Wireless Sensor Network Hardware Platforms and Multi-channel Communication Protocols: A Survey

Manish Karani   Ajinkya Kale   Animesh Kopekar

Department of Electronics Engineering, Vishwakarma Institute of Technology PUNE

ABSTRACT
Wireless sensor networks, at present are one of the major technologies which have the potential of transforming the future of the human civilization. Wireless Sensor Networks are being used in various fields such as industrial, medical, military, disaster relief, etc. They have many proposed applications, and with every application come new protocols. Hence, a survey of the existing hardware platforms and communication protocols is necessary for a comparative overview as a first step in the deployment of wireless sensor networks for application to any field. In this paper we discuss the hardware platforms like Shimmer, TelosB, Ember, and two development kits of Texas Instruments. Also, widely used wireless communication protocols which use channel switching mechanisms have been discussed for improvement of reliable communication.

General Terms
Wireless Sensor Networks

Keywords
Wireless sensor networks, Multi channel communication protocols, Hardware platforms, Survey.

1. INTRODUCTION
Recent advances in embedded computing systems have lead to the emergence of wireless sensor networks, consisting of small, battery-powered "motes" with limited computation and radio communication capabilities. Sensor networks permit data gathering and computation to be deeply embedded in the physical environment. This technology has the potential to impact the delivery of data (of any format) to receiver that automatically collects it, it is fully integrated device and it is used for real-time, correlation with various computation and long-term observations.

Large node number, constrained (energy) resources, data-centric networking, in-network processing – are by now commonplace in the literature. Less considered is the appropriate notion of reliability:

The sensors are cheap and their readings can thus be noisy, giving rise to the *information accuracy problem*. Also they do not have reliable protocols incorporated in them, which is why reliability becomes critical in WSN.

The standard approach is a combination of redundancy, i.e. to deliver multiple sensor readings and improve the signal-to-noise ratio. But in this approach even when an event has been reliably detected, this information must be transported over multiple hops towards desired node which increases energy consumption. The other approach is to go for Hardware switching by using two or more transceivers. But this approach makes the system more complex and increases cost. One more option is frequency switching using software which uses two or more frequency bands for data transmission. This is cost effective, less complex and easy for modeling and simulation.

In this paper, we discuss different hardware platforms and multi frequency transmission protocols (for Software switching) that ensure the reliability of transmitted data in ad-hoc wireless sensor networks.

2. WSN HARDWARE PLATFORMS
2.1 Shimmer
(Secure Health with Intelligence, Modularity, Mobility & Experimental Reusability)

SHIMMER is fully integrated with BioMOBIUS high-level application environment which enables rapid prototyping. It is a Wireless Sensor Platform for Noninvasive Biomedical Research. It enables rapid prototyping of biomedical research applications. SHIMMER DESIGN platform comprises of a baseboard which provides capabilities like, Sensors computation like a passive tilt vibration sensor, A PIR sensor is used as power saving wakeup trigger when user approaches. Data storage that facilitate recording of data to microSD card. Communications includes radio CC2420, Bluetooth and IEEE 802.15.4. And it is also having an included Daughterboard connection in it. SHIMMER also supports TinyOS-2.x. SHIMMER is an extremely flexible sensor platform. It has the ability to seamlessly expand to meet various biomedical researches [1].

2.2 Ember EM250
The EM250 includes 128 kB of onboard Flash Read Only Memory (ROM). It also allows for three different modes of operation. The Active operation will allow for execution of the program code, typically using 8.5 mA of current. The Idle operation allows for the MCU to shut down until an interrupt occurs while allowing peripherals and the transceiver to operate normally. The EM250 also allows for a Deep Sleep operation which powers down the MCU and Transceiver until either an external interrupt or a timer wakes the device. In the Deep Sleep operation, the EM250 typically uses 1.5 mA of current. The EM250 has four ADCs, of which two are used for use of capturing analog data. The EM250 also has the capability of communicating over serial peripheral interface (SPI).

2.3 TelosB
Unit price of TelosB is high, around $150 each and there is no discount for educational purpose. Its power lifetime is around 3–6 months depending on how often the signal is transmitted back to the server, which are somewhat short for medical applications. Its radio components cannot be enhanced (we cannot use a better radio transceiver/antenna to reach a longer distance). It is an ultra low-power wireless module intended for sensor networks applications. The mote platform offers the on-chip RAM of 10 kB and also provides IEEE 802.15.4 Chipcon radio with an integrated on-board antenna providing up to 125 m of range, structured around a TI MSP430 microcontroller. TelosB mote is also referred to as the Tmote Sky [2].

