XYZ: A Motion-Enabled, Power Aware Sensor Node Platform for Distributed Sensor Network Applications

Dimitrios Lymberopoulos and Andreas Savvides
Embedded Networks and Applications Lab (ENALAB)
Yale University
51 Prospect St, Room 000
New Haven, CT 06520
email: {dimitrios.lymberopoulos, andreas.savvides}@yale.edu

***Abstract*— This paper describes the XYZ, a new open-source sensing platform specifically designed to support our experimental research in mobile sensor networks. The XYZ node is designed around the OKI ML67Q500x ARM THUMB Microprocessor and the IEEE 802.15.4 compliant CC2420 radio from Chipcon. Its new features include support for two different CPU sleep modes and a long-term ultra low power sleep mode for the entire node. This allows the XYZ and its peripheral boards to transition into deep sleep for extended time intervals. To support mobility hardware control and computations, XYZ supports a wide dynamic range and power options. In low power configuration the node resembles existing small low power nodes. When needed, the node can scale up its resources to perform more powerful computations. Mobility is enabled with an additional accessory board that allows the node to move along a horizontal string. In this paper we provide an overview of the XYZ architecture and provide an insightful power characterization of the different operational modes to allow the users to optimize their platforms for power.**

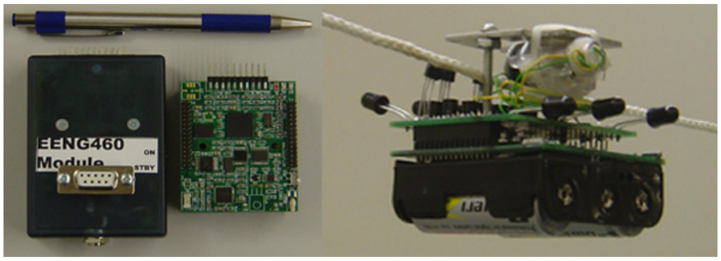

Fig. 1. The *XYZ* sensor node. Left to right: instructional XYZ, testbed module, XYZ in motion

I. Introduction

The rapid progress in sensor networks is constantly revealing new unexplored problems and applications that create the requirement for a diverse set of sensing platforms. On one hand, the creation of tiered architectures using heterogeneous nodes calls for a variety of capabilities and sensor interfaces at different levels of the hierarchy and optimized ultra low cost sensor nodes at the leafs of the network. On the other hand, the lack of complete understanding of many sensing phenomena and applications suggest that the validation of algorithms aiming to develop a fundamental understanding of dealing with sensing uncertainties should receive a higher priority over rigorous power optimizations. Two illustrative examples include the recent efforts for learning-based radio signal strength (RSS) localization [6], and the efforts to exploit mobility in sensor networks [8]. From a research perspective,

the priority in both applications is to discover and demonstrate distributed schemes that operate correctly and robustly on noisy sensor data. Such an endeavor can be better facilitated with the existence of suitable low cost and flexible low cost platforms that reduce the overhead of experimentation and preliminary deployment.

The XYZ platform takes a forward step in this direction by instantiating a new sensor node platform designed to support mobility experiments in sensor networks. Although our design is driven by the research requirements of our group, extra effort was taken during the design phase to specify a feature set that is complimentary to existing platforms and can serve multiple aspects of research and education in sensor networks. The XYZ platform described in this paper is built around an ML67Q500x series ARM/Thumb microcontroller from OKI Semiconductor and a CC2420 radio with a 250kbps raw data rate from Chipcon. The choice of the OKI microcontroller provides a wealth of peripherals and flexible modes of operation. The Chipcon radio and its use with an IEEE 802.15.4 compliant MAC protocol, make our node interoperable with other sensor nodes available in the community such as Telos and Micaz. An additional mobility board inspired from the outdoor NIMS system [15], allows the node to move along horizontal strings in indoor environments. Our design is based on the following key design considerations.

Long term sleep modes The node should be capable of long term sleep modes, that can power-off not only the processor but also all its sensors and peripherals for extended periods of time.

Support for mobility The node should have support for mobility. This includes both autonomous node mobility and mobility of wearable nodes. In addition to having a mobility accessory board, support for mobility also requires faster computation when the node is mobile, and real-time support hardware (such as hardware timers, PWM outputs and a large number of external interrupts) for controlling motors and other actuators

Computation & Memory The node should have ample computation resources and memory to allow researchers to quickly prototype and test their ideas before optimizing their code.

