Wireless Sensor Networks' emergence and Growth- A survey

L.K. HEMA [1], Dr. D. MURUGAN [2], M.CHITRA [3]
[1], [3]Faculty, Aarupadai Veedu Institute of Technology, OMR, Paiyanoor-104, INDIA
[2] Faculty, Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli-12, INDIA

Abstract-
Wireless Technology based smart sensor networks are becoming predominant from research point of view, since these smart sensors possess the exclusive features like mobility, ad-hoc nature of topology, heterogeneity of nodes, and deployment in huge scales along with the hardship of energy harvesting and routing. In this paper the concern is about the survey of emergence, modification, deployment of sensors in various real time applications where human intervention is risky.

Keywords- Reliability, Wireless Networks, Energy harvesting.

I. Introduction

The recent advances in cost-effective, power managed wireless communication coupled with ad-hoc networking and routing optimization, have made "wireless sensor networks" a hot topic. The main players in a WSN environment are motes, base stations, actuators and gateways. In general the hardware components in a wireless sensor network include analog-to-digital converters, sensor circuits, microprocessors, and wireless transceivers. These hardware components with added intelligence to work co-operatively with software for accomplishing a user defined task. The main objective of such a network is data acquisition and RF transmission.

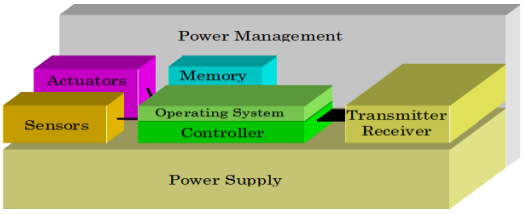

Figure -1 Block diagram of sensor node architecture

Figure 1 shows the block diagram of typical node architecture. Sensor nodes [2] are deployed in real time applications like military, agriculture, food industry, virtual habitat monitoring, health care monitoring, process monitoring in industries et al. The critical areas to be concentrated in the deployment of sensors are energy harvesting [23], effective storage and accessing of aggregated data and determination of the size of the network.. The micro-controller used in a wireless sensor node operates at low frequency compared to traditional contemporary processing units. These resource-constrained sensors are an impressive example of a System on Chip (SoC). Dense deployment of sensor nodes in the sensing field and distributed processing through multi-hop communication among sensor nodes is required to achieve high quality and fault tolerance in WSNs.

Ii. Emergence And Development Of Wsn

The prime investigation of Wireless Sensor Networks are done in military applications. The deployment of wireless sensor networks in military applications [17] is to monitor specific militant activity in remote areas such as country borders, coastal areas and so on. Some sensor networks are also deployed and operated for a required period of one or two months of time .Sensor nodes configure themselves in an ad hoc fashion and communicate among themselves by flooding and they remain reasonably static and moves during operation. The main features to be considered are reconfiguration of the network and monitoring the node failure without any manual intervention.

2.1 Generation of sensor networks

The evolution of sensor networks in terms of generations is as follows.

2.2.1 First generation sensor networks.

Sensor networks consist of individual sensor nodes which are deployed manually. This is a preconfigured network in which retrieval of information is not automatic

2.2.2 Second generation sensor networks

These are preconfigured collaborative sensors deployed manually. The special feature of this generation network is the control node acting as a hub to which the sensors can communicate.

2.2.3Third generation sensor networks

These sensors are self organizing and structured with added flexibility.

Sensors communicate among themselves for delivering messages to network gateway and also for network processing automatically