2.4 eZ430-RF2500

The eZ430-RF2500 is a complete USB-based MSP430 wireless development tool providing all the hardware and software to evaluate the MSP430F2274 microcontroller and CC2500 2.4-GHz wireless transceiver. The cost of the debugger and development tool software is $29 and the target board i.e. mote is $20.

The debugger is unobtrusive, allowing the user to run an application at full speed with both hardware breakpoints and single stepping available while consuming no extra hardware resources.

The eZ430-RF2500T target board is an out-of-the box wireless system that may be used with the USB debugging interface, as a stand-alone system with or without external sensors, or may be incorporated into an existing design [3].

eZ430-RF2500 features includes USB debugging and programming interface featuring a driverless installation and application backchannel. It is having 21 available development pins. Highly integrated, ultra-low-power MSP430 MCU with 16-MHz performance is also given with it. Two general-purpose digital I/O pins connected to green and red LEDs for visual feedback. Interruptible push button for user feedback is present.

MSP430F2274 is having the advantages of 16-MIPS performance,10-bit SAR ADC, Two built-in operational amplifiers, Watchdog timer, two 16-bit timers, USCI module supporting UART/LIN, (2) SPI, I2C, or IrDA, Five low-power modes drawing as little as 700 nA in standby.

CC2500 is having the advantages of 2.4-GHz radio-frequency (RF) transceiver, Programmable data rate up to 500 kbps, Low current consumption eZ430-RF2500 target board was designed to optimize for factors. The eZ430-RF2500 can be used as a stand-alone development tool. The target board features an MSP430F2274 and most of its pins are easily accessible.

The Devices Supported The eZ430-RF USB debugging interface may be used as a standard Flash Emulation Tool through its Spy-Bi-Wire interface. The eZ430-RF USB debugging interface supports the following MSP430 families: MSP430F20xx and MSP430F22xx [3].

2.5 CC1110 & CC2510 Development Kit

The CC1110 and CC2510 are System-on-Chip (SoC) devices from Texas Instruments designed for low power wireless applications. CC1110 operates in the sub-1 GHz unlicensed ISM bands while the CC2510 operates in the 2.4 GHz unlicensed ISM bands. The CC1110 and CC2510 combine the excellent performance of the state-of-the-art CC1101 and CC2500 RF transceivers respectively with an industry-standard enhanced 8051 MCU, up to 32 kB of in-system programmable flash memory and up to 4 kB of RAM, and many other powerful features. This is ensured by several advanced low-power operating modes.

2.6 Some other Hardware Platforms

Table 1. Wireless sensor network boards [8]

Corp.	RF Module	Sensitivity (dBm)	Current (R, T, mA)	Description
TI	CC2420	-95	19,17,4	Transceiver
	CC243x	-94	27,25	SoC,8051 CPU
Freescale	MC1319x	-92	37,30	Transceiver
	MC132x	-92	38,31	SiP,HCS08
Radio Pulse	MG2400	-99	26,33	SoC,8051 CPU
Ember	EM250	-97.5	35.5,35.5	SoC,16bit CPU
Jennie	JN513x	-97	39,39	SoC,32bit CPU
OKI	ML 7222	-90	26,24	SoC,8bit CPU
Nanotron	NA5TRI	-95	27,23	802.15,4a
Echelon	Pyxos FT	….	….	Pyxos
Zensys	ZW0201	-101	21,23	Z-wave
CSR	BlueCore	-85	46,52	Bluetooth

Table 2. Wireless sensor network MCU chips [8]

Corp.	Chip Type	CPU	RAM (KB)	Flash (KB)
Silicon	80C51/C8051F	CIP-51	8	128
Microchip	PICF18F4620	PIC	4	64
Freescale	MC9S08GT	HCS08	4	60
	MCF5222x	ColdFire®	32	256
Atmel	ATMEGA128L	RISC	4	128
	AT91	ARM	256	1024
Intel	8051	MCS-51	1	16
	PXA27X	XScale®	256	32
TI	MSP430F413	MSP430	10	48`
Samsung	S3C44B0	ARM	8	N/A
OKI	4050/4060	ARM	A6	128

Table 3. Wireless sensor network nodes [8]

