Nemo: A High-fidelity Noninvasive Power Meter System for Wireless Sensor Networks

Ruogu Zhou, Guoliang Xing
Department of Computer Science and Engineering, Michigan State University, USA
{zhouruog,glxing}@cse.msu.edu

ABSTRACT

In this paper, we present the design and implementation of *Nemo* – a practical *in situ* power metering system for wireless sensor networks. Nemo features a new circuit design called *shunt resistor switch* that can dynamically adjust the resistance of shunt resistors based on the current load. This allows Nemo to achieve a wide dynamic current range and high measurement accuracy. Nemo transmits real-time power measurements to the host node solely through the power line, by modulating the current load and the supply voltage. This feature leads to a noninvasive, plug & play design that allows Nemo to be easily installed on existing mote platforms without physical wiring or soldering. We have implemented a prototype of Nemo and conducted extensive experimental evaluation. Our results show that Nemo can transmit high-throughput measurement data to the host through voltage/current load modulation. Moreover, it has satisfactory measurement fidelity over a wide range of operating conditions. In particular, Nemo yields a dynamic measurement range of 250,000:1, which is 2.5X and 7X that of two state-of-the-art sensor network power meter systems, while only incurring an average measurement error of 1.34%. We also use a case study to demonstrate that Nemo is able to track the highly dynamic sleep current consumption of TelosB motes, which has important implications for the design of low duty-cycle sensor networks that operate in dynamic environments.

Categories and Subject Descriptors

C.2.3 [**Computer-communication Networks**]: Network Operations—*Network monitoring, Network management*; C.4 [**Performance of Systems**]: Measurement techniques

Keywords

Wireless Sensor Networks, Power Monitoring, Power-line Communication

IPSN'13, April 8–11, 2013, Philadelphia, Pennsylvania, USA.

1. INTRODUCTION

Energy-efficiency is one of the most important design objectives of wireless sensor networks due to limited energy resources. Despite the significant research efforts in energy-aware approaches at various network layers (MAC/routing/application), it remains challenging to actually validate the energy-efficiency claims of existing solutions, largely due to the lack of ability to track the real-time power consumption of a sensor network at runtime. In addition, real-time power usage data is vital for senor nodes to modify their behavior and adapt to variable network conditions and dynamic physical environments. For example, in the four-month habitat monitoring sensor network deployment on Great Duck Island, many nodes experienced unknown energy issues and died prematurely [19]. If low-power *in-situ* meters were employed to continuously monitor the power consumption of nodes, node failures may be diagnosed or even avoided through runtime adaptation.

The aforementioned requirements have motivated the development of *in-situ* power metering systems [5] [12] [17] [11] [8] [13] [18] that can measure the power consumption of sensor nodes in real-time. A practical power meter system must meet several key requirements due to the unique characteristics of wireless sensor networks. First, it must achieve high measurement fidelity, including wide dynamic range, high sampling rate, high measurement resolution and accuracy. The current consumption of a sensor node is highly dynamic and has a wide range of at least 5 orders of magnitude, from about a few uA in sleep state to about several hundreds mA in active state. The high resolution measurement of low current consumption (<10 uA) is particularly important because most sensor networks operate under low duty-cycles and their lifetime is largely determined by the sleep current consumption.

Second, a power meter should be minimally *invasive* to the host node in terms of both installation and operation. Most existing sensor network platforms do not have any built-in power metering capability. To be practically useful, a power meter should be easy to install on existing sensor hardware, with little or no physical wiring/soldering. Moreover, it should operate in a stand alone manner, without relying on host resources like memory and CPU. This ensures that the performance of the host nodes is not compromised in the presence of power metering, improving the fidelity of measurement.

Third, a power meter must be able to communicate with the host node in real time. This will not only allow the host node to dynamically configure the meter, but also enable

real-time feedback of power measurement to the host node for run-time adaptation. Unfortunately, the requirement of real-time host-meter communication often leads to an invasive hardware design. For instance, although I/O ports of MCU can implement high-speed data transfer, they require physical wiring or soldering between power meter and the host node.

In this paper, we present *Nemo* – a Noninvasive high-fidElity power-Meter for sensOrnets. As a key advantage, Nemo connects to the host node using only the power/-ground lines, requiring no dedicated data communication wires. At the same time, Nemo implements real-time, high-speed bi-directional communication with the host node based on current/voltage modulation, in which the current load and the supply voltage of power line are modulated to carry information. The power line communication based on current/voltage modulation allows Nemo to retrofit existing sensor network platforms with power metering capability via a wire-free, plug & play installation. Nemo also employs a circuit design called *shunt resistor switch* that can dynamically adjust the resistance of shunt resistors based on the current load. This allows Nemo to achieve a wide dynamic current range without resorting to expensive and power-hungry components like high-resolution analog-to-digital converters (ADCs).

We have implemented a prototype of Nemo (as shown in Fig. 2) and conducted extensive experimental evaluation. Our results show that Nemo has satisfactory measurement fidelity under a range of operating conditions. In particular, Nemo yields a dynamic measurement range from 0.8 uA to 200 mA, a sampling rate of 8 KHz, and a minimum resolution of 0.013 uA, while only incurring an average measurement error of 1.34%. We also present a case study where Nemo is used to track the current consumption of TelosB motes. Our results reveal that the sleep current consumption varies significantly (as much as five times) with environmental temperature and also across different motes. This finding has important implications for the design of low duty-cycle sensor networks that operate in dynamic environments, demonstrating the benefits of high-fidelity *in-situ* power measurement using Nemo.

2. RELATED WORK

A common practice in sensor network design is to infer the power consumption of a node based on the expected active time of the components and their power consumption models measured offline [16]. A representative example of such approach is PowerTOSSIM [16]. However, the power models measured in laboratory settings cannot reflect the variations of hardware components and environmental factors. Due to this drawback, most software based power estimation approaches suffer large estimation errors. It is shown in [20] that the simulation error of PowerTOSSIM [16] can be as high as 30%.

In-situ power meters can provide run-time power consumption of a hardware device. Commercial power monitoring ICs such as DS2438, BQ2019 and ADE7753 are widely used in portable devices like cellphones for real-time battery monitoring. However, to the best of our knowledge, none of these ICs can meet the requirements of power metering in sensor networks, including wide dynamic range ($10^5 : 1$) and high sampling rate (> 5 KHz). For example, DS2438 only provides a maximum dynamic range of 1024:1 and a sampling rate up to 40 Hz [2].

Targeted low-power sensor networks, SPOT [11] provides a dynamic range of 45,000:1 and a resolution of lower than 1 uA. However, since SPOT needs an external +5.5V power supply, it cannot be directly powered by the onboard batteries. Although SPOT is designed to be integrated with current sensor platforms, it still requires wiring and soldering to the I/O pins of the sensor board. Moreover, SPOT measures energy consumption over a time period, rather than real-time fine-grained power consumption.

iCount [5] is another example of *in-situ* power meters for sensor networks. iCount measures power consumption by differentiating the measured energy, which is inferred from the frequency of the pulses appeared on the inductor pin of the switching regulator. Due to the low oscillating frequency of switching regulators, the sampling rate and resolution of iCount are significantly limited (only 80Hz sampling rate when resolution is 100 uA). Moreover, due to the nonlinear frequency-current relationship of the switching regulator, iCount suffers high measurement errors (up to 20%) [5]. Finally, iCount cannot work in a stand alone manner and must rely on onboard resources (CPU, RAM and timer). Therefore, it often incurs considerable computational overhead (a minimum of 13% host CPU time when sampling at 8 KHz) and cannot conduct measurement when the host falls asleep.