III. WIRELESS SENSOR NETWORK

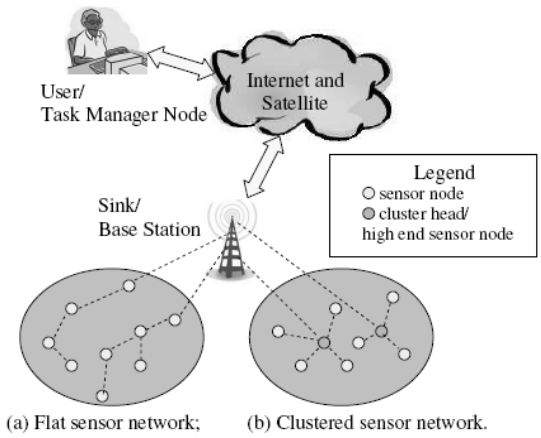

Figure 2. Architecture of WSN

A WSN [2] is a system comprised of radio frequency (RF) transceivers, sensors, microcontrollers and power sources. Recent advances in wireless sensor networking technology have led to the development of low cost, low power, multifunctional sensor nodes. Sensor nodes enable environment sensing together with data processing. Instrumented with a variety of sensors, such as temperature, humidity and volatile compound detection, allow monitoring of different environments. They are able to network with other sensor systems and exchange data with external users .Sensor networks are used for a variety of applications, including wireless data acquisition, process management and maintenance, habitat monitoring, maintenance of smart buildings and highways, environmental monitoring, site security, automated on-site tracking of expensive materials, safety management, agriculture, food industry and in many other areas .A general WSN protocol [2] consists of the application layer, transport layer, network layer, data link layer, physical layer, power management plane, mobility management plane and the task management plane .

The standard technologies available for WSN are ZigBee and Bluetooth. Both operate within the Industrial Scientific and Medical (ISM) band of 2.4 GHz, which provides license free operations, huge spectrum allocation and worldwide compatibility. In general, as frequency increases, bandwidth increases allowing for higher data rates but power requirements are also higher and transmission distance is considerably shorter [12]. Multi-hop communication over the ISM band might well be possible in WSN since it consumes less power than traditional single hop communication [8].It is also possible to create a WSN using Wi-Fi (IEEE 802.11), but this protocol is usually utilized in PC-based systems because it was developed to extend or substitute for a wired LAN [22]. Its power consumption is rather high, and the short autonomy of a battery power supply still remains an important disadvantage [4].

Figure 2 shows the architecture of WSN which consists of spatially distributed sensor nodes. In a WSN, each sensor node is able to independently perform some processing and sensing tasks. Furthermore, sensor nodes communicate with each other in order to forward their sensed information to a central processing unit or conduct some local coordination such as data fusion. One widely used sensor node platform is the Mica2 Mote developed by Crossbow Technology .The usual hardware components of a sensor node include a radio transceiver, an embedded processor, internal and external memories, a power source and one or more sensors.

Iv.History of Wsn

The origin of the research on WSNs can be traced back to the Distributed Sensor Networks (DSN) program at the Defense Advanced Research Projects Agency (DARPA) at around 1980 [2]. By this time, the ARPANET (Advanced Research Projects Agency Network) had been operational for a number of years, with about 200 hosts at universities and research institutes .DSNs were assumed to have many spatially distributed low-cost sensing nodes that collaborated with each other but operated

autonomously, with information being routed to whichever node was best able to use the information. At that time, this was actually an ambitious program. There were no personal computers and workstations processing was mainly performed on minicomputers and the Ethernet was just becoming popular. Technology components for a DSN were identified in a Distributed Sensor Nets workshop in 1978.These included sensors (acoustic), communication and processing modules, and distributed software. Researchers at Carnegie Mellon University (CMU) even developed a communication-oriented operating system called Accent (Rashid & Robertson, 1981), which allowed flexible, transparent access to distributed resources required for a fault-tolerant DSN. A demonstrative application of DSN was a helicopter tracking system (Myers et al., 1984), using a distributed array of acoustic microphones by means of signal abstractions and matching techniques, developed at the Massachusetts Institute of Technology (MIT). Even though early researchers on sensor networks had in mind the vision of a DSN, the technology was not quite ready. More specifically, the sensors were rather large (i.e. shoe box and up) which limited the number of potential applications. Further, the earliest DSNs were not tightly associated with wireless connectivity. Recent advances in computing; communication and micro electromechanical technology have caused a significant shift in WSN research and brought it closer to achieving the original vision. The new wave of research in WSNs started in around 1998 and has been attracting more and more attention and international involvement. In the new wave of sensor network research, networking techniques and networked information processing suitable for highly dynamic ad hoc environments and resource constrained sensor nodes have been the focus. Further, the sensor nodes have been much smaller in size (i.e. pack of cards to dust particle) and much cheaper in price, and thus many new civilian applications of sensor networks such as environment monitoring, vehicular sensor network and body sensor network have emerged. Again, DARPA acted as a pioneer in the new wave of sensor network research by launching an initiative research program which provided the present sensor networks with new capabilities such as ad hoc networking, dynamic querying and tasking, reprogramming and multitasking.