Multiple operational modes The node should support dynamic operation over multiple operational modes that will allow it to have low power characteristics comparable to lower end nodes but also provide it with the option to perform faster and more powerful computations when needed.

A wide choice of sensing peripherals close to the sensors From our previous experiences with sensing platforms we found out that the ability to do fast sensing close to the main processor makes prototyping easier. It eliminates several software development overheads and bottlenecks associated with moving data between processors and reduces the programming learning curve for new node users.

Cost and form factor The node should have a small form factor and compact packaging wearable applications and ubiquitous deployment. The node should also be low cost to allow the creation of scalable testbeds for experimentation.

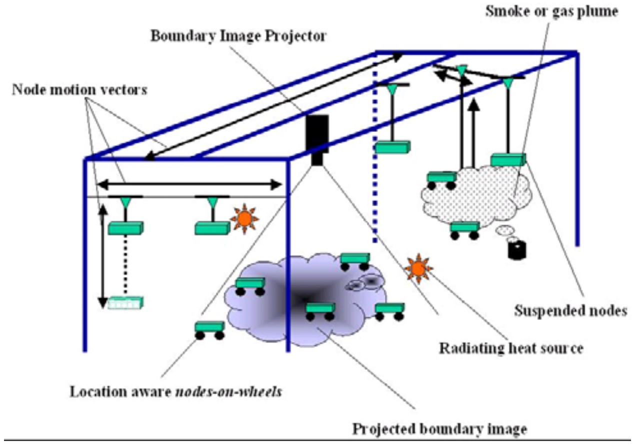

Fig. 2. An overview of the 3-D testebed installed in ENALAB at Yale University.

In this paper we introduce the first generation of the XYZ node, our first attempt to produce such a platform. In the next section, we motivate the use of XYZ in the context of our 3-D, battery operated testbed. This is followed by a description of the XYZ architecture in section III and detailed power characterization in section IV. Section V surveys similar platforms and section VI concludes our presentation.

II. 3D Testbed Overview

The main driver behind the XYZ specification is a three dimensional, battery operated scalable testbed under construction at ENALAB at Yale. The testbed (described in Fig.2) is a three-dimensional structure measuring 4.5m(W) x 6m(L) x 3m(H) deployed inside our lab. It is designed to host a large number of static and mobile nodes to instrument a variety of scenarios related to node localization, sensor calibration and mobile sensor applications such as boundary estimation [8]. A group of motion-enabled suspended nodes will move horizontally and vertically along a mesh of strings. Another group of nodes-on-wheels will roam around the testbed floor. The sensor nodes will respond to a set of stimuli generated by artificial smoke and heat sources and a video projector mounted on the testbed ceiling facing the floor. The consideration of the testbed created the need for a new node architecture that is motion enabled, easy to program and it can sustain a wide variety of sensors. To facilitate future ad-hoc deployment and to allow a high degree of mobility in our testbed, we require that the nodes in the testbed can operate on batteries for a prolonged period of time. For these reasons we designed the XYZ node to have a rich set of interfaces, a large dynamic range of operational modes and a deep sleep mode for the entire node. The details of the XYZ architecture are described in the next section.

III. The XYZ Architecture

The XYZ architecture is depicted in Fig 3. The main innovative hardware features are its ability to support long-term sleep modes through an external supervisor circuit, mobility and flexible processing modes. The main processing unit is an OKI ML67Q5002, ARM7TDMI microcontroller [10]. We found of this processor to be an appealing choice since it provides a rich set of peripherals, multiple power options and a suitable memory configuration. On XYZ the processor is set to operate at a maximum CPU clock frequency of 57.6MHz but an internal software controlled clock divider can slow down the processor in powers of two all the way down to 57.6/32MHz (1.8MHz) to conserve energy. In addition, the clock of the several on-chip peripherals can be enabled or disabled by the software offering the capability of implementing sophisticated power saving schemes. The processor also provides 2 power saving sleep modes STANDBY and HALT. When in STANDBY, the clock oscillation is completely stopped but the processor chip still receives power. In HALT mode the clock oscillation is not stopped but the clock to the CPU bus and several peripherals is blocked. Other features that made the OKI processor an appealing choice include the availability of 7 hardware timers and the option to process external interrupts on all the GPIO pins. The processor also has 256K of Flash, 32K RAM and a 4K boot ROM. We complemented this with an an additional 2Mbits (256K x 16) of external RAM memory that automatically powers off when idle.