Node	MCU	RFModule	Organization
MICAz	Atmega1281	CC2420	UCB
Telos	MSP430	CC2420	Moteiv
M2020	MSP430	CC2420	Dust Inc.
BSN Node	MSP430	CC2420	Imperal
CIT Node	PIC16F877	NordicnRF903	CIT
MIT Node	8051	NordicnRF24	MIT
Pluto	MSP430	CC2420	Harvard
iMote	ARM7TDMI	Bluetooth	Intel
iMote2	Intel PXA	CC2420	Intel
EmberNet	Atmega1281	Ember250	Ember
IP-Link	MSP430	CC2420	Helicomm
Spot	ARM	CC2420	Sun
Zbnode	ARM	CC2420	Taiwan ITRI
XYZ	ARM	CC2420	Yale
WINS	PXA255	802.11b	Sensoria
Embernet	ATmega1281	Ember250	Ember
Cicada1	MC9S08GT60	MC13193	Tsinghua
Cicada2	MC13193		Tsinghua

3. WSN COMMUNICATION PROTOCOLS

3.1 MMAC

Time synchronization between nodes is required and this protocol divides time into beacon intervals. A data structure called *Preferable Channel List* (PCL) is maintained by each node in the network that keeps track of the communications on the channels within its neighborhood. A channel can have three states at a node: high, medium, and low preference. High preference state at a node indicates that the node has already selected this channel for the current beacon interval and must continue to choose this channel in that interval. Medium preference indicates that this channel is vacant and has not been taken by any neighboring node within the neighborhood. Low preference indicates that this channel has already been taken by at least one neighbor within the nodes neighborhood. Each beacon starts with a small window called ATIM window, during an ATIM window, all nodes switch to a default channel to exchange beacons and ATIM frames. When node A wants to send data to node B, node A inserts it's PCL into an ATIM frame and sends it to node B. Node B compares A's PCL with its own PCL in order to select a channel and transmits an ATIM-ACK back to node A. If node A cannot select the channel specified by node B as it has already chosen another channel, it must wait until the next beacon interval. Otherwise, node A sends an ATIM-RES frame with the selected channel so that neighboring nodes within its transmission range can update their PCL. After the ATIM window, the nodes switch to the selected channel to communicate.

3.2 SSCH

SSCH protocol also requires time synchronization among nodes. SSCH protocol divides time into slots where each node maintains a *channel schedule* which contains a list of channels that the node plans to switch to in subsequent time slots. A node's channel schedule is represented as a current channel and a set of rules for updating current channel. Each node iterates through its set of (channel, seed) pairs in each slot to determine the channel it has to visit next [6]. Each node periodically broadcasts its channel schedule and simultaneously keeps track of other nodes' channel schedules. When a node wants to send data to another node, the node has to follow the other node by adopting its channel schedule.

3.3 AMCP

AMCP protocol does not require time synchronization among nodes; it uses a control channel which can be used by all nodes on which the nodes exchange control frames in order to negotiate reservation channel. Each node maintains a channel table which indicates whether a channel is available for communication or the period of time it is being used by other nodes within the node's transmission range. Also each node also maintains a variable *prefer* that indicates its preference of channel for communication.

A nonzero *prefer* value represents the index to the node's preferred channel. A zero *prefer* indicates the node has no preference of channel. When node A wants to send data to node B, it first needs to select a transmission channel. If node A's variable *prefer* contains nonzero value and the chosen channel is available, node A will select this channel, else node A will randomly select one of its available transmission channels. Node A inserts the index of this selected channel into an RTS frame and sends it to node B. On receiving the RTS frame, node B sends a *Confirming CTS* frame with the index of the selected data channel if that channel is available for B.

Otherwise, B sends a *Rejecting CTS* frame with a list of its available channels to A. Upon receiving a *Confirming CTS*, node A switches to the selected channel and transmits packet to B. Otherwise, node A selects a channel available for both A and B and sends a new RTS frame to node B. Other neighboring nodes can overhear the RTS and CTS frames and update their channel tables for channel availability accordingly.

3.4 Centralized Channel Assignment and Routing Multi Channel Protocol

In this protocol, the authors propose an architecture that uses multiple frequency channels in an ad hoc network by equipping nodes wit h multiple NICs.

The bandwidth problem further increases for multi-hop ad hoc networks because of interference from simultaneous adjacent hops in the same path as well as from neighboring paths. The authors have developed 2 channel assignment and bandwidth allocation algorithms for the proposed multi-channel wireless mesh network. The first algorithm,

Neighbor Partitioning Scheme performs channel assignment based only on network topology. The approach discussed to the channel assignment problem is, start with one node, partition its neighbors into groups and assign one group to each of its interface. Each of this node's neighbors partitions its neighbors into groups but with the condition of maintaining the grouping done by the first node as a constant. The process is iteratively repeated until all nodes have partitioned their neighbors. Each group can then be bound to the least-used channel in the neighborhood. This scheme presents a way to partition neighbors in a uniform channel assignment across the network. Each node has 2 NICs, but the resulting network uses 4 channels. Randomization techniques can be used for partitioning of neighbors for a general network.