The current/voltage modulation schemes adopted by Nemo are inspired by the power-line networking technology [21]. However, the power-line networking literature adopts sophisticated techniques, such as OFDM, to modulate the AC voltage of power grid infrastructures for LAN communication, which is significantly different from the voltage/current modulation scheme we propose for low-power mote-class platforms. Several technologies, such as I2C, can realize low-power bi-directional communication over single data wire. However, they are not applicable to Nemo, which utilizes a single power wire, instead of a dedicated data wire, for bi-directional communication.

3. SYSTEM OVERVIEW

3.1 Design Objectives and Challenges

High measurement fidelity. In this work, the fidelity requirement includes wide dynamic range, high sampling rate, high measurement resolution and accuracy. In the design of Nemo, we mainly focus on the first two metrics as high resolution and accuracy are relatively easy to achieve as shown in our experiments in Sec. 7.

The current draw of a sensor node in active state ranges from 2 mA to 200 mA [1] [3] [22]. The sleep current consumption ranges from 2 uA [11] to several hundred uA. Although the sleep current consumption seems to be negligible, it largely determines the system lifetime of low duty-cycle networks. For example, for a TelosB mote with a 0.1% duty cycle, a mere 10 uA increase of sleep current will shorten the mote lifetime by 26%. Moreover, the sleep currents of even the same type of sensor nodes may differ significantly due to environmental factors (see Section 7.5), on-board components that sleep independently of CPU [9], and misconfigurations due to software bugs [23]. To accurately measure the current draw of both active and sleep

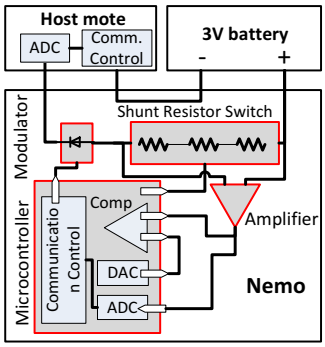

Figure 1: Nemo system architecture.

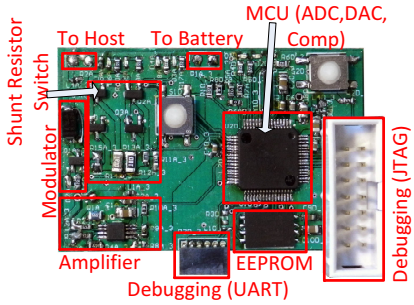

Figure 2: A prototype implementation of Nemo.

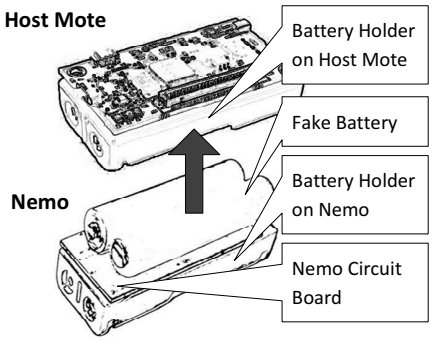

Figure 3: A possible Nemo packaging for easy installation on TelosB motes.

states, Nemo must achieve a dynamic range of 100,000:1 (from 2 uA to 200 mA). The switching between different power states of the electrical components can lead to sudden current consumption spikes [11]. Our experiment shows that such spikes typically occur within a short duration ranging from 200 us to 400 us on TelosB motes. Nemo must capture such dynamic transitions because they provide important temporal variation of node power consumption, which can be used, for example, for system debugging and fault diagnosis. As a result, the minimum sampling rate needs to be at least 5 KHz.

Noninvasiveness. Nemo is designed to be noninvasive to the host sensor node in two aspects. First, hardware wiring or soldering should be minimized or completely avoided when connecting Nemo to the host node. Nemo is a plug & play component that can be easily installed on a variety of different existing sensor network platforms. This is particularly important for aftermarket sensor platforms without accessible I/O ports, e.g., sealed sensor nodes [10] or customized nodes without I/O expansion ports [7]. Second, Nemo must be a stand alone device that does not rely on onboard resources including RAM and CPU during run time. This ensures that the performance of the host nodes is not compromised in the presence of power metering, improving the fidelity of power measurement.

Real-time host-meter communication. A key advantage offered by *in-situ* power meters is that the measurement results can be fed back to the host node for real-time power monitoring and analysis, which enables run-time adaptation of a sensor network system. For example, when an energy-aware routing protocol is adopted, real-time power consumption data is crucial for making network-wide routing decisions. Furthermore, the host-meter communication also allows the host node to dynamically configure Nemo, e.g., shutting it down for energy conservation when real-time power monitoring is not needed.

Low power consumption. Low power consumption is another critical requirement for a power meter due to the limited energy resource of sensor nodes. In particular, many system issues are difficult to diagnose without long term power monitoring at a high sampling rate. However, achieving low power consumption and high measurement fidelity at the same time is challenging.

3.2 System Architecture

Fig. 1 illustrates the system architecture of Nemo, which consists of a microcontroller (MCU), a current measurement circuit, and a voltage modulator. A prototype implementation of Nemo is shown in Fig. 2. The measurement circuit measures the current draw of the host node, and sends the measurement to the MCU. The voltage modulator, which is directly connected to an I/O pin of the MCU, modulates the voltage on the power line to transmit data to the host node. A battery pack is connected to the meter through which the host sensor node is powered. The power and ground wires are the only physical connection between the meter and the host node.

The MCU inside the meter processes measurement data, and stores the data into an EEPROM on the meter. The MCU also runs the host-meter communication protocol. The current that passes through the measurement circuit creates a small voltage over the shunt resistor, which is proportional to the current intensity. The voltage is then amplified by a differential amplifier. After amplification, the voltage signal is first digitalized by the on-chip low power 12-bit ADC in the MCU, and then converted to the current intensity.

The key difference between Nemo's current measurement circuit and traditional designs [15] is the shunt resistor. Typical current sensing design uses a single shunt resistor and a low resolution ADC, which cannot achieve wide dynamic current range and high sampling rate at the same time. In contrast, Nemo adopts a series of shunt resistors called *shunt resistor switch* whose resistance can be dynamically adjusted according to the required dynamic range. This design provides wide dynamic range without requiring expensive and power-hungry high-resolution ADCs.

During sleep state, the current draw of the sensor node is small and does not change drastically. In our design, Nemo can automatically enter sleep state when the host node falls asleep. A comparator on Nemo acts as a host wake-up detector which notifies the MCU of Nemo when the host wakes up. This design offers good energy saving without compromising measurement fidelity.

A key feature of Nemo is that the meter can communicate with the host node without dedicated data wires[1]. This is achieved by a novel technique called current/voltage modulation, in which the current load and the supply voltage are modulated to carry information. Specifically, when the host node transmits data to Nemo, it modulates its own current draw to encode data bits. On the reversed link where the data is transmitted from Nemo to the host node, the supply voltage of the host is modulated by Nemo to

[1]Nemo also supports conventional I/O or bus communications.

encode data bits. Our design achieves high link throughput while incurring low computational overhead. As shown in Section 7, although this technique introduces minor supply voltage fluctuation, it has no impact on the performance of host nodes.

The power line modulation techniques remove the need for any data wires between Nemo and the host node. As a result, with proper packaging, Nemo can be easily installed on almost any existing mote platforms without hard wiring or soldering. Fig. 3 illustrates a possible Nemo packaging[2] for easy installation on existing mote platforms with a battery pack.

