2950 Niles Road, St. Joseph, MI 49085-9659, USA
269.429.0300 fax 269.429.3852 hq@asabe.org www.asabe.org

An ASABE Meeting Presentation

Paper Number: 131620559

Investigation of Bluetooth Communications for Low-Power Embedded Sensor Networks in Agriculture

Balmos, Andrew D.[1]; Layton, Alex W.[1]; Ault, Aaron[1]; Krogmeier, James V.[1]; Buckmaster, Dennis R.[2]

1. Electrical and Computer Engineering, Purdue University, West Lafayette, IN United States
2. Agricultural and Biological Engineering, Purdue University, West Lafayette, IN United States

Written for presentation at the

2013 ASABE Annual International Meeting

Sponsored by ASABE

Kansas City, Missouri

July 21 – 24, 2013

Abstract. *Wireless sensor networks that monitor agricultural equipment can provide useful data to optimize a farm's productivity. To become widely adopted, such a sensor network requires sensors with battery lives at least as long as a full farming season. The standard low power wireless sensor communication platforms, e.g. Zigbee and ANT, lack common hardware support in consumer devices. Using more ubiquitous technology that is already owned by many farmers, such as cellphones and tablets, can significantly reduce cost and complexity.*

Nearly all mobile devices currently on the market have Bluetooth built-in, making it a viable wireless protocol choice. In addition, the recent release and initial adoption of the Bluetooth 4.0 Low Energy (BLE) standard is a strong candidate for use in very low power, long battery life sensor networking.

This work investigates how various BLE configurations affect network size, network throughput, and sensor battery lifetimes. We present a strategy to select the best network parameters that achieve a certain sensor Update Interval (UI) and Connection Latency (CL), as well as minimize a sensors power requirements, reduce data latency, and conform to size and throughput constraints. We present a simple BLE power model created from current consumption measurements of the commercially available TI CC2540 BLE module. Combining this BLE power model with a simple capacity model of a battery allows estimation and validation of sufficient battery life. Sensor lifetimes on the order of multiple years can easily be achieved for large UI and CL sensors.

Keywords. *Bluetooth, Bluetooth Low Energy (LE/BLE), Sensor, Sensor Network, Farm Data Automation.*

Introduction

Farming has recently seen, and will continue to see, a rapid growth in the use of technology. Farm owners are now better able to manage, plan, and work their farms with precision (Sassenrath, et al. 2008). For example, some tractors use high quality Global Positioning System (GPS) sensors to guide steering, ensuring work covers the entire field while simultaneously preventing wasteful overlap (Li, et al. 2009). However, this type of technology is often less useful than it could be. For example, many situations limit the user to the sensors, monitor, and software installed by the manufacturer. Many times if a farmer could add simple, low cost, wireless sensors that could bypass the limitation of the typical wired communication systems, the results would be significantly more useful and lead to better decision making.

Consider an identification (ID) tag sensor attached to every tractor implement, whose sole purpose is to announce itself to any nearby user. Combined with data from the tractor's GPS a farmer could automatically track not only the field worked with a particular implement but also the specific parts of the field. To further increase accuracy, one could include a simple sensor that would determine if an implement was engaged with the ground or not, e.g., by monitoring pressure within a hydraulic line. Such a sensor could also double as an issue detection system, alerting the user to the failing of the hydraulic line. As a final example of many possibilities, consider adding weight sensors to a silage cart: the weight measurements combined with the tractor's GPS could be used to create silage yield maps, a novel application.

Unfortunately, until recently, most of the traditional wireless communications protocols have had certain shortcomings that raise the potential barrier to adoption. For example, the lack of common commercial off the shelf hardware leads to a high cost of producing, purchasing, and installing an aftermarket system. However, Bluetooth Low Energy (BLE), a new standard in the recent release of the Bluetooth 4.0 specification has significant potential for use in an agricultural sensor network. Previous Bluetooth standards already widely exist in most cell phones and tablets, and adoption of BLE is progressing quickly as well. Therefore, a network clusterhead can be the inexpensive and ubiquitous mobile device, already often owned by individuals. Mobile devices offer users and developers flexible, extendable, and familiar software environments that are already portable and battery operated. However, there remain some unanswered questions about BLE's ability to meet technical constraints:

- can a BLE network support enough sensors,
- does it have sufficient throughput, and
- is it low-power enough?

This paper investigates those questions and presents evidence suggesting BLE is a viable technology for use in an agricultural sensor network. The work outlines the basic design decisions with a simple description of the physical and link layers. The influence of these decisions on the ability of the sensor network to support enough sensors and have sufficient throughput is developed. Using that knowledge, a simple strategy to achieve a particular sensor update interval (UI) and connection latency (CL) that maximizes battery life, reduces data latency, and does not constrain the network is presented. Finally, to show the resulting sensors can achieve sufficient battery life, a basic model is formulated using the sensor's average current and a very simple battery capacity model. This BLE life model makes use of current measurements from a Texas Instruments CC2540 BLE module and is parameterized by UI, CL, and the sensor's yearly usage (YU). With these results, it is clear that sensors equipped with BLE can satisfy an agricultural sensor network's requirements.

Bluetooth Low Energy

BLE, newly defined in the Bluetooth 4.0 Specifications (The Bluetooth Special Interest Group 2010), simplifies the Classic protocol, and eliminates some of the Classic features, in order to achieve power savings. While building off the same core principles, it relaxes some requirements like timing accuracy, channel bandwidth, and modulation filtering. The in-depth details of BLE are outside the scope of this work, see e.g., (Gomez, Oller and Paradells 2012) and (The Bluetooth Special Interest Group 2010) for further information. Here we focus only on the components of BLE that are important for power consumption, particularly on the sensor side.

There are five major operating modes of BLE: Scanning, Advertising, Connected, Initiating, and Standby. The Advertising mode broadcasts data or connection requests to other peers; the Scanning and Initiating modes receive broadcasts and form connections, respectively. Once a connection is established, the Connected mode transmits packets to the connected peer. The Standby mode is simply a low power state reserved for use when the device is not participating in a BLE network.