XYZ	Operating Mode					
Component	1	2	3	5	6	LDS
CPU mode	57.6 - 1.8 MHz		standby	halt		off
I/O Config.	Timers only	Custom	none	Timers only	Custom	off off
Radio	on / off					off

TABLE I
The XYZs different operating modes.

The CC2420 radio is wired to a hardware interrupt that can wakeup the processor upon the arrival of an incoming packet. Furthermore, to reduce production costs and for instructional purposes, the XYZ circuit board also includes a basic sensing subsystem comprised of light, temperature and acceleration sensors and a uart driver module.

To reduce the battery operation cost, the XYZ power-supply was designed to use three 1.2 volt, Ni-MH rechargeable batteries, with typical capacities between 1200 and 2000mAH. The XYZ sensor node is available to the research community from Cogent Computer Systems Inc. [2] at the cost of $150 per node for a 20 node batch.

A. Long Term Deep Sleep (LDS) Modes

To support long term deep sleep we have implemented a supervisor circuit outside the processor operated by a real-time clock(RTC) with two interrupts [1]. With this circuit, the node has software control to transition into a deep-sleep mode by disabling its main power supply regulator. As Fig 4 shows in more detail, the RTC is directly powered by the batteries and not by the on-board voltage regulator. This allows the RTC to remain operational and keep track of time with its own oscillator, when the voltage regulator powering the rest of the board is disabled. In deep sleep mode, the node consumes approximately 30μA and the RTC can be scheduled to wakeup the node in intervals ranging from 1 minute to 200 years.

The sleep mode is implemented as follows. The power switch and a GPIO pin of the OKI processor control the voltage on the enable pin of the voltage regulator. After booting up the sensor node using the power switch, the CPU takes control of the voltage regulator through the GPIO pin. Driving this pin low enables the voltage regulator while driving it high disables it and shuts down all the components on the node except the RTC. One of the RTC interrupts can be used to re-enable the voltage regulator and thus restart the sensor node. After restarting the sensor node, the CPU

regains control of the enable pin on the voltage regulator. The second RTC interrupt signal is directly connected to one of the

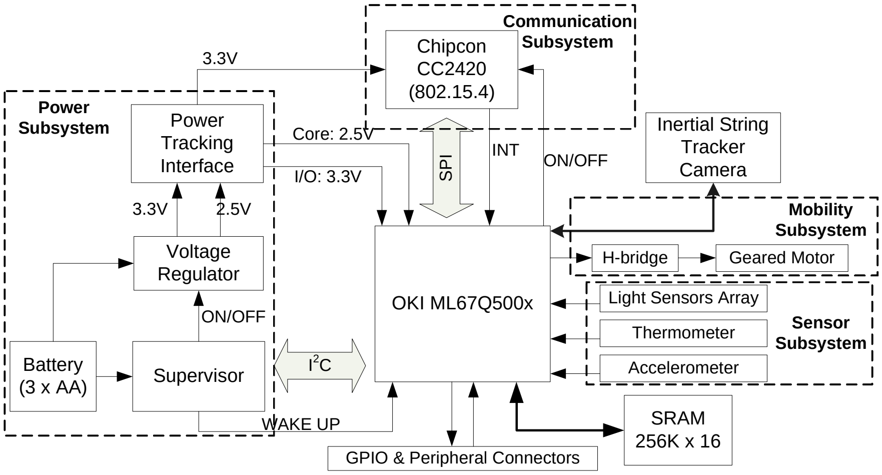

Fig. 3. *T*he XYZ node architecture

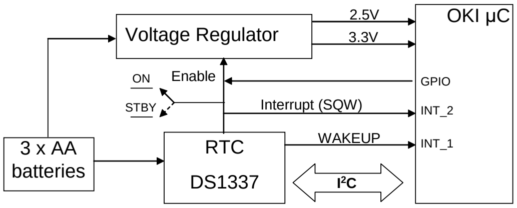

Fig. 4. Supervisor circuit for deep sleep modes

processors interrupt signals and can be used to wake up the processor from the STANDBY or HALT mode.