4. HIGH FIDELITY CURRENT MEASUREMENT

The core of Nemo measurement subsystem is the current sensing circuit. A typical current sensing circuit consists of a shunt resistor, preamplifiers and a digital convertor. Two popular design choices that can achieve wide dynamic range are adjusting the amplification rate or using high resolution digital converters. However, the former requires sophisticated, power-hungry noise reduction circuits to achieve the desirable dynamic range while the latter incurs expensive, high power consumption convertors. As a result, limited by the low power consumption budget, neither of these two approaches can achieve favorable dynamic range.

4.1 Shunt Resistor Switch

Nemo adopts a technique used in auto-ranging digital multimeters to satisfy the fidelity requirement without incurring high power consumption. It features a series of shunt resistors which we refer to as *shunt resistor switch*. As illustrated in Fig. 4, the shunt resistor switch is composed of a series of resistors and electrically controlled switches. The resistance of the shunt resistor switch can be adjusted by shorting one or more resistors via switches. According to the ADC readings, a large (small) resistance is chosen when measuring small (large) current. With this design, both low and high currents can be accurately amplified to a proper voltage level for digitalization. As a result, a fixed preamplifier and a low resolution ADC can be adopted in the following subsequent stages of Nemo without compromising the measurement fidelity.

When the resistance of shunt resistor switch is high (i.e., measuring small current), a sudden current surge, which typically happens within tens of microseconds when host node switches its working mode, may cause a large voltage drop on the shunt resistor switch. This in turn leads to a significant supply voltage drop to the host node and even the malfunction of onboard components. If MCU only monitors the ADC readings, it cannot react to the sudden current surge promptly by adjusting the shunt resistor switch due to the long ADC sampling interval (>100 us). This issue is particularly critical when the host node wakes up from deep sleep, resulting in a sharp current increase up to four orders of magnitudes in several microseconds. We address this issue by using a comparator to generate an interrupt to MCU upon the sudden current increase. A comparator compares the voltage on its two inputs, i.e. non-inverting input and inverting input, and outputs a high or low voltage indicating which input has larger voltage. In our design, the non-inverting input is tethered to the output of the preamplifier. The inverting input is connected to the output of a DAC which provides a reference voltage. The DAC output is set according to the maximum allowable voltage drop on the shunt resistor switch. For example, if the maximum allowable voltage drop is 30 mV and the amplification rate is 50X, then a reference voltage of 1.5V is output by the DAC. A voltage higher than 1.5V triggers the comparator to generate an interrupt to the MCU, which immediately adjusts the resistance of the shunt resistor switch. We note that the delay before the actual resistance adjustment, typically shorter than 2us in our measurements, results in a short transient high voltage drop. However, due to the decoupling capacitors and inductors on the power loop of the host, the voltage drop on the shunt resistor switch is slowly built up, resulting in no significant impact on the supply voltage.

The voltage after pre-amplification is digitalized by a 12-bit ADC on the MCU, and then converted to the current intensity. The adjustment of shunt resistor switch creates a sudden voltage change at the inputs of the differential amplifier and ADC. Nemo pauses the ADC sampling for 5us after switching, which allows these components to settle and avoids generating erroneous measurement results. The ADC measurement results are stored in an EEPROM on Nemo. Since the ADC generates measurement results at a high rate, the EEPROM can be filled up quickly. Nemo uses a simple compression algorithm to reduce the volume of data. The MCU stores a new measurement result only if it differs significantly from the previously stored one. Our experiments show that when a difference threshold of 1.6% is chosen, Nemo can achieve a compressing ratio up to 0.6% on TelosB motes running a typical sense-and-send application. We carefully optimized the code of compression and were able to process the measurement sampled at 8 KHz on the 8MHz MSP430 MCU. When Nemo is connected to a PC to upload measurements, a sampling rate of 100 KHz can be achieved by disabling the compressing algorithm.

4.2 Sleep Management

Even though Nemo employs a low power design, it still consumes considerable amount of power in the active state. Sensor network applications often employ duty cycles to conserve energy. When the host node is asleep, its current consumption is almost constant in a short time window. Our experiment shows that the variance of sleep current consumption of TelosB motes is less than 0.5 uA in 5 s time windows. This constant current draw clearly offers an opportunity for Nemo to save energy via sleep scheduling.

In our design, Nemo automatically falls asleep when the host node enters sleep state, and periodically wakes up to conduct measurements. Since the current draw fluctuation of the host node in sleep is rather small, the reduced sampling rate does not cause degradation of measurement fidelity. Wake-up of host components usually causes significant surge of the current draw, which must be captured to ensure high measurement fidelity. To detect such events, Nemo utilizes a comparator which raises an interrupt and immediately wakes up Nemo to resume high frequency sampling when the current draw of the node exceeds a certain thresh-

[2]The dimension of the current prototype of Nemo is 3" by 4". The PCB board can be made smaller in future by removing debugging components, including JTAG ports, LEDs, and buttons, and easily fit into a 2-AA battery pack.

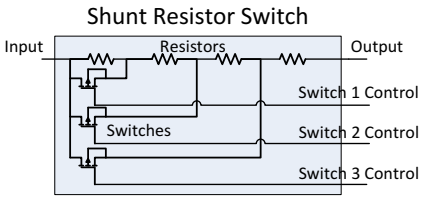

Figure 4: The resistance of the shunt resistor switch is adjusted by shorting one or more resistors.

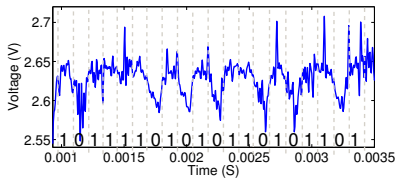

Figure 5: A voltage signal modulated at 8 Kbps carrying the data 0x0BD5AD.

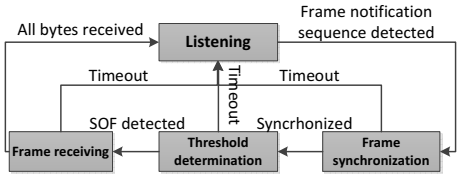

Figure 6: The receiving procedure state machine.

old. The threshold can be determined offline or measured in system initialization phase. During sleep, the MCU, ADC, and internal voltage reference are turned off to conserve energy. The amplifier, comparator, and DAC remains powered to detect the wake-up of the host node. The total current consumption of Nemo in sleep state is 150 uA.

5. HOST-METER COMMUNICATION

A key advantage of Nemo is that it enables a wire-free, plug & play installation on aftermarket sensor systems. However, this design choice rules out the possibility of using any dedicated on-board data wires of the host node. To implement the communication between host and meter, Nemo modulates voltage/current load of the power line, without using any dedicated communication wire. Specifically, the host node transmits data to the meter by modulating its own current draw. On the reversed link, the meter modulates the supply voltage of the host node to carry data. The two links work in half-duplex, sharing the same single power line.

5.1 Supply Voltage Modulation

Our basic idea of enabling the communication link from meter to host node is to encode information by modulating the amplitude of supply voltage while the host node decodes the information by measuring the voltage change. As a result, the measurement data can be transmitted from Nemo via a single power line. This approach is motivated by the fact that today's sensor network platforms can readily measure the supply voltage with on-MCU ADCs. In our design, a diode paralleled with an electrically controlled switch is used as the voltage modulator. As the diode causes a constant voltage drop, switching it on and off will generate a pulse signal over the amplitude of supply voltage. By controlling the switch of diode, the supply voltage of the host can be precisely modulated to carry information bits. However, a potential concern of this approach is that the fluctuation of the supply voltage caused by modulation may lead to malfunctioning of the host node. To address this issue, a low forward voltage drop Schottky diode is employed to create only a 100 mV voltage drop during modulation. In Section. 7.4, we show that such a small fluctuation has little impact on the performance of the host node. On the host node, the modulated signal is sampled by the ADC. The voltage samples are then decoded by a simple demodulating routine, as discussed in Section. 5.3.