Figure 1a illustrates the various paths taken to activate a mode. If a device activates Connected mode via the

Initiating mode it becomes a master. On the other hand, if a device activates the Connected mode via the Advertising mode it becomes a slave. At any given time, a slave may only be connected with one master. However, a master may have many simultaneous slave connections. Therefore, a sensor network clusterhead must operate as a master and a sensor as a slave. All communications are between master and slave: slaves do not talk with other slaves nor masters with other masters.

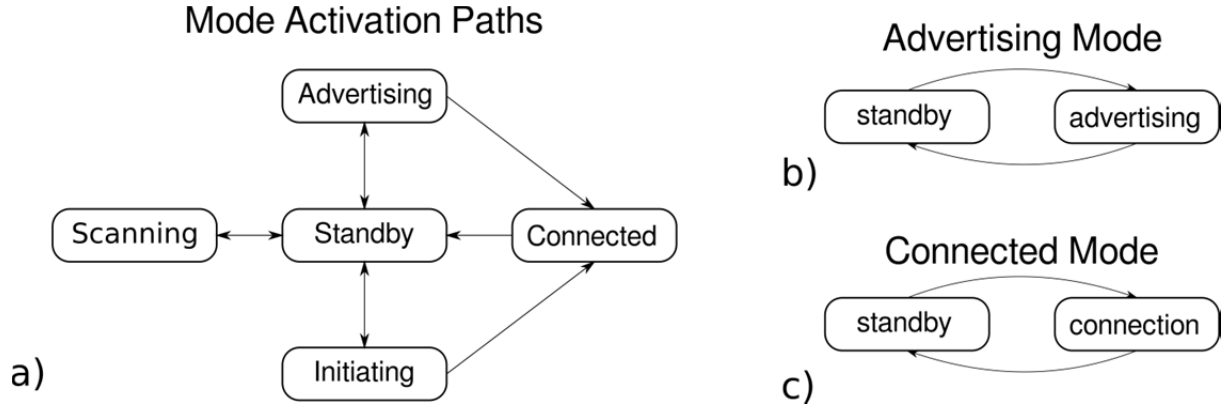

Figure 1: a) A diagram showing the allowable paths a BLE module can take to activate a mode. A sensor would activate Connected mode through Advertising and a clusterhead through Initiating. b) The state diagram of a BLE module exclusively in the Advertising mode. c) The state diagram a BLE module exclusively in the Connected mode.

In the work considered here, we are primarily interested in the sensor side and therefore the mode activations that matter are only the Advertising and Connected modes. We assume in an agricultural sensor network, that sensors will be either in the Advertising mode or in the Connected mode. This is a reasonable assumption because a sensor design concerned with power usage should be simple and would not waste power transmitting in Connected and Advertising modes simultaneously.

Advertising Mode

As shown in Figure 1b, a BLE device exclusively in the advertising mode has only two possible states: standby and advertising. The standby state has no required tasks and therefore most of the hardware is asleep, consuming very little power. However, when in the advertising state the hardware is consecutively transmitting an advertisement packet onto three separate radio frequencies. The advertising state may require a relatively high peak power when transmitting and this peak power necessitates high peak currents that have an impact on battery choice, but this is beyond the scope of the work considered here.

As illustrated in Figure 2, Advertising mode is characterized by only one parameter, advInterval, which is the desired length of time between two advertising states. To help prevent BLE devices having the same advInterval from repeatedly interfering with each other, each device automatically adds a random delay, *advDelay*, to advInterval at the beginning of every advertising state. advDelay is between 0 ms and 10 ms and is chosen independently each time is it used.

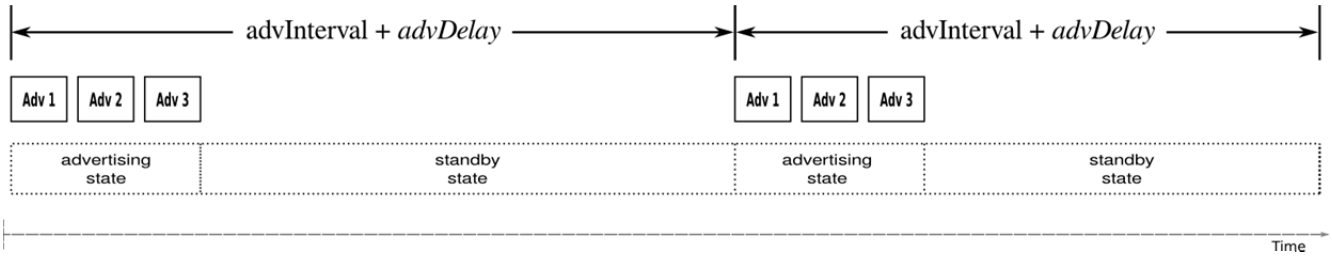

Figure 2: An example of the advertising packets transmitted during the Advertising mode. Transmissions are represented by solid line rectangles; Adv 1, Adv2, and Adv 3 are transmitted on a separate radio frequency reserved for advertisements only. *advDelay* is a random delay added by the BLE physical layer to avoid continuous collisions with other advertising devices. Also noted are the states of the device over time, represented as dotted rectangles.

All devices may have their own value for advInterval; however, it must remain between 20 ms to 10.24 seconds and can only be incremented in steps of 625 µs. As advInterval increases, the device transmits less often resulting in smaller power consumption.

Connected Mode

As shown in Figure 1c, a BLE device exclusively in the connected mode only has two possible states: standby and connection. Again, the standby state has no required tasks and therefore most of the hardware is asleep and consuming very little power. However, the connection state always transmits at least one packet to its connected peer, even if there is no data to send. The connection state may require high peak power and therefore high peak currents that could affect battery selection.