B. Mobility Support Mechanism

The mobility subsystem (see Fig 1 right) consists of an H-bridge and a miniature geared motor from a pager device. The mobility subsystem is implemented on an additional accessory board that transforms XYZ to a 2-D motion enabled sensor node that can move along a horizontal string as described in section II. Two output pins of the processor control the motor direction and braking. The motor is controlled via an H-bridge which converts the 3.3V logic of the CPU to a 5V DC supply necessary for supporting speed and motor power requirements. A LED/phototransistor pair focused on a 4 segment black and white pattern that is pasted directly onto a wheel acts as an optical encoder and is used as odometer. A transition from one segment to another, detected by the sensor, occurs when the sensor node is moving. The transition is fed directly to a counter on the sensor node, and by accumulating the counter values, a position value is determined.

C. Software Infrastructure

1) Operating System and Communication Protocol Stack: To make XYZ interoperable with other devices, we ported Chipcons IEEE 802.15.4 compliant medium access control protocol which we operate inside the SOS operating system developed in UCLA [5], [16].

SOS is a lightweight operating system that follows an event driven design similar to the widely used TinyOS. Unlike TinyOS however, instead of compiling a static OS image, SOS supports the use of dynamically loadable modules. These support the installation and modification of programs at runtime in a manner that resembles conventional application modules on PCs. We found this functionality to be particularly useful for maintaining and developing applications on our testbed and decided to adopt SOS. The existence of an SOS port for Crossbows Micaz node makes our platform interoperable with Micaz, and enables the effortless creation of tiered schemes.

2) XYZ Support API: To support the special features of XYZ we implemented a specialized API in SOS that consists of the following modules:

Device Drivers Module This component includes the low level interfaces to all the devices and the I/O peripherals on the XYZ sensor node including the radio, the Real Time Clock, the ADC, the UART, the timers and the DMA controller. Using this module, applications can easily initialize, configure and use all the devices connected to the various I/O interfaces of the node.

Sleep Mode Driver Module This component manages the process of entering and exiting the various XYZ sleep modes(STANBY, HALT, deep sleep). Applications have to specify the type of the sleep mode that they want to enter and the time period for which they want to remain in that mode. Then, the current module programs the onboard Real Time Clock to wake up the node after the requested time period and puts the node in the requested sleep mode.

Radio Manager Module This module is responsible for configuring the communication subsystem. Applications can call this component to change the transmission power level of the radio and/or its security configuration. In addition, this module is capable of turning on/off the radio and putting it in sleep mode.

Frequency Manager Module The frequency manager is responsible for changing the operating clock frequency of the node while preserving the correct timing of the operating system, the Zigbee MAC layer and the application running on the node. Applications only have to specify the new clock frequency. Then, all the the used timers and peripherals will be reconfigured such that the timing of the software running on the node remains the same independently of the operating clock frequency.

All of these API modules are used as the main tool for performing the XYZ power characterization described in the next section.

IV. Power Characterization

A. Measurement Setup

Our measurement setup, shown in Figure 6, uses the three 100mΩ low tolerance (1%) resistors of the XYZs power tracking interface.

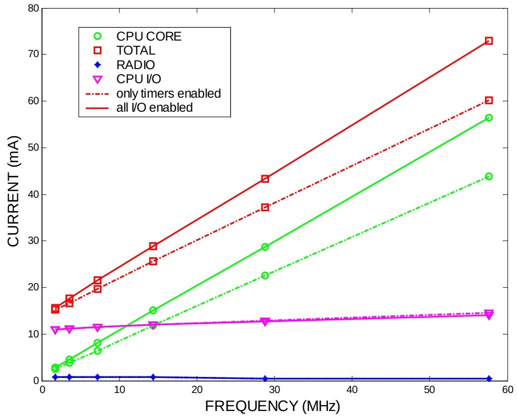

Fig. 5. Current consumption breakdown in different operating frequencies and for different I/O configurations (Total: 3.3V, CPU Core: 2.5V, CPU I/O:3.25V and Radio: 3.25V).

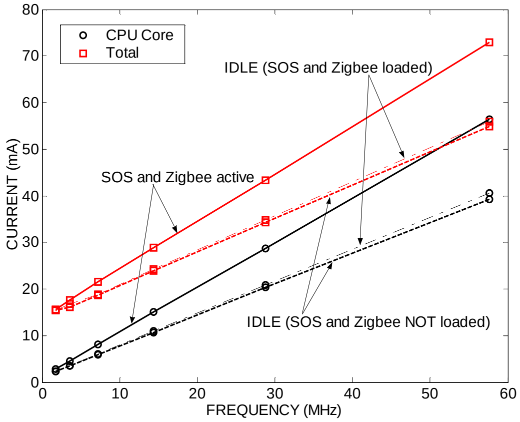

Fig. 7. The SOS and Zigbee protocol stack overhead (Total: 3.3V, CPU Core: 2.5V).