5.2 Current Load Modulation

The idea of supply voltage modulation is not applicable to the communication link from host node to meter because the host node usually cannot vary the supply voltage. To realize the communication on the power line, the host node modulates its current draw, and the modulated current signal is then measured and decoded by the meter.

Controlled by the MCU, various electrical components on the host node can be turned on/off to create variation of the current draw. These components form simple but effective current modulators. Information can be encoded into the current draw patterns, which are measured by the ADC on the meter for receiving information. To achieve high communication bandwidth, the modulator must be toggled at a high frequency and generate sufficient current change. Commonly available on most sensor networks platforms, LEDs make perfect current modulators. They are usually directly connected to I/O pins of the MCU and thus can be switched at a high frequency. The current draw of a typical LED is several milliamp, which can generate sufficient current change during modulation while incurring low extra power consumption.

Other onboard components can also cause variations of current draw, leading to interference to data transmissions. To address this issue, the communication can be initialed at the end of active period in a duty cycle, when most onboard components fall asleep. The resulting low bandwidth is not a concern because the host node usually only sends short poll messages while most of the data is originated from the meter. On the receiving side, the meter measures the current consumption of the host and decodes the current modulated signal. As the meter samples the current consumption at a high frequency, it ensures a sufficient modulation rate on the link.

We note that on nodes equipped with energy harvesting devices such as solar panels, the supply voltage may not be constant. This does not affect the current/voltage-modulated communication performance because each transmission only lasts for a very short time period (several hundred ms), during which the output voltages of most energy harvesting devices remain largely constant.

5.3 The Communication Protocol

We now discuss the host-meter communication protocol in detail. The protocol implements half-duplex communication between host node and Nemo. The half-duplex mode is sufficient because most traffic occurs in the direction from the Nemo to the host when Nemo responds to the host's queries and sends back the power consumption data. The main design objective of the protocol is to achieve high throughput, which ensures system energy efficiency even when the host needs to frequently query Nemo.

The communication frame consists of a header, the payload and a checksum byte, as shown in Tab. 1. The Start of Frame (SOF) field which is always 0x5a is used to notify the receiver of the frame beginning. The one-byte command (CMD) field indicates the purpose of this frame. The two-

SOF	CMD	LEN	Timestamp	Payload	Checksum
1 byte	1 byte	2 bytes	4 bytes	N bytes	1 byte

Table 1: Structure of the frame

byte frame length field (LEN) indicates the total length of the frame in bytes. The timestamp field is filled upon the actual transmission of the frame. It can be used for time synchronization between the host node and the meter. The checksum field allows the receiver to check the correctness of the received frame.

The frame is modulated using binary ASK which is similar to the Universal Asynchronous Receiving/ Transmitting (UART) protocol. We choose binary ASK mainly because it can be easily implemented and incurs little computational overhead on the host node. The modulation rate can be set to 2Kbps, 4Kbps, 8Kbps or 16Kbps, according to the link quality. A preamble consisting of a series of alternating symbols is always sent before each frame. The preamble has two parts that are sent sequentially: the frame notification sequence and the receiver training sequence. The former is always modulated at 4 Kbps to notify the receiver of the incoming of a frame. The latter, which is modulated at the same rate as the frame, provides information for the receiver to learn the receiving parameters. Fig. 5 depicts the waveform of a voltage modulated signal captured by an oscilloscope. The current modulated signal has a similar shape and thus is not shown here.

On the receiver, a state machine controls the receiving procedure. The state machine has four states: listening, frame synchronization, threshold determination, and frame receiving. The receiver stays in listening state after powering up, seeking for the frame notification sequence in the preamble. After seeing a frame notification sequence, the receiver synchronizes to the modulated signal, which is important for achieving high SNR. Frame synchronization is performed by measuring the modulation rate and the optimal sampling timing from the receiver training sequence. The modulation rate is calculated by the receiver from the measured symbol period. After frame synchronization, the receiver determines decision threshold by measuring and averaging the signal amplitudes of 10 consecutive symbols. After the decision threshold is measured, the type of the symbol can be determined by comparing the signal amplitude against the decision threshold. After all the bytes are received, the receiver goes back to listening state, searching for a new preamble. Fig. 6 depicts the whole receiving process. Three types of frames are used in communication: configuration, data request, and response. We omit the details of the frame format here due to the space limitation.

5.4 Discussion

The power-line modulation techniques described in this section enable the bidirectional communication between the host node and Nemo. To avoid high overhead on the host node, Nemo adopts a poll-response communication scheme in which the communication is always initiated by the host. Moreover, to achieve satisfactory link quality, the communication is only initiated when the host node is in a stable power state (e.g., at the end of active period in a duty cycle when most of the host components enter sleep state). Because of these requirements, Nemo and the host cannot maintain a "always-on" communication link. However, this limitation does not lead to performance degradation of power metering because the data transmitted between host and Nemo is not delay-sensitive. First, Nemo can be configured by the host at any time without degrading the measurement performance. Second, power measurement results are time-stamped and buffered on Nemo, which can be queried by the host later. The power measurement results of the current duty-cycle can be queried at the end of the duty-cycle, which incurs little delay. The delay between the host issues a query and the communication may occur is typically small, e.g., in the order of a duty cycle. Such short delay does not affect the host's capability of real-time adaption (e.g., adjusting its duty cycle) based on the feedback from Nemo.

```
interface host_meter_comm {
  // Interface control
  command error_t enable_rx_comm();
  command error_t disable_rx_comm();

  // Transmission and receiving
  command error_t transmit(NemoCom * frame, uint16_t len);
  async event void receive(NemoCom * frame, uint16_t len);
}
```

Figure 7: TinyOS API for host-meter communication.

6. IMPLEMENTATION

We have implemented a prototype of Nemo. The dimension of the implementation is 3" by 4". The size of the PCB board can be further reduced in future generations (e.g., by removing debugging components including JTAG ports, LEDs, and buttons) and easily fit into a 2-AA battery pack, as shown in Fig. 3. This would allow a wire-free installation of Nemo on any sensor platforms that have a 2-AA battery pack. A TI MSP430F2618 ultra-low power MCU is adopted on Nemo, which has 96KB Flash ROM, 8KB RAM, and on-chip peripherals such as ADC, DAC and comparator. The abundant on-chip resources enable us to use a single chip to implement various tasks, eliminating the need of dedicated ICs such as ADC. This design reduces the cost and power consumption of the system. The MCU has a maximum clock rate of 16 MHz, which is deliberately downclocked to 8 MHz to conserve energy. The shunt resistor switch is composed of 5 resistors (0.1 Ohm, 1 Ohm, 10 Ohm, 100 Ohm and 470 Ohm) and 4 MOSFETs as switches. Additional resistors and switches can be added to further extend the dynamic range. We choose TI OPA2333 as the pre-amplifier, which offers sufficient bandwidth, low offset error, and low quiescent current consumption (17 uA). A Winbond 8 MByte high-speed SPI Flash chip is adopted to store measurement results.

The firmware of Nemo is mainly implemented in C. Some performance critical code such as ADC sampling and data compressing is written in assembly. The implementation of measurement control, compression, and host-meter communication protocol has a footprint of 8KB and uses 5 KB memory. We define an interface in TinyOS to support the host-meter communication on host nodes, as shown in Fig. 7. Our implementation of the host-side protocol has a footprint of 1 KB and uses 250 Bytes RAM. We calibrated Nemo with an Aglient 34410A benchtop digital multimeter (DMM). The calibration data is loaded into MCU.