As illustrated in Figure 3, Connected mode is partially characterized by a shared parameter connInterval. It is the desired length of time between connection states. During a connection state, the master addresses the slave by transmitting a packet, and the slave responds by transmitting a packet, completing a round-trip event. If either device has no data to send, it sends a packet with an empty data payload. A slave using the slave latency feature to save power may skip the connection state for up to connSlaveLatency consecutive lengths of connInterval time. Figure 4 is a flow chart describing the decisions made when considering whether to enter the connection state or not.

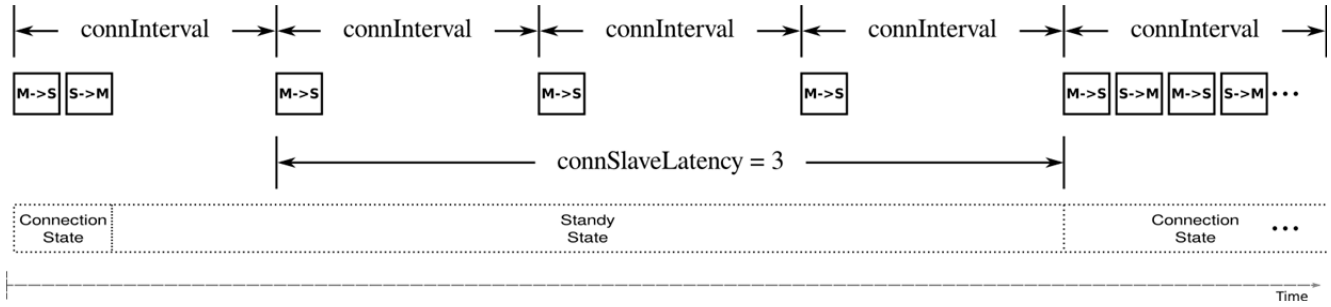

Figure 3: An example of the interaction between a connected master and slave. In this particular connection, connSlaveLatency is set to three and the slave elects to miss the full length of time. The first connInterval shown has only the initial round-trip event. However, in the last connInterval shown the master has initiated several round-trip events. Solid line rectangles labeled M->S are packets sent from the master to the slave and the response packets from the slave to the master are labeled S->M. Also noted are the states of the slave device over time, represented as dotted rectangles.

connInterval can be set to a value between 7.5 ms and 4 seconds in steps of 1.25 ms, and connSlaveLatency can be any integer between 0 and 499 connection states. As connInterval increases, the device transmits less often, resulting in smaller power consumption. When a slave increases connSlaveLatency it can choose to transmit less and therefore reduce the power consumption further.

The example in Figure 3 indicates that there may be more than one round-trip event per connection state. The master determines the number of round-trips that occur. If either the master or the slave has more data to send, the master can elect to continue the conversation. However, communication must not be continued if any of the following conditions occur:

- the master has commitments to other slaves,
- there is not enough room in the current connInterval to complete a full round-trip,
- neither side has more data to send,
- or if too many errors have occurred.

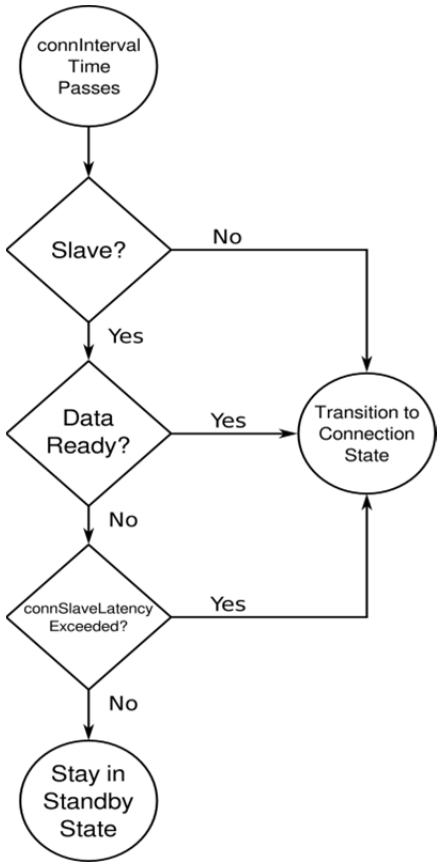

Figure 4: The decision logic used by a BLE device when considering whether to enter the connection state or not.

Connection mode can be terminated upon the request of either the master or the slave, or because the physical radio link is broken for longer than connSupervisionTimeout seconds. Therefore, it is required that Equation 1 be satisfied.

$$(\text{connSlaveLatency} + 1) \times \text{connInterval} \leq \text{connSupervisionTimeout} \quad (1)$$

In agricultural scenarios, it is unlikely that sensors would need to quickly disconnect from one clusterhead and connect to another; therefore, it is assumed that connSupervisionTimeout should always be set to its maximum value of 32 seconds. As an example, if the tractor driver changes, all of the tractor's sensors need to establish a connection with the new operator's mobile device (clusterhead). However, the sensors would not know to do this until the connection with the old, now unavailable, clusterhead is determined lost, connSupervisionTimeout seconds later. Even with the maximum timeout, the new connections would typically form before the operator had the opportunity to begin any work. In the event that this is not true, a sensor may buffer data during the outage, ensuring no data loss. The connSupervisionTimeout parameters may range from 100 ms to 32 seconds in steps of 10 ms. connSupervisionTimeout has no direct effect on power consumption; however, if set too small, its limitations on other parameters could result in suboptimal power usage.

Higher Layers

Higher layers of BLE include features to help facilitate the communication and data transfer; in general, these protocols are not important when selecting the physical layer parameters. However, to calculate network throughput it is important to know how many data octets are in a packet. We assume the sensor will utilize a Generic Attribute Profile (GATT) feature called *notification*. A GATT notification allows a slave to send 20 octets, or 160 bits, of data automatically. Therefore, the master does not need to waste any time or throughput sending specific read requests. We assume for the remainder of this work during Connection mode, all packets sent by the slaves are GATT notifications and all packets sent by the master are empty, which represents the

most common scenario of sensors generating data to be utilized at the clusterhead.