At the same time, the IEEE noticed the low expense and high capabilities that sensor networks offer. The organization has defined the IEEE 802.15.4 standard for low data rate wireless personal area networks. Based on IEEE 802.15.4, ZigBee Alliance has published the ZigBee standard which specifies a suite of high level communication protocols which can be used by WSNs. Currently; WSN has been viewed as one of the most important technologies for the 21st century. The commercialization of WSNs is also being accelerated by new formed companies like Crossbow Technology and Dust Networks.

V.Hardware

A WSN consists of spatially distributed sensor nodes. In a WSN, each sensor node is able to independently perform some processing and sensing tasks [17]. Furthermore, sensor nodes communicate with each other in order to forward their sensed information to a central processing unit or conduct some local coordination such as data fusion. One widely used sensor node platform is the Mica2 Mote developed by Crossbow Technology .The usual hardware components of a sensor node include a radio transceiver, an embedded processor, internal and external memories, a power source and one or more sensors.

(a) Embedded Processor

In a sensor node, the functionality of an embedded processor is to schedule tasks, process data and control the functionality of other hardware components. The types of embedded processors that can be used in a sensor node include Microcontroller, Digital Signal Processor (DSP), Field Programmable Gate Array (FPGA) and Application Specific Integrated Circuit (ASIC). Among all these alternatives, the Microcontroller has been the most used embedded processor for sensor nodes because of its flexibility to connect to other devices and its cheap price. For example, the newest CC2531 development board provided by Chipcon (acquired by Texas Instruments) uses 8051 microcontroller, and the Mica2 Mote platform provided by Crossbow uses ATMega128L microcontroller and also the MSP430 microcontroller from Texas Instruments.

(b)Transceiver

A transceiver is responsible for the wireless communication of a sensor node. The various choices of wireless transmission media include Radio Frequency (RF), Laser and Infrared. RF based communication fits to most of WSN applications. The operational states of a transceiver are Transmit, Receive, Idle and Sleep. Mica2 Mote uses two kinds of RF radios: RFM TR1000 and Chipcon CC1000. The outdoor transmission range of Mica2 Mote is about 150 meters.

(c)Memory

Memories in a sensor node include in-chip flash memory and RAM of a microcontroller and external flash memory. For example, the ATMega128L microcontroller running on Mica2 Mote has 128-Kbyte flash program memory and 4-Kbyte static RAM. Further, a 4-Mbit Atmel AT45DB041B serial flash chip can provide external memories for Mica and Mica2Motes

(d) Power Source

In a sensor node, power is consumed by sensing, communication and data processing. More energy is required for data communication than for sensing and data processing. Power can be stored in batteries or capacitors. Batteries are the main source of power supply for sensor nodes. For example, Mica2 Mote runs on 2 AA batteries. Due to the limited capacity of batteries, minimizing the energy consumption is always a key concern during WSN operations. To remove the energy constraint, some preliminary research working on energy-harvesting techniques for WSNs has also been conducted. Energy-harvesting techniques

convert ambient energy (e.g. solar, wind, pressure to electrical energy and the aim is to revolutionize the power supply on sensor nodes.

(e) Sensors

A sensor is a hardware device that produces a measurable response signal to a change in a physical condition such as temperature, pressure and humidity. The continual analog signal sensed by the sensors is digitized by an analog-to-digital converter and sent to the embedded processor for further processing. Because a sensor node is a micro-electronic device powered by a limited power source, the attached sensors should also be small in size and consume extremely low energy. A sensor node can have one or several types of sensors integrated in or connected to the node.

Vi. Operating System

The role of any operating system (OS) is to promote the development of reliable application software by providing a convenient and safe abstraction of hardware resources. OSs for WSN nodes are typically less complex than general-purpose OSs both because of the special requirements of WSN applications and because of the resource constraints in WSN hardware platforms [19].