7. PERFORMANCE EVALUATION

Section 7.1 – Section 7.4 evaluate the performance of Nemo. Section 7.5 presents a case study of using Nemo to track dynamic sleep power consumption of motes under different temperatures. Lastly, we compare the performance of Nemo with two state-of-the-art power meters in Section 7.6.

7.1 Measurement Fidelity

We evaluate the measurement fidelity including dynamic range, resolution, and measurement accuracy in this section. To measure dynamic range, resolution and static accuracy, potentiometers are used to generate current load ranging from 0.1 uA to 200mA. In the experiment of dynamic measurement accuracy, a TelosB mote is used instead. An Aglient 34410A benchtop digital multi-meter is connected in series with the potentiometer or TelosB mote to measure the ground-truth current. The current measurement of the digital multi-meter is transmitted to a desktop PC via Ethernet at 10 KHz rate. Nemo transmits its raw ADC readings and the shunt resistor setting to the same PC via the UART debugging port. During the experiment, we slowly vary the current from 0.1 uA to 200 mA by changing the resistance of the potentiometer. The measurements from both the meter and the digital multi-meter are recorded by the PC for data analysis.

7.1.1 Dynamic Range and Resolution

Dynamic range and resolution are important performance measures of a power meter. They give the maximum range and the minimum quanta of the current that the meter can accurately measure. Fig. 9(a) depicts the relationship between input current and output raw ADC readings. It can be seen that the ADC reading first linearly increases with the input current and then suddenly drops to a lower level. This pattern is repeated throughout the whole input current range. The sudden drop is caused by the resistance adjustment of the shunt resistor switch, when the voltage drop on the shunt resistor switch is larger than 20 mV. The dynamic range of Nemo is the linear region of the input-output curve, which ranges from 0.8 uA to 202 mA, corresponding to a dynamic range of over 250,000:1.

The resolution of Nemo is the difference between input current of two adjacent ADC readings, which can be also interpreted as the slope of the input-output curve of Nemo. Fig. 9(b) shows the resolution computed from the slope of the curve in Fig. 9(a). We notice that the resolution is not constant and increases each time when a resistance adjustment occurs. The minimum and maximum resolutions are 0.013 uA and 48 uA, respectively. This variable resolution is resulted from the dynamic input current ranges of each shunt resistor switch setting. Since the digitalization resolution of Nemo is always 12 bits, the measurement resolution increases when a higher measurement range is chosen.

The results in this section show that Nemo has satisfactory dynamic range and measurement resolution. As shown in Section 7.6, Nemo significantly outperforms state-of-the-art sensor network power meters in both metrics. These features make Nemo ideal for measuring the power consumption of sensor networks under a wide range of applications and operating conditions. In particular, as many sensor networks operate under low duty cycles and stay in sleep state (with just a few uA current consumption) most of the time, the fine measurement resolution (∼0.01uA) of Nemo enables accurate assessment of the system lifetime.

7.1.2 Measurement Accuracy

In this set of experiments, we measure the accuracy of Nemo. Fig. 10(a) shows the measurement errors across the whole dynamic range. A CDF of the errors is given in Fig. 10(b). The error is computed as the ratio of the absolute measurement error to the ground-truth data. It can be seen from Fig. 10(a) that, the error has multiple peaks across the dynamic range. They are mainly resulted from the quantization error of the ADC after shunt resistor switch is adjusted. The maximum error, 8.3%, occurs at the lowest end of the dynamic range. When the input current is larger than 10 uA, most of the errors fall below 2%. As can be seen from the CDF in Fig. 10(b), 90% of the errors are below 3%, and the mean error is only 1.34%.

We also examine the measurement error of Nemo in real deployment scenarios when a TelosB mote is attached. The TelosB mote runs a typical sense-and-send application in this experiment. Due to the significant current variation of mote, the measurements of Nemo and digital multi-meter need to be synchronized in order to compare the accuracy. At the beginning of this experiment, a pulse signal is output to an I/O pin of the host mote, which triggers Nemo and multi-meter to begin their measurements. Fig. 11(a) shows the current measurement containing a wake-up event of the mote. It can be seen that the current profile generated by the mote is highly dynamic, containing sharp current increases and decreases over 2 orders of magnitude. However, even with such significant dynamics, Nemo can track the change of the current closely. Fig. 11(b) shows the CDF of the measurement errors. Over 90% of the errors fall below 5%. The mean error is only 2.09%.

7.2 Host-Meter Communication

We first evaluate the BER of the host-meter link in both directions under different modulation rates and frame lengths. A TelosB mote is used as the host mote. We modified the sense-and-send application used in previous experiments, so that the mote transmits a frame to the meter when a button is pressed. The meter transmits back the same frame. The payload contains 3000 bytes of random numbers. The modulation rates tested in this experiment are 2Kbps, 4Kbps, 8Kbps, and 16Kbps. For each modulation rate, 100 runs of the experiments are conducted.

We measure the BER of each modulation rate with different frame sizes. Fig. 12 shows the BER of both links. Note that we did not observe any bit error on the host to meter link at 2 Kbps rate and thus omit the result here. We can see that, on both links, the frame length has a substantial impact on the BER of the link. For smaller frame sizes (<400 Bytes), no bit error occurred. However, when the frame size increases to a certain level, BER starts to grow rapidly. This critical frame length varies with different modulation rates, but is generally larger when a lower modulation rate is adopted. The reason for this phenomenon is that the synchronization between frame and the receiver gradually deteriorates after the initial frame synchronization, due to the different clock rates of transmitter and receiver. As a result, the SNR of the receiver gradually decreases and bit errors appear after the SNR drops below a certain threshold. We note that the bit errors resulted from large frame size can

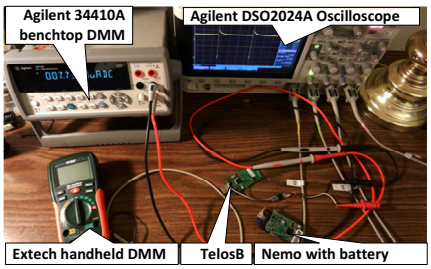

Figure 8: Experimental set up of performance evaluation.

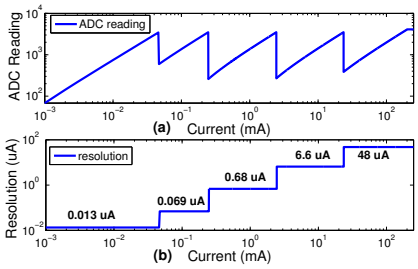

Figure 9: Dynamic range and resolution of Nemo.

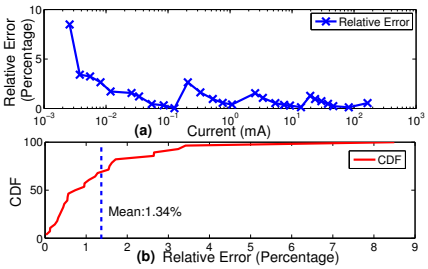

Figure 10: Measurement error of Nemo.

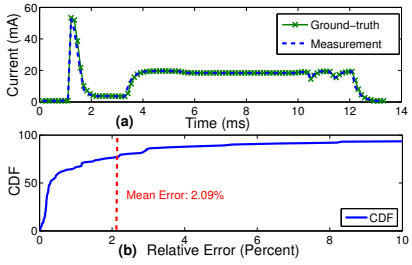

Figure 11: Dynamic accuracy of Nemo.