The above *neighbor partitioning scheme* allows a network to use more data channels than the number of interfaces per node, but it does not consider the traffic load on the virtual links between neighboring nodes. If each virtual link in the network has the same traffic load this scheme thus would work well.

Load-Aware Channel Assignment further exploits traffic load information. The load-aware channel assignment algorithm can work with different routing algorithms and is not restricted to one algorithm. Two different routing algorithms have been explored –Shortest path routing, and randomized multipath routing. The shortest path routing is based on standard Bellman-Ford algorithm with minimum hop-count metric. The shortest path here refers to a shortest "feasible" path, *i.e.*, a path with sufficient available bandwidth and least hop-count [4]. The Randomized Multi-path routing algorithm attempts to achieve data load balancing by distributing the data traffic between a pair of nodes among multiple available paths at run time. The exact set of paths between a communicating node pair is chosen randomly out of the set of available paths with sufficient bandwidth. The traffic between a node pair is segregated across multiple paths, but packets associated with a network connection still follow a single path to avoid TCP re-ordering.

3.5 Hyacinth

Hyacinth is an extension of the *Centralized Channel Assignment and Routing Multi Channel Protocol* [4]. *Hyacinth* effectively addresses the bandwidth issue by completely utilizing non-overlapped radio channels made available by the IEEE 802.11 standards. In *Hyacinth* each mesh node is equipped with multiple 802.11 NICs; it then performs a traffic-aware channel assignment and routing to utilize multiple channels. Equipping each mesh network node with just 2 NICs can increase the network goodput by a factor of 6 to 7 when compared with conventional single-channel ad hoc network architecture.

Load-Balancing Routing decides how to route packets across a WMN in case of *Hyacinth* architecture. The traffic distribution of a WMN is skewed – most of the WMN nodes communicate primarily with nodes on the wired network. This is the case because most users are primarily interested in accessing the Internet or enterprise servers, both of which are likely to reside on the wired network. The Load Balancing algorithm therefore determines route(s) between each traffic aggregation access point and the wired network in a way that balances the load on the mesh network [5].

The advantages of Load balancing are that it avoids bottleneck links, and increases the network resource utilization efficiency.

4. CONCLUSIONS AND FUTURE WORK

This paper mainly surveys current research status of various wireless communication protocols, and hardware platforms. We described various multi-channel routing protocols that improve reliability using multiple frequency channels.

Reliability in sensor networks is multi-faceted and reliable data transport is a very important and interesting issue and we have tried to find the best communication protocol's which will ensure the reliable transmission. Some of the protocols discussed in this paper attack the reliability problem by using multiple radio channels.

There is accordingly plenty of room for interesting research:

Design and evaluation of mechanisms for further improvement of reliability, taking into account the complex behavior of wireless networks, and experimental studies regarding reliability and energy-efficiency in wireless sensor networks.

5. ACKNOWLEDGMENTS

We sincerely thank Asst. Prof. R. B. Vatti for his guidance and insightful discussions and our colleague Alok Jadhav for his efforts in this paper.

6. REFERENCES

[1] Burns A, Greene B.R, McGrath M.J, O'Shea, Kuris B, Ayer S.M., Stroiescu F. Cionca V. Sept. 2010. SHIMMER™ – A Wireless Sensor Platform for Noninvasive Biomedical Research. Sensor Journal, IEEE.

[2] Fei Hu, Meng Jiang, Laura Celentano, Yang Xiao. 2007. Robust medical ad hoc sensor networks (MASN) with wavelet-based ECG data mining. Ad Hoc Networks Volume 6, Issue 7, September 2008, Pages 986-1012.

[3] "eZ430-RF2500 Development Tool User's Guide" Texas Instruments.

[4] Ashish Raniwala, Kartik Gopalan, Tzi-cker Chiue. Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks. Mobile Computing and Communications Review, Volume 8, Number 2.

[5] Ashish Raniwala Tzi-cker Chiueh. Architecting a High-Capacity Last-Mile Wireless Mesh Network.

[6] Long Le. 2010. Performance Analysis of Multi-Channel MAC Protocols in Multi-Hop Ad Hoc Networks. IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

[7] CC1110 & CC2510 Mini Development Kit User's Guide" Texas Instruments.

[8] Zhongmin Pei, Zhidong Deng, Bo Yang, Xiaoliang Cheng. 2008. Application-Oriented Wireless Sensor Network Communication Protocols and Hardware Platforms: a Survey. IEEE. IEEE international conference on industrial technology (IEEE ICIT 2008).