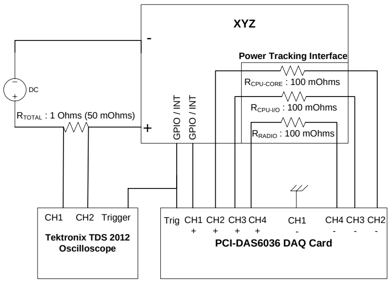

Fig. 6. The power measurement setup used for the power characterization of the XYZ sensor node.

These are connected to a PCI-DAS6036 [3] data acquisition system from Measurement Computing, which provides 8 differential channels and offers a maximum sampling rate of 200K samples per second. Using this data acquisition system we measured the voltage drop across the three resistors connected in series with the power supply lines of the CPU core, CPU peripherals and radio. In order to be able to measure the total current drawn by XYZ, we also measured the voltage drop across an external 50mΩ (or 1Ω depending on the operating mode of the node) low tolerance (1%) resistor that is connected in series with the power line of the XYZ sensor node. The maximum measurement accuracy of our data acquisition system is +/-25.6μV after averaging over 1000 samples. For each of the measurements presented in this paper at least 50K samples were acquired.

During our experiments, XYZ was powered at 3.3V from a DC regulated power supply. The on-board voltage regulator was able to deliver approximately 3.25V and 2.5V on the CPU I/O and CPU core power lines. The radio is actually powered by its internal 1.8V voltage regulator but part of the chip is directly powered by the 3.3V line. Therefore, in order to measure the total current drawn by the radio chip we had to measure the 3.3V line and not the 1.8V line of the internal voltage regulator.

The time and energy overheads for transitioning to different modes were measured with a TDS2012 oscilloscope from Tektronix. This has two measurement channels and an external trigger channel that can be used only for triggering. It supports a maximum frequency of 100MHz and is capable of acquiring 1Gs/s. The channels were used to measure the voltage drop across the resistors that are connected in series with the power lines of either XYZ or radio. The external trigger channel was connected to either a GPIO or an interrupt pin of the CPU that was enabling/disabling the acquisition. With this setup we measured the time and energy overhead of XYZ to transition into and out of its various sleep modes. With the same setup we also measured the time and the energy overhead required by the radio to transmit packets of different lengths at different power levels while using or not the supported encryption/authentication mechanisms.

All the measurements presented here were obtained with SOS and the IEEE 802.15.4 protocol stack running on the XYZ node. In terms of processing overhead this corresponds to 2 running schedulers (SOS and Zigbee schedulers) that make use of 4 hardware timers and 1 software timer implemented in software under SOS.

B. CPU Frequency Scaling

Figure 5 shows the total current drawn by XYZ and the current drawn by the whole node, the processor core and the microcontroller peripherals in all possible operating frequencies and for different configurations. The measurements were taken while the radio was in idle state and the external RAM in standby mode. The total current drawn by the node varies from almost 15.5mA in the minimum frequency to 72mA in the maximum frequency. It is obvious that both the total current and the CPU core current are a linear function of the nodes clock frequency. However, from Figure 5 it can be seen that the reduction in the total current and the current drawn by the CPU core when reducing the clock speed, heavily depends on the operating frequency. For instance, when transitioning from 57.6MHz to 57.6MHz/2 the reduction in the current drawn is much higher than when transitioning from 57.6MHz/16 to 57.6MHz/32. This trend can be explained by the static power consumption of the various components on the XYZ. As the clock speed is reduced, the overall power consumption is also reduced but the static power consumption of the various XYZ components remains almost unchanged.