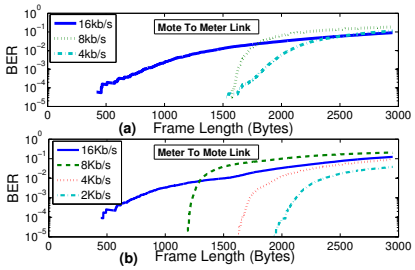

Figure 12: BER vs datarate vs. frame length.

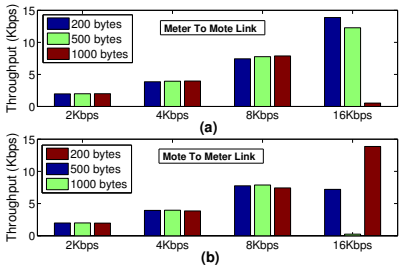

Figure 13: Host-meter throughput vs. data rate vs. frame length.

be mitigated by maintaining clock synchronization between host mote and meter. However, this is left for future work.

In the second experiment, we examine the throughput of the host-meter communication. The mote continuously transmits 100 frames to the meter, who replies by transmitting the same frames back to the host mote. The frames with incorrect checksums are discarded. We test three frame size settings: 200, 500, and 1000 bytes. For each modulation rate and frame size combination, 10 rounds of experiments are conducted.

Fig. 13 shows the resulted throughput. We can see that, for the 2Kbps, 4Kbps and 8Kbps modulation rates, the throughputs are very close to the corresponding modulation rates, although shorter frames lead to slightly lower throughput due to the higher link overhead. For the 16Kbps modulation rate, the throughput under different frame size settings shows large variations. For example, on the meter to host link, the three frame settings can achieve a throughput of 13.87 Kbps, 7.76 Kbps and 0.25 Kbps, respectively. This observation suggests that frame segmentation is needed for transmitting data chunks larger than 400 bytes using 16Kbps.

In summary, the results in this section show that the Nemo and host can achieve robust communication performance. The high communication throughput allows Nemo to continuously track the system power consumption and feed back to the host in real-time. It also leads to low overhead to the host, as we show in next subsection.

7.3 Power Consumption and Overhead

In this section, we evaluate Nemo's power consumption and the overhead of host-meter communication. Nemo is connected to a host mote running a sense-and-send application. We use the Aglient 34410A benchtop to measure the total current consumption, and use Nemo to measure the current consumption of the host mote. The difference is the current consumption of Nemo. Figure. 16 shows the dynamic current consumption of Nemo when host mote varies its working states. The host mote wakes up around 0.8 ms, right before the appearance of the high current spike. Nemo wakes up immediately after the host mote and its current consumption increases sharply from the sleep level (150 uA) to the active level (4.6 mA). During the active state, Nemo maintains a stable current consumption. The minor spikes of Nemo's current consumption are mainly resulted from measurement errors. As discussed in Section 4, to conserve energy, Nemo can automatically enter the sleep state after the host falls asleep. As shown in Fig. 16, after the host mote goes back to sleep, Nemo remains active for another 1 ms. This is due to the fact that the sleep state of host mote is confirmed by Nemo only when the current draw is constantly below a threshold longer than 1 ms. We note that, when the host does not need to monitor its sleep current, it can command Nemo to completely shut down before falling asleep, which will further reduce the current consumption by 150 uA.

As shown in Section 7.2, due to the low SNR of power line, the host-meter communication can only achieve a maximum throughput of 14 Kbps, which is slower than using I/O pins or onboard buses. This incurs additional energy overhead since data transmissions will take longer to finish. We now evaluate this overhead in a typical sense-and-send application. The mote wakes up every 10 s and remains active for 10 ms, resulting in a 0.1% duty-cycle. When the host is active, Nemo continuously measures its power consumption at 8KHz sampling rate. When the host is asleep, Nemo takes a measurement once every 2 seconds because the sleep power consumption remains constant within a short period of time. During the experiment, Nemo maintains a data buffer of 4K bytes and transmits the buffered data to the host once the buffer is full. We are interested in the ratio of the time it takes Nemo transmit the 4K data to the host and the time it takes to collect the measurement. As both the host and Nemo must remain active for transferring the measurement data, this ratio quantifies the overhead of host-meter communication. We conduct the experiment for 10 runs, and on average, it takes Nemo 389 s to fill the 4K buffer and 2.32 s to transmit the data to host mote using 16 Kbps

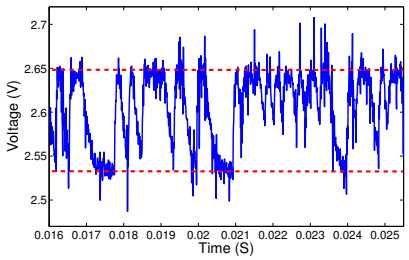

Figure 14: Supply voltage fluctuation caused by modulation.

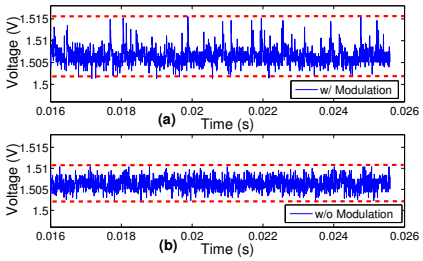

Figure 15: Variation of the reference voltage on ADC w/ and w/o modulation.

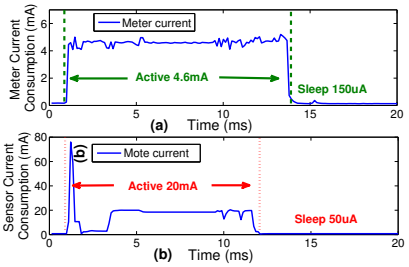

Figure 16: Dynamic current consumption of Nemo and host mote running a sense-and-send application.

modulation rate, resulting in a 0.6% overhead. We note that this overhead is likely even lower in practice because the host may not need all the power measurement data.

7.4 Impact on Host Mote

Nemo communicates with the host by modulating the supply voltage. A potential concern is that the resulted fluctuation of the supply voltage may cause some components on the host to malfunction. In this experiment, we study the impact of the supply voltage modulation on the operation of the host mote.

We first measure the voltage fluctuation experienced by the host under the voltage modulation. We use an oscilloscope to measure the supply voltage during the host-meter communication. Fig. 14 depicts the waveform on the power line during supply voltage modulation. The modulation causes a maximum fluctuation of 130 mV. For digital components that are powered by switching regulators, a 130 mV fluctuation is common during their normal operations. Due to the switching nature, these regulators often cause supply voltage fluctuations called ripples. Typical ripples of boost regulators with 3.3 V output are 100 mV [14]. For example, the boost regulator MAX1724 used in iCount [5] has a ripple of at least 75mv when attached to the sensor node. As a result, the 130 mV supply voltage fluctuation introduced by the modulation will not cause problems to the normal operation of most digital components.

Analog components like sensors are usually sensitive to power supply noise. In particular, the voltage fluctuations may have impact on the conversion accuracy of ADC[3] even when there is small variation of the reference voltage. The MCU of TelosB mote exposes the internal reference voltage on an I/O pin, whose stability directly reflects the ADC performance. We measure the reference voltage using oscilloscope during a meter to host transmission. For comparison, the voltage without ongoing voltage modulation is also measured. Fig. 15 shows the results. We can see that the modulation causes a 4 mV peak-to-peak increase of the reference voltage, which is mainly resulted from the occasional minor voltage spikes. This variation will lead to a maximum ADC conversion error of mere 0.27% (4 mV/ 1.5 V). The impact of such a small ADC error on sensor readings is negligible.