(a)Tiny OS

Tiny OS is perhaps the first operating system specifically designed for WSNs. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in WSNs. Tiny OS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools - all of which can be further refined for a custom application. Unlike most other OSs, Tiny OS is based on an event-driven programming model instead of multithreading. Tiny OS programs are composed into event handlers and tasks with run-to-completion semantics. When an external event occurs, such as an incoming data packet or a sensor reading, TinyOS calls the appropriate event handler to handle the event. Event handlers can post tasks that are scheduled by the TinyOS kernel at a later

stage. Both the TinyOS system and programs written for TinyOS are written in a special programming language called nesC which is an extension of the C programming language. NesC is designed to detect race conditions between tasks and event handlers. Currently, Tiny OS has been ported to over a dozen platforms and numerous sensor boards. A wide community uses it in simulation to develop and test various algorithms and protocols. As Tiny OS is open source, numerous groups are actively contributing code to the development of TinyOS and thus making it even more competitive.

(b)Contiki

Contiki is another open source OS specifically designed for WSNs. The Contiki kernel is event-driven, like TinyOS, but the system supports multithreading on a per application basis. Furthermore, Contiki includes proto threads that provide a thread-like programming abstraction but with a very small memory overhead. Contiki provides IP communication, both for IPv4 and IPv6. Many key mechanisms and ideas from Contiki have been widely adopted within the industry. The IP embedded IP stack, is today used by hundreds of companies in systems such as freighter ships, satellites and oil drilling equipment. Contiki's proto threads, first released in 2005, have been used in many different embedded systems, ranging from digital TV decoders to wireless vibration sensors. Contiki's idea of using IP communication in low-power WSNs has lead to an IETF standard and an international industry alliance - IP for Smart Objects (IPSO) Alliance

(c). Others

There is also other Operating system that can be used by WSNs. For example, SOS is an event-driven OS for mote-class sensor nodes that adopts a more dynamic point on the design spectrum. The prime feature of SOS is its support for loadable modules. A complete system is built from smaller modules, possibly at run-time. To support the inherent dynamism in its module interface, SOS also focuses on supporting dynamic memory management. Unfortunately, SOS is no longer under active development due to the graduation of the core developers. LiteOS is an open source, interactive, UNIX like operating system designed for WSNs. With the tools that come from LiteOS, it is possible to operate one or more WSNs in a Unix-like manner. It is also possible to develop programs for nodes, and wirelessly distribute such programs to sensor nodes.

Vii. Protocol Stack of Wsn

The protocol stack used by the sink, cluster head and sensor nodes are shown in Fig. 3. According to (Akyildiz et al., 2002), the sensor network protocol stack is much like the traditional protocol stack, with the following layers: application, transport, network, data link, and physical. The physical layer is responsible for frequency selection, carrier frequency generation, signal detection, modulation and data encryption. The data link layer is responsible for the multiplexing of data streams, data frame detection, medium access and error control. It ensures reliable point-to-point and point-to-multipoint connections in a communication network. The network layer takes care of routing the data supplied by the transport layer. The network layer design in WSNs must consider the power efficiency, data-centric communication, data aggregation, etc. The transportation layer helps to maintain the data flow and may be important if WSNs are planned to be accessed through the Internet or other external

networks. Depending on the sensing tasks, different types of application software can be set up and used on the application layer. WSNs must also be aware of the following management planes in order to function efficiently: mobility, power, task, quality of service (QoS) and security management planes. Among them, the functions of task, mobility and power management planes have been elaborated in (Akyildiz et al., 2002). The power management plane is responsible for minimizing power consumption and may turn off functionality in order to preserve energy. The mobility management plane detects and registers movement of nodes so a data route to the sink is always maintained. The task management plane balances and schedules the sensing tasks assigned to the sensing field and thus only the necessary nodes are assigned with sensing tasks and the remainder is able to focus on routing and data aggregation. QoS management in WSNs (Howitt et al., 2006) can be very important if there is a real-time requirement with regard to the data services. QoS management also deals with fault tolerance, error control and performance optimization in terms of certain QoS metrics. Security management is the process of managing, monitoring, and controlling the security related behavior of a network. The primary function of security management is in controlling access points to critical or sensitive data. Security management also includes the seamless integration of different security function modules, including encryption, authentication and intrusion detection. (Wang & Zhang, 2008; 2009) in WSNs. It is obvious that networking protocols developed for WSNs must address all five of these management planes.

