowing the base station team to notify the driver in the case of emergency or as desired. Finally, we plan to test the system during one of the 2014 amateur endurance racing events.

REFERENCES

- [1] XBee-Pro® 900 HP (900 MHz RF module) datasheet http://www.digi.com/pdf/ds_xbeepro900hp.pdf, accessed 1/9/2013
- [2] IEEE 802.15™: Wireless Personal Area Networks (PANs), <http://standards.ieee.org/about/get/802/802.15.html>, accessed 1/10/13
- [3] XBee® & XBee-Pro® ZB (ZigBee® Embedded RF Module Family for OEMs) datasheet, http://www.digi.com/pdf/ds_xbeezbmodules.pdf, accessed 1/10/13
- [4] OPNET IT Guru version 17.0. OPNET Technologies, Inc®, <http://www.opnet.com>, accessed 1/10/13
- [5] Trackmaster, Trackaroo LLC, <http://trackmaster.trackaroo.com/>, accessed 1/14/2013
- [6] New Hampshire Motor Speedway, <http://www.nhms.com/>, accessed 1/14/13
- [7] RaceChrono™ - a GPS based lap timing software for S60 and Windows Mobile smart phones. <http://www.racechrono.com/>, Accessed 1/15/12