[3]All sensors are connected to the ADC for digitalizing the sensor measurements.

7.5 Case Study

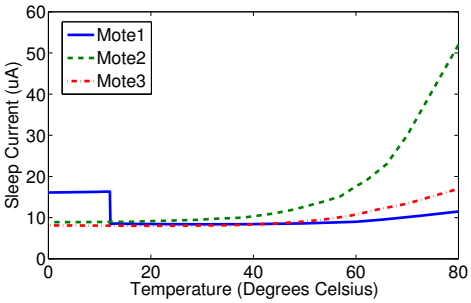

Figure 17: Sleep current of TelosB vs. temperature

With the high measurement fidelity and robust host-meter communication performance, Nemo can enable a wide range of sensor network applications to track their power consumption in real-time. This section presents a case study to demonstrate the benefits of power metering with Nemo. In the case study, we use Nemo to track the sleep current consumption of TelosB motes across different temperatures. It is well known that heat can lead to leakage power of electronic components [6]. However, to our best knowledge, the scale of such heat-induced power consumption dynamics and how it affects sensor network lifetime has not been systematically studied.

We install the *NULL* application from the TinyOS distribution on three TelosB motes. The *NULL* application simply enters the sleep mode after booting, which results in a very low power consumption. We attach Nemo to the three motes and also use a thermal probe to measure the surface temperature of their circuit boards. The probe is connected to a digital multi-meter through which the temperature readings can be logged. The motes are initially placed on top of an electric heater. When their temperature reaches 80 °C, they are moved outdoor where the temperature is about 0 °C. We log the temperature of the motes as they cool down. Fig. 17 shows the sleep current measurement of three motes.

We can see from Fig. 17 that the three motes have very different current consumption profiles. Their current consumption generally increases with the temperature. This is expected as the current leakage of most electric components increases with temperature. However, the slopes of current growth are significantly different. The three motes have similar sleep current consumption at room temperature (8, 8, and 9 uA). However, when they are heated to 80 °C, they consume 11.5, 52, and 17 uA, respectively, resulting in up to 6 times of difference. For a TelosB mote operating at 0.1% duty cycle, an increase of sleep current consumption

	Nemo	iCount	SPOT
Dynamic range	250,000:1 (0.8 uA - 202 mA)	100,000:1	45,000:1 (1 uA - 45 mA)
Resolution	0.013 uA (<50 uA), 0.068 uA (50 uA-250 uA), 0.68 uA (250 uA-2.5 mA), 6.6 uA (2.5 mA -25 mA), 48 uA (>25 mA))	varies w/ sampling rate 10 uA (8 Hz), 100 uA (80 Hz), 1mA (800 Hz)	varies w/ sampling rate 10 uA(220 Hz), 100 uA (2200 Hz), 1 mA (22 KHz)
Sampling rate	8 KHz (w/ compression), 100 KHz (w/o compression)	66 KHz max 80 Hz @ 100 uA resolution	N/A
Measurement error	average 1.34%, max 8%	max ±20%	average 3%
Sleep power measurement	Yes	No	Yes
Power consumption	154 uA (0.1% duty-cycle) 195 uA (1% duty-cycle)	1% of host current plus energy loss on regulator (>10%)	1.7 mA
Host CPU overhead	0.6% w/ comm., otherwise none	13% at 8KHz sampling rate	N/A
Host resource usage	none	Timer, one I/O pin	I2C bus, multiple I/O pins
Ease of installation	very easy, wire-free plug n' play	soldering of wire to host mote	soldering of board onto host; extra 5.5V power supply

Table 2: Comparison between Nemo, iCount and SPOT.

from 8 uA to 52 uA will shorten the mote lifetime by 61.1%. Another interesting observation is that the current of Mote 1 suddenly increases by 100% when cooled to 12 °C. The same experiment was repeated for a number of times, and the same phenomenon was always observed. We suspect that this has to do with the circuit design and thermal characteristics of some components on this mote. However, a detailed investigation is left for future work.

In summary, our results show that the sleep current of motes varies significantly with environmental temperature. We believe this finding has important implications for the design and deployment of sensor network applications. As many sensor networks operate under extremely low duty-cycles, their lifetime is often determined by the sleep power consumption. Existing work often assumes that low-power motes consume constant power during sleep. While this assumption may hold in static environments, when deployed in the field, motes may yield significant variations and dynamics in their sleep power, largely due to environmental factors like temperature. Nemo can track the resulted power consumption dynamics and energy imbalance in real-time, and enable the host mote to make informed decisions on runtime adaptation for prolonging network lifetime.

We also note that, the power consumption of Nemo is affected by the temperature, which potentially introduces uncertainty in the power measurement. We have carefully designed the component layout and ensured sufficient shielding in the circuit board of Nemo, which are known effective to mitigate temperate-induced dynamics. Moreover, in several critical circuit sections, we choose components (e.g., industry-grade components) that can tolerate wider temperature range. These measures prove effective, as no significant change of the power consumption of Nemo is observed when it experienced severe temperature variations.

7.6 Comparison with iCount and SPOT

In this section, we compare Nemo with two state-of-the-art sensor network power meter systems iCount [5] and SPOT [11]. Since we do not have access to the iCount and SPOT hardware, the performance data of the two systems are obtained from two papers [5] [11]. Tab. 2 summarizes the comparison of the three power metering systems. Note we focus on the performance of power measurement instead of energy measurement.

As shown in Tab. 2, Nemo outperforms both iCount and SPOT in terms of dynamic range and resolution. In particular, Nemo's dynamic measurement range is 2.5X and 7X the range of iCount and SPOT, respectively. It's important to note that there exists a fundamental tradeoff between the resolution and sampling rate of iCount and SPOT. Both of them convert current to frequency for achieving wide dynamic ranges. A high resolution requires a long sampling time to collect sufficient number of pulses, which inevitably leads to a low sampling rate. This is particularly serious for iCount whose highest counting frequency is only around 100 KHz. We now use an example to illustrate the issue. The Max1724 regulator adopted by iCount has a 800 Hz oscillating frequency when the output current is 1 mA [5]. Assume that the regulator has a linear frequency-current relationship and zero offset. If a resolution of 100 uA is needed, the counter must gather at least 10 pulses (1 mA/ 100 uA) during a sampling interval to ensure a 100 uA resolution. However, since the oscillating frequency is only 800 Hz, this would lead to a sampling rate of only 80 Hz (800/ 10). As a result, such approaches cannot achieve high resolution and sampling rate at the same time.

Both Nemo and SPOT can achieve high measurement accuracy while the error of iCount is as high as 20%. Moreover, iCount uses host CPU for frequency counting and hence cannot measure the sleep power consumption of the host. As shown in Section 7.5, real-time measurement of sleep power consumption is critical for estimating the system lifetime for sensor networks deployed in dynamic environments. The power consumption of Nemo can be adjusted based on sleep duty-cycles while iCount and SPOT consumes fixed and much higher power. Compared with iCount and SPOT, Nemo poses negligible CPU overhead for the host node and consumes no host resources. Lastly, installation of both iCount and SPOT normally requires soldering between the meter and host. In contrast, Nemo is specially designed for easy, noninvasive installation on existing sensor network platforms, with the power line being the only physical connection between the host and Nemo.

8. DISCUSSION

Nemo is particularly suitable for *in-situ* power monitoring of sensor network systems that must operate for long periods of time in dynamic environments. Representative examples include habitat monitoring [19], civil structure health monitoring [4], etc. These sensor network systems often need

real-time feedback on power consumption to adapt their working modes and duty cycles in response to environmental dynamics.