Figure 5 also shows the total current and the CPU core current drawn when all the peripherals except the timers are disabled. The total current drawn by the node now varies from 15mA to 60mA which results to a reduction of 0.5mA to 12mA depending on the operating clock frequency. What is even more interesting is the fact that this reduction in power is mainly caused by the reduction in the CPU core power consumption and not by the reduction in the CPU peripherals power consumption. The reason for this is that some of the peripherals, such as the UART, are embedded in the ARM7TDMI core of the OKI processor. The detailed measurements for each power more are shown in Table II. An important observation here is that the CPU I/O subsystem draws between 11 and 14mA (35.75 45.5mW) from the power supply. We chose to pay this energy overhead to support a rich set of peripherals and

MODE			CPU Core(mA)	CPU I/O(mA)	Radio(mA)	Rest of the board(mA)	Total(mA)
CPU(MHz)	Radio	I/O receiving clock	@2.5V	@3.25V	@3.25V	@3.25V	@3.3V
57.6	ON	all	56.4	13.97	0.44	1.48	72.28
57.6	ON	timers only	43.84	14.47	0.44	1.47	60.22
57.6/2	ON	all	28.76	12.72	0.44	1.33	43.25
57.6/2	ON	timers only	22.54	12.88	0.44	1.33	37.2
57.6/4	ON	all	15	11.96	0.44	1.25	28.91
57.6/4	ON	timers only	11.84	12.05	0.44	1.04	25.68
57.6/8	ON	all	8.04	11.47	0.44	1.17	21.45
57.6/8	ON	timers only	6.46	11.52	0.44	0.98	19.74
57.6/16	ON	all	4.55	11.15	0.44	1.13	17.62
57.6/16	ON	timers only	3.75	11.21	0.44	0.94	16.68
57.6/32	ON	all	2.8	10.92	0.44	1.07	15.62
57.6/32	ON	only timers	2.4	11.05	0.44	0.97	15.18
DEEP SLEEP	OFF	OFF	0	0	0	0	0.03

TABLE II
BREAKDOWN OF THE CURRENT DRAWN BY THE XYZ NODE AT DIFFERENT OPERATING MODES.

Transition from(MHz):	STANDBY		HALT	
	Current(mA)			
	Core	Total	Core	Total
57.6(radio idle)	0	4.1	32.2	43.76
57.6/32(radio idle)	0	3.5	2.02	13.93
57.6(radio listening)	0	23.62	32.24	63.2
57.6/32(radio listening)	0	23.62	2.3	34.85

TABLE III
CURRENT DRAWN WHILE TRANSITIONING FROM DIFFERENT CLOCK FREQUENCIES TO STANDBY AND HALT MODES.

Memory Operation	FLASH		Ext. RAM	
	Time (ns)	Current (mA)	Time (ns)	Current (mA)
Read(2 bytes)	20	20	70	20
Write(2 bytes)	20	20.2	70	17.6
Erase Sector(2KB)	25	20.2	N/A	N/A
Erase Block(64KB)	25	20.2	N/A	N/A

TABLE V
INTERNAL FLASH AND EXTERNAL RAM CHARACTERIZATION.

hardware accelerators on our design. We do not consider this overhead a limiting factor since it can be optimized in an end user application using a customization process offered by OKI Semiconductor.

Figure 7 shows the overhead of running the SOS operating system and the Zigbee protocol stack at different operating frequencies. The SOS operating system uses one hardware timer while the Zigbee protocol stack uses 3 hardware timers and its own scheduler. The measurements presented in Figure 7 were taken after loading both the SOS operating system and the Zigbee MAC layer and then stopping all the hardware timers on the node. It turns out that the overhead of the operating system and the communication protocol stack is approximately 20mA at the maximum operating clock frequency. However, this overhead becomes significantly smaller as the operating clock frequency is reduced down to the minimum possible.

C. Transitioning Overheads

Table III gives the current drawn in STANDBY and HALT modes. Note, that in active mode the total current drawn in the minimum operating frequency (see Figure 5) can be more than 4 times higher than the current drawn in STANDBY mode. However, this does not seem to be the case for the HALT mode. Since the clock oscillation is not stopped in the HALT mode and most of the on-chip peripherals are active, the power consumption is higher compared to the STANDBY mode. The key observation about the HALT mode is that the power consumption in this mode heavily depends on the previous state of the processor. For instance, when the processor transitions from the maximum possible clock frequency to HALT mode the total current drawn is almost 44mA while the total current drawn becomes almost 14mA when the node enters the HALT mode from the minimum possible clock frequency. This suggests that transitioning to the HALT state should only be done from the lower clock frequencies, otherwise the reduction in the overall power consumption will not be the best possible.