Network Design Problem

Table 1 reviews the parameters of a BLE sensor network design. Power requirements of the master are included only for completeness. The Advertising mode parameters do not influence the Connected mode and vice versa; therefore, we can view them as two independent problems that can be solved individually.

Table 1: A review of a BLE networks parameters and their effect on power consumption and data latency. The effects on the master are included only for completeness.

BLE parameter	Range	Step	Advertiser/Slave Power	Scanner/Master Power	Latency
Advertising Mode					***Connection***
advInterval	20 ms - 10.24 s	0.625 ms	Inversely proportional	Proportional	Proportional
Connected Mode					***Data***
connInterval	7.5 ms - 4 s	1.25 ms	Inversely proportional	Inversely Proportional	Proportional
connSlaveLatency	0 - 499	1	Inversely proportional	Not affected	Slave to master not affected Master to slave proportional
connSupervionTimeout	100 ms - 32 s	10 ms	N/A	N/A	N/A

The Advertising mode design is simple; it only requires the selection of advInterval. When there is a nearby clusterhead initiating, advInterval sets the worst-case connection latency (CL). Other than impacts on battery life, advInterval does not affect the other network constraints, therefore only CL is important when determining its value.

The Connected mode design is slightly more complex, requiring the selection of: connInterval, connSlaveLatency, and connSupervisionTimeout. From an earlier assumption, connSupervisionTimeout is set 32 seconds. Every update interval (UI), i.e., the interval of time between sensor data being available, may have many possible configuration options. The designer must choose the best one that still meets the network requirements.

The following work develops a strategy for selecting the configuration options that maximize battery life while reducing data latency, and does not over-constrain the number of sensors or restrict the network's throughput too severely.

Network Size Constraints

To develop a configuration scheme that respects a necessary network size, the reason network size matters must be understood. The BLE specifications technically allow a network to have a nearly unlimited number of sensors; however, there is a bottleneck. For a master to ensure that it will not miss any connection states, it may not schedule a connection state to begin during the first round-trip of another. Therefore, as shown in Equation 2, the maximum number of slaves supported in a network (N_S) is the number of complete round-trips that fit within the smallest connInterval of the entire network. The length of a round-trip for the type of network considered in this paper is $676\ \mu s$.

$$N_s = \left\lfloor \frac{\text{connInterval}}{676\ \mu\text{s}} \right\rfloor \tag{2}$$

The number of supported slaves is, in general, large compared to the number of sensors that may be in operation at once for a typical agricultural sensor network. However, the network does become limited if there is a sensor with a small UI. An example of this would be, a low UI sensor operating with large connSlaveLatency to reduce power consumption that also uses the smallest connInterval (7.5 ms) to improve its data latency. In

this case, the network is limited to 11 total sensors ($7.5\ \text{ms} / 0.676\ \text{ms}$).

If network size is of concern, a network designer should use Equation 2 to determine a lower bound on connInterval based on the projected maximum network size.

Network Throughput Constraint

Similar to the network size constraint, one must understand how throughput is restricted in order to design a configuration strategy. The maximum achievable throughput is a completely full notification (20 octets or 160 bits), multiplied by the number of round-trips per second. The restriction on the number of packets per second is from the requirement that a master not miss the beginning of any connection states. Therefore, assuming the master has scheduled all of the slaves such that there is not any wasted time, the number of packets that can fit within the smallest connInterval of the entire network is also the largest block of consecutive packets. For example, either the network has the maximum number of slaves, each completing only one round-trip, or there are fewer slaves, some transmitting multiple notifications per connection state. Therefore, the total network throughput (T) follows Equation 3.

$$T = \frac{160\ \text{bits}}{\text{connInterval}} \left\lfloor \frac{\text{connInterval}}{676\ \mu\text{s}} \right\rfloor \tag{3}$$

As can be seen in Figure 5, the network throughput staggers up and down. The stagger becomes less in amplitude and approaches an asymptote of $236.86\ \text{kbits/sec}$ as connInterval is increased. This stagger is a result of there not being enough time to complete a round-trip communication at the end of some connIntervals, and therefore the time is wasted. As connInterval becomes larger, the wasted time is a smaller percentage of the total time transmitting, so the throughput is less affected.