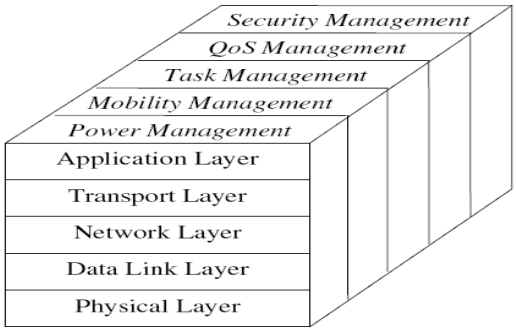

Fig 3 .The Protocol Stack of WSN

Viii. Security Threats in Wsn

Confidentiality means keeping information secret from unauthorized parties. A sensor network should not leak sensor readings to neighboring networks. In many applications (e.g. key distribution) nodes communicate highly sensitive data. The standard approach for keeping sensitive data secret is to encrypt the data with a secret key that only intended receivers possess, hence achieving confidentiality. Since public-key cryptography is too expensive to be used in the resource constrained sensor networks, most of the proposed protocols use symmetric key encryption methods. The creators of tiny Sec argue that cipher block chaining (CBC) is the most appropriate encryption scheme for sensor networks. They found RC5 and Skipjack to be most appropriate for software implementation on embedded microcontrollers. The default block cipher in tiny Sec is Skipjack. SPINS uses RC6 as its cipher [1].

In a sensor network, an adversary can easily inject messages, so the receiver needs to make sure that the data used in any decision-making process originates from the correct source. Data authentication prevents unauthorized parties from participating in the network and legitimate nodes should be able to detect messages from unauthorized nodes and reject them. In the two party communication case, data authentication can be achieved through a purely symmetric mechanism: The sender and the receiver share a secret key to compute a message authentication code (MAC) of all communicated data. When a message with a correct MAC arrives, the receiver knows that it must have been sent by the sender. However, authentication for broadcast messages requires stronger trust assumptions on the network nodes. The creators of SPINS contend that if one sender wants to send authentic data to mutually entrusted receivers, using a symmetric MAC is insecure since any one of the receivers know the MAC key, and hence could impersonate the sender and forge messages to other receivers. SPINS constructs authenticated broadcast from symmetric primitives, but introduces asymmetry with delayed key disclosure and one-way function key chains. LEAP uses a globally shared symmetric key for broadcast messages to the whole group.

However, since the group key is shared among all the nodes in the network, an efficient reeking mechanism is defined for updating this key after a compromised node is revoked. This means that LEAP has also defined an efficient mechanism to verify whether a node has been compromised.

Data integrity ensures the receiver that the received data is not altered in transit by an adversary. Note that Data Authentication can provide Data Integrity also.

Data freshness implies that the data is recent, and it ensures that an adversary has not replayed old messages. A common defense is to include a monotonically increasing counter with every message and reject messages with old counter values. With this policy, every recipient must maintain a table of the last value from every

sender it receives. However, for RAM constrained sensor nodes, this defense becomes problematic for even modestly sized networks. Assuming nodes devote only a small fraction of their RAM for this neighbor table, an adversary replaying broadcast messages from many different senders can fill up the table. At this point, the recipient has one of two options: ignore any messages from senders not in its neighbor table, or purge entries from the table. Neither is acceptable; the first creates a DOS attack and the second permits replay attacks. Some researchers contend that protection against the replay of data packets should be provided at the application layer and not by a secure routing protocol as only the application can fully and accurately detect the replay of data packets Whereas some authors reason that by using information about the network's topology and communication patterns, the application and routing layers can properly and efficiently manage a limited amount of memory devoted to replay detection. Mostly Researchers have identified two types of freshness: weak freshness, which provides partial message ordering, but carries no delay information, and strong freshness, which provides a total order on a request-response pair, and allows for delay estimation. Weak freshness is required by sensor measurements, while strong freshness is useful foretime synchronization within the network [26].

The sensor network should be robust against various security attacks, and if an attack succeeds, its impact should be minimized. The compromise of a single node should not break.