Transitioning from the STANDBY or HALT state to an idle state and vice versa also incurs some overheads. Table IV shows the time required for the node to enter and exit a sleep mode while running at 57.6MHz, 57.6MHz/4 and 57.6MHz/32. It is obvious that the transition times for the STANDBY mode are significantly higher compared to the transition times for the HALT mode. This becomes apparent in the case of the wake up times, where in order to go from the STANDBY mode to an operational state with 57.6MHz clock frequency a time equal to 23.8ms is needed. The time it takes to wake up from the STANDBY mode is that high because the system has to wait until the clock oscillation is stabilized. What is even more interesting is the fact that wake up and sleep times depend on the clock frequency of the operational state we are transitioning to. The time that is required for a transition from the STANDBY mode to the minimum clock frequency is 17 times lower than the time required for transitioning to the maximum clock frequency. As a result of this, the optimal way of transitioning to STANDBY mode is to first reduce the clock frequency to the minimum possible and then jump to the sleep mode. This will reduce both the sleep and wake up times and thus it will reduce the overall power consumption.

From Table IV we note that the energy overhead for transitioning into and out of sleep modes closely follows the time overhead trend. Moreover, we note that the energy required to transition from STANDBY to the maximum clock frequency is 13.5 times higher than the energy required to transition to the minimum clock frequency.

D. Memory, Communication and Mobility Overheads

The energy cost of the other node components is given below:
Memory overhead: is summarized in Table V.
Communication overhead: The time overhead to transmit an IEEE 802.15.4 is linear to the number of bytes in the payload. Based on our measurements the time to transmit a packet was found to be

$$Transmit\_Time(ms) = 0.031 * packet\_payload + 1$$

The 1ms time corresponds to the time required for transmitting 10 payload packets (this was the minimum number of payload packets that we used in our measurements). The current drawn by the radio at different power

Frequency (MHz)	STANDBY				HALT			
	Sleep		Wake up		Sleep		Wake up	
	Time(μs)	Energy(μJ)	Time(ms)	Energy(mJ)	Time(μs)	Energy(μJ)	Time(μs)	Energy(μJ)
57.6	300	22.49	24.2	1.53	204	37.43	552	105.41
57.6/4	320	20.63	23.8	1.47	60	5.35	400	36.7
57.6/32	320	18.39	1.4	0.1	40	2.38	148	9.54

TABLE IV

TIME AND ENERGY OVERHEAD FOR TRANSITIONING INTO AND OUT OF SLEEP MODES

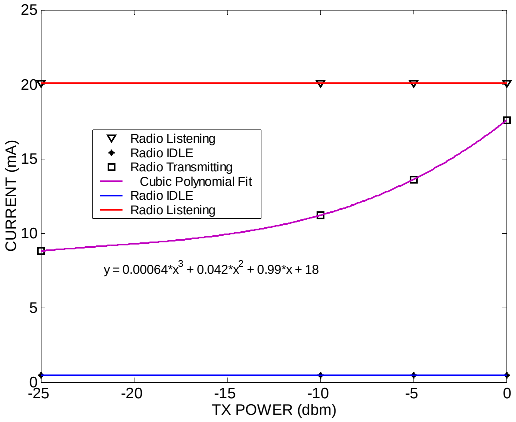

Fig. 8. Current drawn by the radio in listen, idle and transmit states for different power levels (@3.25V).

levels is given in Fig 8. Note that even at the maximum transmit power, the radio consumes less power than when listening for packets.

Mobility cost Driving the miniature geared motor (used to transform XYZ to a 2-D motion enabled sensor node) at 6 Volts results to approximately 80mA current overhead. Using three AA rechargeable batteries to power the miniature geared motor (XYZ was not powered) we found that the maximum distance that can be traveled by our sensor node is approximately 165 meters. However, when XYZ is also powered by the same power source, the maximum travel distance is smaller depending on the operating mode of the XYZ.

V. RELATED WORK

A number of sensor nodes have been developed by the community. These include the Mica series from UC Berkeley and Crossbow, Telos from Mote*iv* [14]. Micaz and Telos use the same radio as XYZ but have smaller 8-bit processors. These nodes have lower power consumption than XYZ, but also have more constraints in terms of memory computation and peripherals. Other IEEE 802.15.4 compliant nodes similar to Telos and Mica are also manufactured by Ember Inc [12], Dust Inc [11], Millenial [13] and others. The iMote from Intel [7] features an ARM/THUMB processor and a Bluetooth radio. The Stargate node[4] designed by Intel manufactured by Crossbow is a higher end node. Stargate has a PXA processor with variable clock speed ranging from 100 to 400MHz, 32MB of Flash memory and 16MB of SDRAM. The Stargate has a mote interface and can also support IEEE 802.11 and Ethernet communication. The PASTA node [9] is a hierarchical collection of hardware modules that can be individually powered down on demand. PASTA features multiple processors and has a dynamic range $>$ 1000X ranging from $<$ 1mW to $>$ 10W.