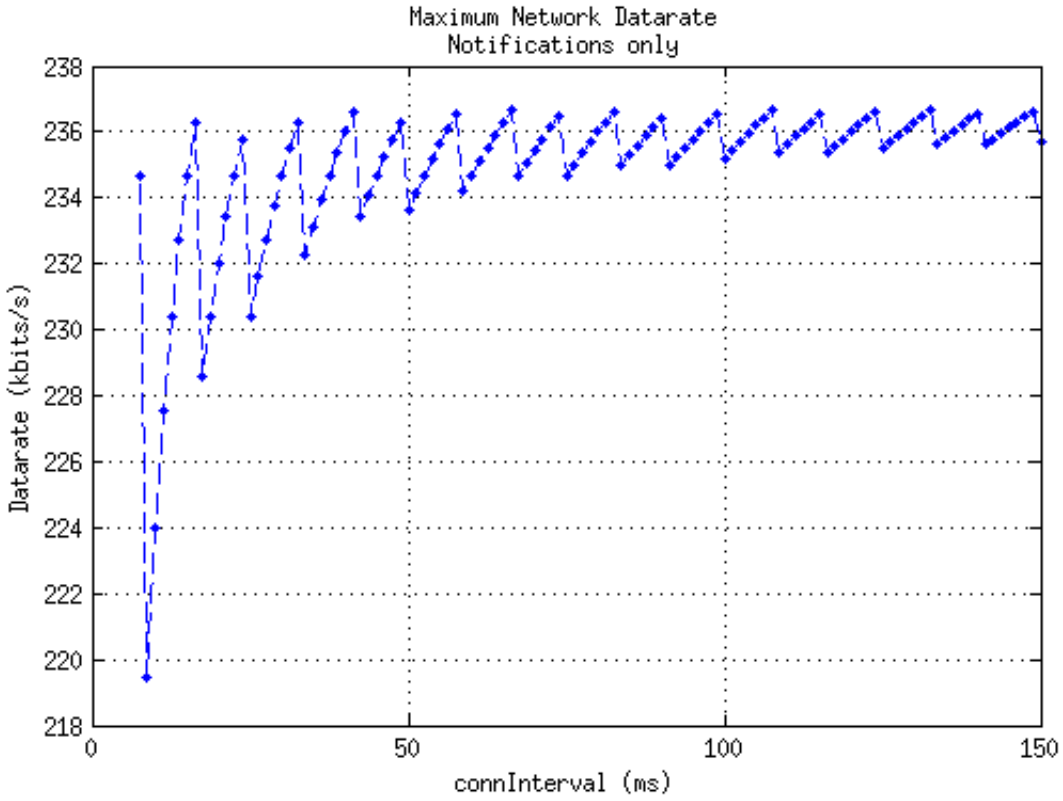

Figure 5: The theoretical maximum network throughput, shared amongst the slaves, as a function of the smallest connInterval in an entire network. The trend continues, eventually reaching the maximum of 236.86 kbits/sec, as connInterval is increased to 4 seconds.

Network throughput is not likely to be an issue for an agricultural BLE sensor network because even the minimum theoretical throughput, $219.4\ \text{kbits/sec}$, is significantly larger than the needs of a typical sensor. As an example, again consider the silage-yield mapping weight sensor. If the tractor were using a low-cost GPS, roughly 3 meters of error, and traveling at the speed of $5\ \text{mph}\ (2.2\ \text{m/s})$, the sensor would need an UI of roughly 1.3 seconds to produce a yield map with the same level of spatial error. Assuming the weight measurement requires 5 octets of data for each set of axels, and that the silage cart has three, then an overall measurement is 15 octets in length. Therefore, the sensor's throughput needs are 15 octets of data per 1.3 seconds, or .09 kbits/sec. If the GPS was of the best quality, roughly 3 cm of error, then the sensor would require a UI of 13.6ms, and therefore a throughput of roughly $8.8\ \text{kbits/sec}$. It would be hard to imagine a silage yield map with a resolution of 1 cm or even a weight sensor that could provide accurate measurements that quickly, so it is likely the designer would probably choose a larger UI. However, this is a good example of the throughput requirements for a very low UI sensor, such as a sensor logging data at a high rate for later processing.

If the network throughput is of concern, a network designer should use Equation 3 to determine a lower bound on the value of connInterval based on the required network throughput.

Parameter Selection

This section presents a simple method that a network designer could use to select the network parameters that meet the required UI, CL, network size, and network throughput.

Advertising Mode

There is only one parameter to select in Advertising mode, advInterval. As mentioned previously, this parameter is entirely based on the sensor's required CL. In order to maximize battery life, advInterval should equal the maximum CL the sensor can tolerate, thereby minimizing the required number of advertisement transmissions and therefore minimizing the mode's power. For example, an ID tag can easily use the largest CL (10.24 seconds), because in general, it takes a much longer time for an operator to begin work after arriving at a tractor. However, weight sensors on a silage cart may need to connect quickly because it typically pulls along the side a tractor already in operation.

Connected Mode

An obvious strategy for picking connInterval and connSlaveLatency is to maximize connInterval and then maximize connSlaveLatency, ensuring that the desired UI is achieved, i.e., $\text{connInterval} \times \text{connSlaveLatency} = \text{UI}$. However, this method provides the worst-case data latency. If the data is ready early or errors caused the initial round-trip to fail, then the slave may have to wait the full connInterval again to transmit. Because of this, we alternatively opt to minimize connInterval and maximize connSlaveLatency while still meeting the UI requirement.

The alternative method achieves the maximum battery life because the sensor is still only required to transmit once per UI. However, because connInterval is minimized, the potential data latency is reduced. Equation 4 provides the optimal value of connInterval. However, connInterval may be set to a larger value if required for network size and/or network throughput constraints. Equation 5 calculates the necessary connSlaveLatency to meet the UI requirement and still minimize power consumption. As written, the three input minimize function of Equation 5 ensures that Equation 1 is satisfied and that the allowable range of connSlaveLatency (maximum of 499) is respected. If the UI is not exactly achievable, Equation 5 will slightly undershoot the interval. If preferred, Equation 5' can be used instead to produce a configuration that meets or slightly overshoots the UI. Note that these equations only differ in the use of the ceiling versus the floor operator in the first argument of the minimum function.

$$\text{connInterval} = \begin{cases} 7.5\ \text{ms} & 0 < \text{UI} < 3.7425\ \text{s} \\ \left\lfloor \frac{\text{UI}}{499 \times 1.25\ \text{ms}} \right\rfloor \times 1.25\ \text{ms} & 3.7425\ \text{s} \le \text{UI} < 32\ \text{s} \\ 63.75\ \text{ms} & \text{UI} \ge 32\ \text{s} \end{cases} \tag{4}$$

$$\text{connSlaveLatency} = \min\left(\left\lfloor \frac{\text{UI}}{\text{connInterval}} \right\rfloor, \left\lceil \frac{32\ \text{s}}{\text{connInterval}} \right\rceil - 1, 499\right) \tag{5}$$

$$\text{connSlaveLatency} = \min\left(\left\lceil \frac{\text{UI}}{\text{connInterval}} \right\rceil, \left\lceil \frac{32\ \text{s}}{\text{connInterval}} \right\rceil - 1, 499\right) \tag{5'}$$

If UI exceeds connSupervisionTimeout (32 s), then the method is no longer able to minimize the slave's power in the same sense. In this case, the sensor cannot be silent for the full UI because a supervision timeout would occur. Therefore, the method selects network parameters such that a slave connection event occurs once per connSupervisionTimeout as well as minimizes connInterval.