IX. APPLICATIONS

Area	Applications
Military[17]	Detection of enemy unit movements on land and sea. Battle field surveillances. Sensing intruders on basis.
Industrial applications[8]	Inventory Tracking In-Process Parts Tracking Automated Problem Reporting RFID – Theft Deterrent and Customer Tracing Plant Equipment Maintenance Monitoring Factory process control and Industrial automation. Monitoring and control of industrial equipment
Precision agriculture and animal tracking [13]	Environmental monitoring of water and soil . Insect disease and weed monitoring. Habitual monitoring. Observation of biological and artificial systems.
Environmental monitoring [3]	Disaster management. Fire/water detectors. Hazardous chemical level and fires. To reveal unknown tracks and leakages within underground pipelines

Security and surveillance	To monitor and restrict the intruders into the prohibited area.
Entertainment[18]	To maintain the capacity of the shopping mall 3D visualization for games on cellular phones Interactive role-playing games with sensors
Health care (health monitoring, medical diagnostics) [25]	To monitor the health condition (Glucose ,Heart rate ,Cancer detection) and identifying the chronic diseases (artificial retina , cochlear implants)
Smart grids and energy control systems	Monitoring and control of energy and grid components. Self-healing energy network automation
Smart buildings	Security and safety. Heating, Ventilation and air conditioning systems (HVAC).

X. Future Work

To formulate Data Analysis algorithm for intelligent applications which can sense events, send data to a remote center for analysis and receive a response in the form of information to assist in a decision or initiate an action.

X. Conclusion

This paper deals about the survey on Wireless Sensor Networks (WSN) and their technologies, standards and applications. The recent researches are taking place in designing the sensor nodes with energy harvesting for many real time applications and to resolve the Security threats in WSN. The flexibility, fault tolerance, high sensing fidelity, low cost, and rapid deployment characteristics of sensor networks create many new and exciting application areas for remote sensing. In the future, this wide range of application areas will make sensor networks an integral part of our lives. However, realization of sensor networks needs to satisfy the constraints introduced by factors such as fault tolerance, scalability, cost, hardware, topology change, environment, and power consumption.

REFERENCES

[1] Adrian Perrig, John Stankovic and David Wagner, "Security in Wireless Sensor Networks", Communications of the ACM, Vol. 47, No.5, pp. 53-57, June 24.

[2] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y. & Cayirci, E. (2002). A survey on sensor networks, *IEEE Communications Magazine* **40**(8): 102–114.

[3] Barrenetxea, G., Ingelrest, F., Schaefer, G. & Vetterli, M. (2008). Wireless sensor networks for environmental monitoring: The sensorscope experience, *Proc. of 20th IEEE International*

[4] Chong .C.Y & Kumar, S.P (2003) Sensor networks: Evolution, opportunities, and challenges, *Proceedings of the IEEE* 91(8): 1247–1256.