Nemo has a non-negligible power consumption (about 150 uA) which is an inevitable overhead for its high fidelity and low cost. On the other hand, many applications do not always need to be monitored at the highest fidelity. Nemo can be configured to work in a standby mode most of the time in which Nemo consumes very little power (a few uW). Nemo can also be duty-cycled and dynamically configured by the host node at run time to adapt to the different accuracy and power requirements. In the current design, many components (e.g., ADC, DAC and comparator) are integrated in MCU, which achieves a good balance among power consumption, cost, system complexity, and design flexibility. By adopting ultra-low power components (DAC, comparator, etc.), the sleep power consumption of Nemo could be further reduced to around 50 uA , although this will likely increase the cost of Nemo (by $20 - $50) and introduce extra complexity to the implementation.

9. CONCLUSION AND FUTURE WORK

This paper presents *Nemo* – a practical *in-situ* power metering system for wireless sensor networks. Nemo is based on a noninvasive, plug & play design that allows it to be easily installed on existing sensor platforms without physical wiring or soldering. Using only the power line, Nemo implements real-time, high-speed communication with the host node by modulating the current load and the supply voltage to transmit information. Nemo achieves a wide dynamic current range and high measurement accuracy based on a new circuit design called *shunt resistor switch* that can dynamically adjust the resistance of shunt resistors based on the current load. Nemo has a dynamic measurement range of 250,000:1 while only incurring an average measurement error of 1.34%. We also present a case study to demonstrate the benefits of high-fidelity *in-situ* power measurement using Nemo. We show that Nemo is able to track the highly dynamic sleep current consumption of motes.

In the future, we will integrate Nemo with Quanto [8], which is a component-level energy tracking algorithm for sensor networks. The high-fidelity power measurement of Nemo will enable Quanto to profile energy consumption of system components and programmer-defined activities in fine granularity. We believe that the power line modulation techniques adopted by Nemo can be applied to a more general class of embedded and mobile devices. For instance, there is a growing need for real-time energy usage profiling on smartphone systems. However, the battery meter of current smartphones only provides low-frequency (about 1 Hz) energy readings to the OS. Leveraging the power line modulation techniques of Nemo, a specially packaged power meter may be attached to the original phone battery to provide high-frequency energy sensing. The measurement results can be transmitted back to the system via voltage modulation.

10. ACKNOWLEDGEMENT

The authors thank the shepherd Dr. Adam Wolisz and anonymous reviewers for providing valuable feedbacks to this paper. This work was supported in part by U.S. National Science Foundation under grants OIA-1125163 and CNS-0954039 (CAREER).

11. REFERENCES

[1] Corssbow telosb production web page. `http://bullseye.xbow.com:81/Products/productdetails.aspx?sid=252`.

[2] Ds2438 production web page. `http://www.maxim-ic.com/datasheet/index.mvp/id/2919`.

[3] Imote2 documents website. `http://wsn.cse.wustl.edu/index.php?title=Imote2_Documents`.

[4] E. Clayton, B.-H. Koh, G. Xing, C.-L. Fok, S. Dyke, and C. Lu. Damage detection and correlation-based localization using wireless mote sensors. In *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, pages 304 –309, june 2005.

[5] P. Dutta, M. Feldmeier, J. Paradiso, and D. Culler. Energy metering for free: Augmenting switching regulators for real-time monitoring. In *Proceedings of the 7th international conference on Information processing in sensor networks*, IPSN '08, pages 283–294, Washington, DC, USA, 2008. IEEE Computer Society.

[6] F. Fallah and M. Pedram. Standby and active leakage current control and minimization in cmos vlsi circuits. *IEICE Transactions*, 88-C(4):509–519, 2005.

[7] C. Fay, S. Anastasova, C. Slater, S. Buda, R. Shepherd, B. Corcoran, N. O'Connor, G. Wallace, A. Radu, and D. Diamond. Wireless ion-selective electrode autonomous sensing system. *Sensors Journal, IEEE*, 11(10):2374 –2382, oct. 2011.

[8] R. Fonseca, P. Dutta, P. Levis, and I. Stoica. Quanto: tracking energy in networked embedded systems. In *Proceedings of the 8th USENIX conference on Operating systems design and implementation*, OSDI'08, pages 323–338, Berkeley, CA, USA, 2008. USENIX Association.

[9] R. N. Handcock, D. L. Swain, G. J. Bishop-Hurley, K. P. Patison, T. Wark, P. Valencia, P. Corke, and C. J. OąŕNeill. Monitoring animal behaviour and environmental interactions using wireless sensor networks, gps collars and satellite remote sensing. *Sensors*, 9(5):3586–3603, 2009.

[10] J. Hayes, S. Beirne, K.-T. Lau, and D. Diamond. Evaluation of a low cost wireless chemical sensor network for environmental monitoring. In *Sensors, 2008 IEEE*, pages 530 –533, oct. 2008.

[11] X. Jiang, P. Dutta, D. Culler, and I. Stoica. Micro power meter for energy monitoring of wireless sensor networks at scale. In *Proceedings of the 6th international conference on Information processing in sensor networks*, IPSN '07, pages 186–195, New York, NY, USA, 2007. ACM.

[12] X. Jiang, J. Taneja, J. Ortiz, A. Tavakoli, P. Dutta, J. Jeong, D. Culler, P. Levis, and S. Shenker. An architecture for energy management in wireless sensor networks. *SIGBED Rev.*, 4(3):31–36, July 2007.

[13] G. Lu, D. De, M. Xu, W.-Z. Song, and J. Cao. Telosw: Enabling ultra-low power wake-on sensor network. In *INSS*, 2010.

[14] M. S. Mike Wens. *Design and Implementation of Fully-Integrated Inductive DC-DC Converters in Standard CMOS*. Springer, 2011.

[15] A. Rice and S. Hay. Decomposing power measurements for mobile devices. In *Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on*, pages 70 –78, 29 2010-april 2 2010.

[16] V. Shnayder, M. Hempstead, B.-r. Chen, G. W. Allen, and M. Welsh. Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 188–200, New York, NY, USA, 2004. ACM.

[17] T. Stathopoulos, D. McIntire, and W. J. Kaiser. The energy endoscope: Real-time detailed energy accounting for

wireless sensor nodes. In *Proceedings of the 7th international conference on Information processing in sensor networks*, IPSN '08, pages 383–394, Washington, DC, USA, 2008. IEEE Computer Society.

[18] B. S.-H. C. Sukwon Choi, Hayun Hwang. Hardware-assisted energy monitoring architecture for micro sensor nodes. *Journal of System Architecture, Elsevier*, 58:73–85, 2012.

[19] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler. An analysis of a large scale habitat monitoring application. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 214–226, New York, NY, USA, 2004. ACM.

[20] M. J. Thomas Trathnigg and R. Weiss. A low-cost energy measurement setup and improving the accuracy of energy simulators for wireless sensor networks. In *REALWSN*, 2008.

[21] Y.-M. Wang. Towards dependable home networking: an experience report. In *DSN*, 2000.

[22] Z. Zheng Peng; Zhong Zhou ; Jun-Hong Cui ; Shi. Aqua-net: An underwater sensor network architecture: Design, implementation, and initial testing. In *OCEANS*, 2009.

[23] Y. Zhou, X. Chen, M. Lyu, and J. Liu. Sentomist: Unveiling transient sensor network bugs via symptom mining. In *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, pages 784 –794, june 2010.