As an example of the method's usage, consider the silage yield-mapping weight sensors discussed earlier. It required UIs of 1.3 seconds and 13.6 ms for the low-cost and high quality GPS situations, respectively. Therefore, using Equation 4, connInterval is set to 7.5 ms in both cases. Equation 5 produces a connSlaveLatency of 173 for the low-cost GPS, resulting in sensor update every 1.2975 seconds, and 1 for the high quality GPS, resulting in a sensor update every 7.5 ms.

As an additional example, consider the implement-engaged sensor that reports if a tractor's implement is working the ground. Assume that it is updating at a speed suitable for pairing with a low-cost GPS and is in a network that has a size restriction of at least 55 sensors. In this case, connInterval must be at least 37.5 ms to allow for that many sensors. However, equation 4 suggests the optimal value for only power usage optimization is 7.5ms. Because the network size restriction requires the larger 37.5 ms, it must be chosen instead of the 7.5ms value, sacrificing some power usage in order to support a large network size. As a result, the connSlaveLatency is set to 35, resulting in a sensor update every 1.3125 seconds.

BLE Battery Model

In order to estimate the battery life of a BLE sensor, a power usage model is required. Therefore, we determine the average current from a chosen usage scenario and from current measurements of a Texas Instruments CC2540 BLE module. To do so, we utilized the CC2540DK-MINI development kit from Texas Instruments. However, the development board has various sensors unrelated to the operation of Bluetooth that would affect the current measurements. As suggested by the AN092 application notes (Kamath and Lindh 2012), we modified the board by removing the unnecessary parts and added connections to bypass the onboard 3.0 V coin cell battery so that a measureable external power source could be used.

To measure the current draw versus time we installed a .47 Ω shunt resistor between the external power supply and the development board. Then we used a Texas Instruments INA210 Bi-Directional Zero-Drift Series Current Shunt Monitor and an Agilent MSO-X 4024A oscilloscope's capture mode to record the instantaneous current draws.

Figure 6 and Figure 7 are plots of several selected examples of the current profiles recorded during an advertising and connection state, respectively. The plots were captured by triggering on a rush current spike that appears as the device is transitioning out of the standby state. Clearly there is some variation in the timing of these events, although the size and shapes of the various curves are quite predictable. An advertising state requires an average current draw of 8.616 mA and takes 3.994 ms to complete. A connection state requires an average current draw of 7.793 mA and takes 3.027 ms to complete. The average current consumption during the standby state is 1.1 μA. For all transmissions the transmit power was set to 0 dBm.

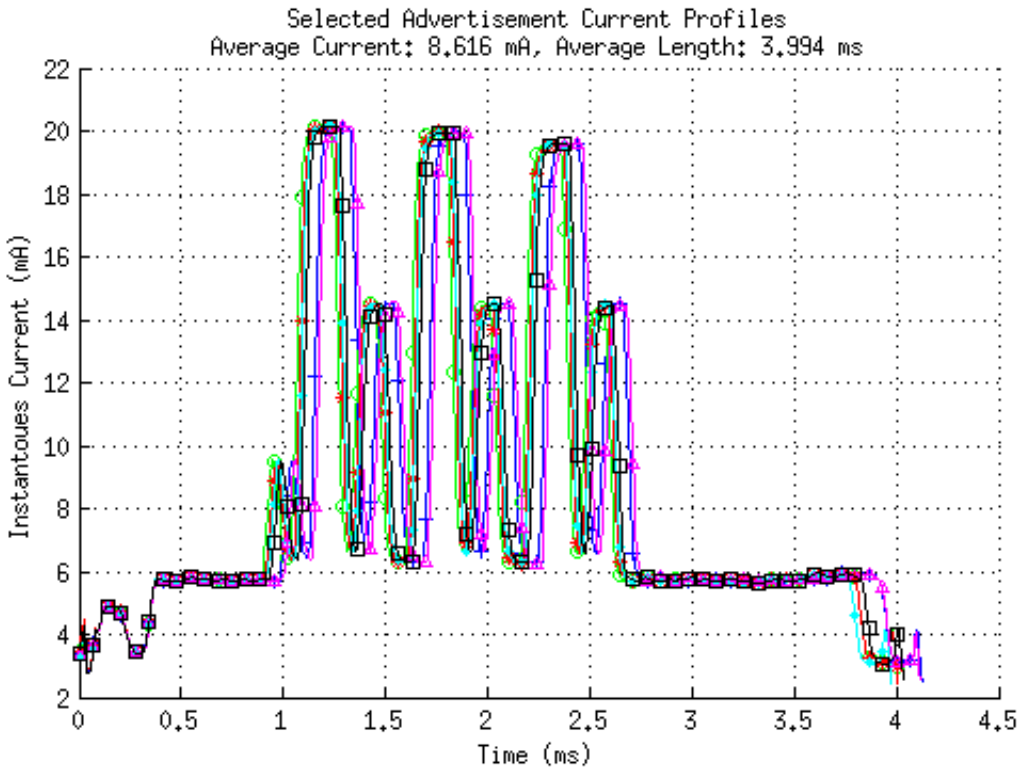

Figure 6: Various selected current profiles from a CC2540 during the advertising state. The profiles were captured by triggering on a current rush caused by the transition from standby state. The CC2540 transmit power was sent to 0 dBm.