[5] Connolly, M. & O'Reilly, F. (2005) Sensor networks and the food industry, Proc. of the Workshop on Real-World Wireless Sensor Networks.
[6] Garcia Hermandez, Carlos Felipe; Villanueva-Cruz, Jose Alonso; 'Security in AODV Protocol routing for Mobile Adhoc Networks', IEEE ROC & C,2005, C-03, P-11, Acapulco Gro Mexico , 29/November-04/December 2009.
[7] Hill, J. L. (2003). *System Architecture for Wireless Sensor Networks*, PhD thesis, Doctor of Philosophy in Computer Science, University of California at Berkeley, USA.
[8] Howitt, I., Manges, W. W., Kuruganti, P. T., Allgood, G., Gutierrez, J. A. & Conrad, J. M.(2006). Wireless industrial sensor networks: Framework for qos assessment and qos management, *ISA Transactions* **45**(3): 347–359.I*EEE 802.15 WPAN Task Group 4*
[9] Information Processing and Routing in Wireless Sensor Networks, World Scientific Publishing Co. Pte Ltd.,http://www.worldscibooks.com/compsci/6288.html
[10] International Telecommunications Union – Telecommunications (ITU-T), *Ubiquitous Sensor Networks (USN), ITU-T Technology Watching Brief Report Series, No. 4*, (February 2008)
[11] Kumar, S. & Shepherd, D. (2001). Sensit: Sensor information technology for the war fighter, *Proc. of the 4th International Conference on Information Fusion (FUSION'01)*, pp. 3–9 (TuC1).
[12] Luis Javier Garcia Villalba *, Ana Lucila Sandoval Orozco, Alicia Triviño Cabrera and Cláudia Jacy Barenco Abbas," Routing Protocols in Wireless Sensor Networks", Sensors 2009, 8399-8421
[13] Luca Bencini, Davide Di Palma, Giovanni Collodi and Gianfranco Manes Department of Electronics and Telecommunications - University of Florence Italy Antonio Manes Netsens S.r.l.Italy Wireless Sensor Networks for On-field Agricultural Management Process. Wireless Sensor Networks: Application-Centric Design, Published by InTech
[14] LiteOS (n.d.). http://www.liteos.net.
[15] Myers, C., Oppenheim, A., Davis, R. & Dove, W. (1984).Knowledge-based speech analysis and enhancement, Proc. of the International Conference on Acoustics, Speech and Signal Processing.
[16] Ni, L.M. (2008). China's national research project on wireless sensor networks, Proc. of the 2008IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'08), p. 19.
[17] Pister, K. S. J. (2000). Military applications of sensor networks, of Institute for Defense Analyses Paper P-3531, Defense Science Study Group. Proceedings if the Distributed Sensor Nets Workshop (1978) Pittsburgh, USA. Department of Computer Science, Carnegie Mellon University
[18] Peter Corke, Fellow IEEE, Tim Wark, Member IEEE, Raja Jurdak, Member IEEE, Wen Hu, Member IEEE, Philip Valencia, Member IEEE, and Darren Moore, Member IEEE, Environmental Wireless Sensor Networks, Vol. 0018-9219/$26.00 _2010 IEEE 98, No. 11, November 2010 | Proceedings of the IEEE TinyOS Community Forum.
[19] Rashid, R. & Robertson, G. (1981). Accent: A communication oriented network operating system kernel, *Proc. of the 8th Symposium on Operating System Principles*, pp. 64–75.*Rohrback Cosasco Systems* (n.d.). http://www.cosasco.com.
[20] Saurabh Mehta, Ju-A Lee, Jae-Hyum Kim, "IS-MAC Based Flooding Protocol for Sensor Networks", PE-WASUN´05, Montreal, Quebec, Canada, October 13, 2005, pp. 79-83
[21] Sabbah, E., Majeed, A., Kang, K., Liu, K. & Abu Ghazaleh,N.(2006).(n.d.). https://projects.nesl.ucla.edu/public/sos-2x/doc/.
[22] Steere, D., Baptista, A., McNamee, D., Pu, C. & Walpole, J. (2000). Research challenges in environmental observation and forecasting systems, Proc. of 6th International Conference on Mobile Computing and Networking (MOBICOMM'00), pp. 292–299.
[23] Sudevalayam, S. & Kulkarni, P. (2008). Energy Harvesting Sensor Nodes: Survey and Implications, Technical Report TR-CSE-2008-19, Department of Computer Science and Engineering, Indian Institute of Technology Bombay.
[24] TinyOS Community Forum (n.d,).An application driven perspective on wireless sensor network security, Proceedings of the 2nd ACM Workshop on QoS and Security for Wireless and Mobile Networks, SOS Embedded Operating System. http://www.tinyos.net
[25] G. Virone et al., An Advanced Wireless Sensor Network for Health Monitoring, Department of Computer Science, University of Virginia, www.cs.virginia.edu/papers/d2h206-health.pdf, retrieved 5 October 2010
[26] Wang, Q. & Zhang, T. (2008). Sec-snmp: Policy-based security management for sensor networks, Proc. of the International Conference on Security and Cryptography (SECRYPT'08),in conjunction with ICETE 2008.
[27] ZigBee Alliance (n.d.). http://www.zigbee.org.
[28] A. Zerger, R. Viscarra Rossel, D. Swain, T. Wark, R. Handcock, V. Doerr, G. Bishop-Hurley, E. Doerr, P. Gibbons, and C. Lobsey, "Environmental sensor networks for vegetation, animal and soil sciences," *Journal of Applied Earth Observation and Geo Information*, vol.12, no. 5, pp. 304-313, Oct 2010.