Our work is complementary to this in the sense that it has similar functionality and features. In addition to these features, XYZ has a larger dynamic range of operation and a deep sleep mode that allows it to hibernate for prolonged time periods. At its lowest power operational mode, XYZ resembles a Micaz mote, and at its most powerful mode it begins to overlap with the functionality of an iMote or a Stargate.

VI. CONCLUSIONS

In this paper we have described the XYZ architecture and demonstrated the usage of its main power features through a detailed power characterization. The results of the characterization give an indication of the overall energy consumption of XYZ during operation and provide valuable insight on what would be the best way to operate the different modes. The XYZ node is currently being used as the main building block of our sensor infrastructure and testbed. XYZ is also the main instructional platform in EENG460a a new Networked Embedded Systems and Sensor Networks course at Yale. Our up-to-day experience with XYZ has shown that further power supply and computation performance optimizations are possible. Our future plans include the development of smart environments using XYZ. After completing and rigorously characterizing a set of applications in our research, we plan to develop a second generation XYZ node. At the same time, based on the results presented in this paper we will pursue more rigorous energy optimization schemes at the OS and network levels, to better leverage the existing XYZ features. More details and updates of our work can be found on the XYZ website at http://www.eng.yale.edu/enalab/XYZ.

ACKNOWLEDGMENT

We would like to acknowledge Cogent Computer, OKI Semiconductor, and Chipcon for their help in the fabrication of XYZ. We are also thankful to Simon Han from NESL, UCLA for his help with SOS and Quentin Lindsey, Stephen Tully and Jimmy Zhang for their help with the mobility board.

REFERENCES

[1] Dallas Semiconductor MAXIM: DS1337 Real Time Clock. http://www.maxim-ic.com/quick_view2.cfm/qv_pk/3128.

[2] Inc. Cogent Computer Systems. http://www.cogcomp.com/.

[3] PCI-DAS6036 Data Acquisition System from Measurement Computing. http://www.measurementcomputing.com/pci-das6036specs.html.

[4] Stargate gateway by Crossbow. http://www.xbow.com/products/xscale.htm.

[5] Chih-Chieh Han, Ram Kumar, Roy Shea, Eddie Kohler, and Mani Srivastava. A dynamic operating system for sensor nodes. Technical Report NESL-TM-2004-11-01, University of California Los Angeles, Networked Embedded Systems Lab, November 2004.

[6] D. Madigan, E. Elnahrawy, Richard P. Martin, Wen-Hua Ju, P. krishnan, and A. S. Krishnakumar. Bayesian indoor positioning systems. In *INFOCOM (to appear)*, 2005.

[7] Intel mote (*i*Mote). http://www.intel.com/research/exploratory/motes.htm.

[8] A. Savvides, J. Fang, and D. Lymberopoulos. Using mobile sensing nodes for dynamic boundary estimation. In *WAMES workshop in conjuction with MobiSys*, Boston, Massachusetts, USA, 2004.

[9] B. Schott, M. Bajura, J. Czarnaski, J. Flidr, T. Tho, and L. Wang. A modular power-aware microsensor with ¿ 1000x dynamic power range. In *Proceedings of Information Processing in Sensor Networks Los Angeles, CA*, April 2005.

[10] OKI semiconductor: 32-bit ARM Based General Purpose Microcontrollers ML67Q5002. http://www2.okisemi.com/us/docs/intro-9980.html.

[11] Dust Inc. website. http://www.dust-inc.com/.

[12] Ember Inc. website. http://www.ember.com/.

[13] Millenial Net website. http://www.millennial.net/.

[14] Mote*iv* Website. http://www.moteiv.com.

[15] Networked Infomechanical Systems(NIMS) Website. www.cens.ucla.edu/research/technology/nims_web/nims_web/.

[16] SOS Website. http://nesl.ee.ucla.edu/projects/sos/.