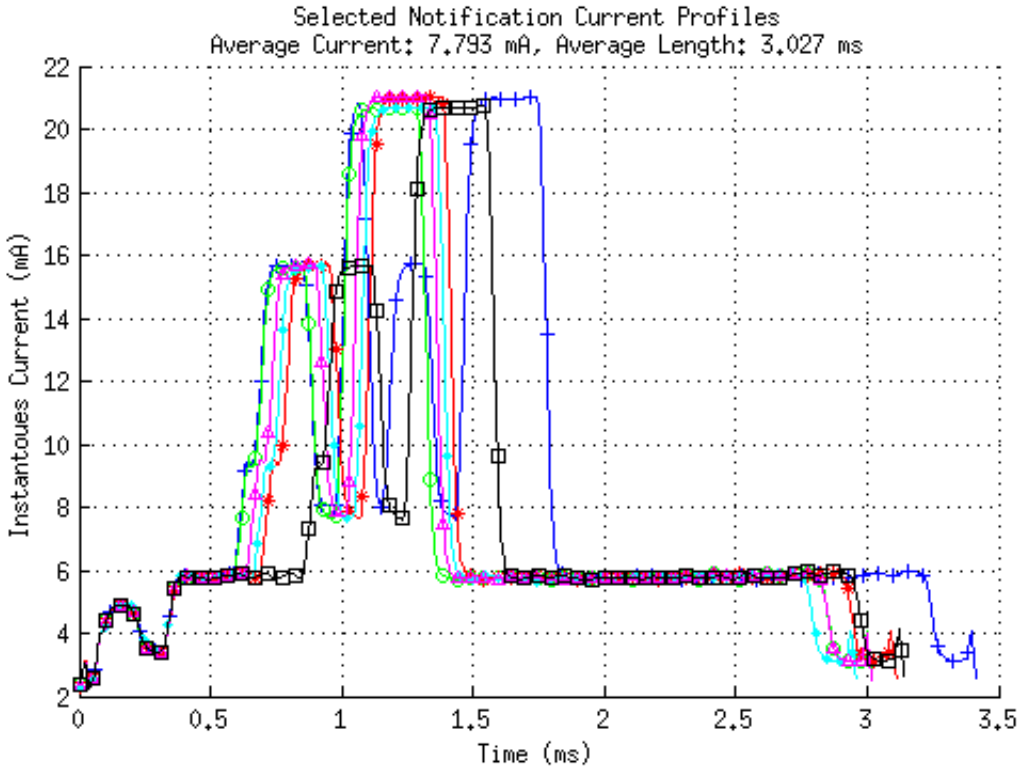

Figure 7: Various selected current profiles from a CC2540 during the connection state. The profiles were captured by triggering on a current rush caused by the transition from standby state. The CC2540 transmit power was sent to 0 dBm.

To calculate the average current, there needs to be an assumption on the sensor's usage. Here we define a

parameter yearly usage (YU) that is the number of hours per year that the sensor is in connected mode, entering the connection state every UI seconds. The rest of the time, the sensor is in advertising mode entering the adverting state every CL seconds.

The total time per year that device is in the connected state is therefore, $T_c = YU \times 60 \times 60$ seconds and the time spent in the advertising state is $T_a = 365 \times 24 \times 60 \times 60 - T_c$ seconds. When in the advertising state, the total time spent transmitting advertisements equals $T_a^T = 3.994\ \text{ms}\ \left(\frac{T_a}{AI}\right)$ seconds and the total time in standby is $T_a^S = T_a - T_a^T$ seconds. Similarly, for the connected state $T_c^T = 3.027\ \text{ms}\ \left(\frac{T_c}{UI}\right)$ seconds and $T_c^S = T_c - T_c^T$ seconds. Therefore, the average yearly current as a function of UI, AI, and YU is given in Equation 6.

$$\hat{I} = \frac{8.616\ \text{mA} \times T_a^T + 7.793\ \text{mA} \times T_c^T + 1.1\ \mu\text{A} \times (T_c^S + T_a^S)}{T_a + T_c} \tag{6}$$

A simple battery capacity model, shown in Equation 7, is used to estimate the time until a battery would die under the average yearly current usage.

$$T_{death} = \frac{\text{Capacity}}{\hat{I}} \tag{7}$$

Figure 8 shows example battery life estimates curves as a function of UI, AI, and YU, assuming the sensor was configured as suggested in Equation 4 and 5. Clearly, the AI is very important and can have dramatic effects on the sensor's lifetime. This is due to a combination of advertising being more costly in terms of power than notifications and the sensor spending the majority of its life in the Advertising mode. This leads to a curious result; large UI sensors have longer battery lives when in use more often, because for large UI sensors the connected state is the lower power state. In fact, the lifetime curves cross each other at the UI where the power spent in the Advertising mode equals the power in the Connected mode. To the left of the crossing point, Connected mode uses the most power and to the right, Advertising mode is more expensive. For very large UIs, not shown here, the battery life model can predict unbelievably large lifespans. In this case, second order battery effects, such as self-discharge, non-constant current discharge, and battery inefficiency, would be the limiting factor for battery lifetime.

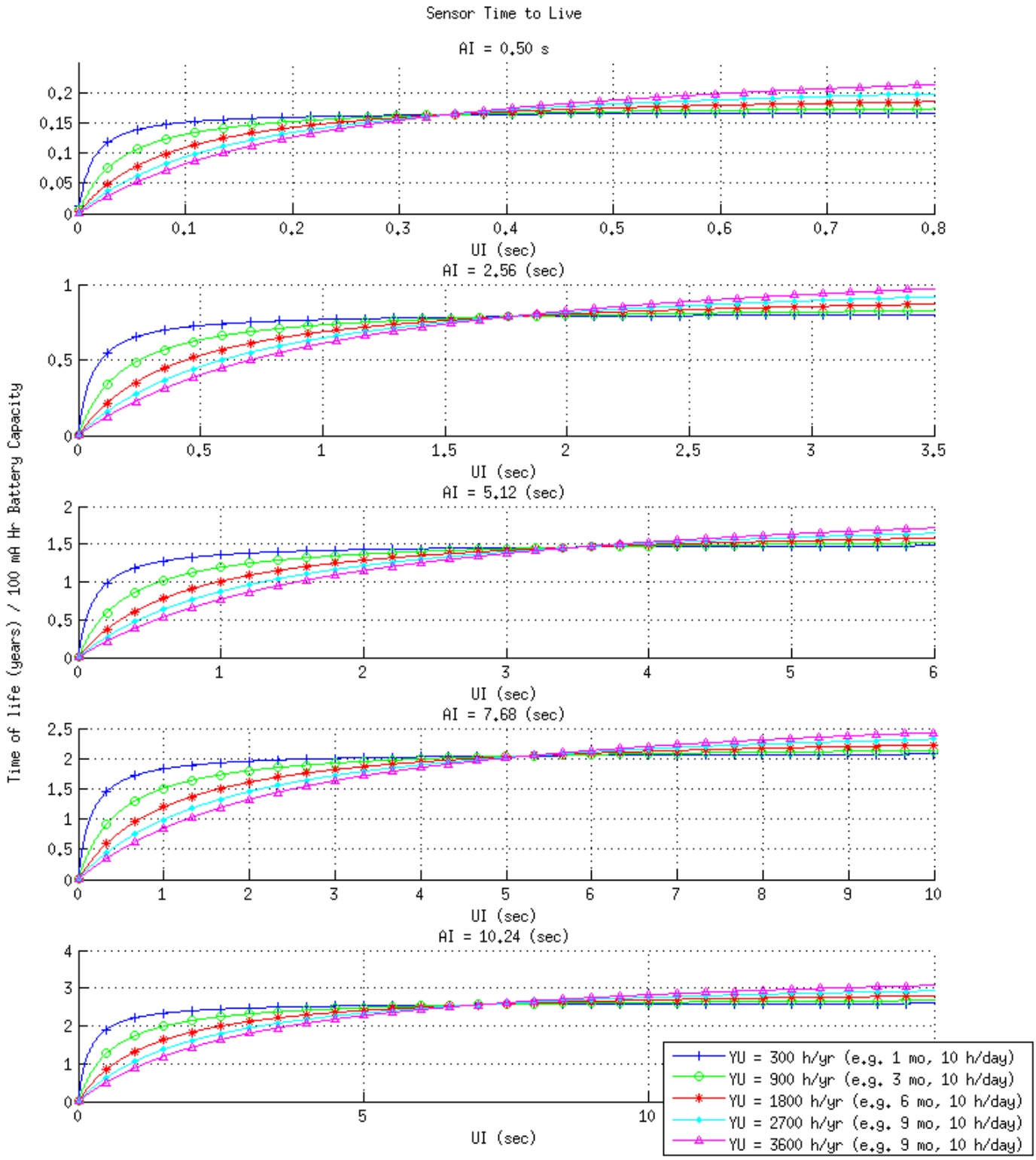

Figure 8: Expected years of life per 100 mA-Hr of battery capacity for various AI, UI, YU values, assuming the sensor has the configuration given by Equation 4 and 5. The expected battery life is highly dependent on the value of AI. For most reasonable AI and UI choices in an agricultural sensor network, the expected battery life is longer than a farming season.

Clearly, the expected battery lives, for reasonable values of UI, AI, YU, are, in general, longer than a typical farming season. Only for very low AI and UI sensors, rare in an agricultural sensor network, would a sensor die midseason requiring a battery change.

To exemplify this, the ID sensor tag that has been configured with the largest AI, 10.24 seconds, and a UI of 15 seconds can be expected to last anywhere from 2.6 to 3.1 years on a single 100 mA-Hr coin cell battery for

usages between 300 and 3600 hours per year, respectively.

A slightly more intense data example, the implement-engaged detection sensor is expected to last approximately 2.3 years when configured for a low-cost GPS (UI is 1.3 seconds), the AI set to the maximum 10.24 seconds, and a YU of 300 hours per year. However, the silage yield mapping weight sensor, configured the same way, but with a more appropriate $\mathrm{AI\ of}\ 2.56$ seconds, is expected to last 0.8 years, or about 10 months.

Finally as the high data rate sensor example, consider the silage yield mapping weight sensor configured for a high quality GPS. That is, a sensor with a UI selection of $13.6\ \mathrm{ms}$, AI equal to 2.56 seconds, and a YU of 300 hours per year, is expected to last $.14$ years, or about two months.

Conclusion

This paper presented several basic, yet representative examples of sensors that would be used in an agricultural wireless sensor network, and has demonstrated their viability. In particular, this paper presented:

- evidence that a BLE-equipped sensor in an agricultural sensor network can easily achieve sufficient support for network size, network throughput, and most importantly produce sensors with battery lives at least as long, if not longer, than a normal farming season,
- how the BLE parameters affect the possible network size and throughput,
- a strategy to configure the network to maximize the battery life and reduce data latency,
- and a battery life model that can estimate the life of a sensor in a certain configuration to verify the sensor will operate with sufficient lifetime.

Acknowledgements

The work reported in this paper was partially supported by a USDA NIFA grant titled “Improving Agricultural Management with Autogenic Mobile Technology”.

References

Gomez, Carles, Joaquim Oller, and Josep Paradells. "Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology." *Sensors*, 2012: 11734-11753.

Kamath, Sandeep, and Joakim Lindh. *Measuring Bluetooth® Low Energy Power Consumption.* Application Notes, Dallas: Texas Instruments Incorporated, 2012.

Li, Ming, Kenji Imou, Katsuhiro Wakabayashi, and Shinya Yokoyama. "Review of research on agricultural vehicle autonomous guidance." *International Journal of Agricultural and Biological Engineering.* September 2009.

Sassenrath, G F, et al. "Technology, complexity and change in agricultural production systems." *Renewable Agriculture and Food Systems.* Cambridge: Cambridge University Press, December 2008.

The Bluetooth Special Interest Group. "Specification of the Bluetooth System, Covered Core Package verion: 4.0." Standard, Kirkland, 2010.