International Journal of Recent Research and Review, Vol. VII, Issue 3, September 2014
ISSN 2277 – 8322

Wireless Sensor Network: Protocols and Standards

[1]Sunita Gupta,[2]K.C.Roy,Sakar Gupta[3]
[1]CSE Deptt., JECRC University, Jaipur
[2]ECE Deptt. Poornima University Jaipur
[3]ECE, Deptt.KITE Jaipur India.
gupta_1982sunita@yahoo.com

***Abstract:*-A wireless sensor network (WSN) has important applications such as remote environmental monitoring and target tracking. This has been enabled by the availability, particularly in recent years, of sensors that are smaller, cheaper, and intelligent. These sensors are equipped with wireless interfaces with which they can communicate with one another to form a network. The design of a WSN depends significantly on the application, and it must consider factors such as the environment, the application's design objectives, and cost, hardware, and system constraints. Wireless sensor networks (WSNs) have gained worldwide attention in recent years, particularly with the proliferation in Micro-Electro-Mechanical Systems (MEMS) technology which has facilitated the development of smart sensors. These sensors are small, with limited processing and computing resources, and they are inexpensive as compared to traditional sensors. These sensor nodes can sense, measure, and gather information from the environment and, based on some local decision process, they can transmit the sensed data to the user. Smart sensor nodes are low power devices equipped with one or more sensors, a processor, memory, a power supply, a radio, and an actuator. A variety of mechanical, thermal, biological, chemical, optical, and magnetic sensors may be attached to the sensor node to measure properties. The goal of our survey is to present a comprehensive review of the literature about composition of sensor nodes, its characteristics, challenges, applications, different standards, manufactures, protocols and tools used etc.**

***Keywords:*-Wireless sensor network, Energy efficient target coverage, Energy minimization, Lifetime of WSN, Network architecture, Cover set, Coverage, Connectivity.**

I. INTRODUCTION

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation, and traffic control. Since the sensor nodes have limited memory and are typically deployed in difficult-to-access locations, a radio is implemented for wireless communication to transfer the data to a base station (e.g., a laptop, a personal handheld device, or an access point to a fixed infrastructure).Battery is the main power source in a sensor node. Secondary power supply that harvests power from the environment such as solar panels may be added to the node depending on the appropriateness of the environment where the sensor will be deployed. WSNs have great potential for many applications in scenarios such as military target tracking and surveillance [1-2], natural disaster relief [3], biomedical health monitoring [4-5], and hazardous environment exploration and seismic sensing [6].

II. ADVANTAGES OF WIRELESS SENSOR NETWORK

Wireless sensor network have many features like very large number of nodes, Asymmetric flow of information, limited amount of energy, low cost, small size and weight per nodes, broadcast communication instead of point to point etc. Based on this Wireless sensor network have various advantages like:-

- Easy to deploy.
- Enhanced flexibility.
- Reduced cabling.
- Mobility and ease to network configuration.
- Location tracking of mobile equipments.
- Increase assets utilizations.
- Low power.
- Reduced inventory.
- Reduced deployment costs.
- Decreased maintenance costs.

III. APPLICATIONS

The applications for WSNs are many and varied, but typically involve some kind of monitoring, tracking, and controlling. Specific applications for WSNs include habitat monitoring, object tracking, nuclear reactor control, fire detection, and traffic monitoring. In a typical application, a WSN is scattered in a region where it is meant to collect data through its sensor nodes.

Area monitoring

Area monitoring is a common application of WSNs. In area monitoring, the WSN is deployed over a region where some phenomenon is to be monitored. For example, a large quantity of sensor nodes could be deployed over a battlefield to detect enemy intrusion instead of using landmines. When the sensors detect the event being monitored (heat, pressure, sound, light, electro-magnetic field, vibration, etc), the event needs to be reported to one of the base stations, which can take appropriate action (e.g., send a message on the internet or to a satellite).

Environmental monitoring

A number of WSN deployments have been done in the past in the context of environmental monitoring. Many of these have been short lived, often due to the prototypical nature of the projects.

Industrial Monitoring

1) Water/Wastewater Monitoring There are opportunities for using wireless sensor networks within the water/wastewater industries. Facilities not wired for power or data transmission can be monitored using industrial wireless input-output devices and sensors powered using solar panels or battery packs.

II) Landfill Ground Well Level Monitoring and Pump Counter

Wireless sensor networks can be used to measure and monitor the water levels within all ground wells in the landfill site and monitor leach-ate accumulation and removal. A wireless device and submersible pressure transmitter monitors the leach-ate level. The sensor information is wirelessly transmitted to a central data logging system to store the level data, perform calculations, or notify personnel when a service vehicle is needed at a specific well.

IV. CHALLENGES ON WSN

Managing a wide range of application types in a WSN is hardly possible with a single conception and design of the wireless network. However, certain attributes identifies are related to characteristic requirements and the mechanisms of such system. The realization of these characteristics with newer mechanisms is the major challenge foreseen to WSNs. Many current WSN solutions are developed with simplifying assumptions about wireless communication and the environment, even though the realities of wireless communication and environmental sensing are well known. Many of these solutions work very well in simulation.

- WSN nodes have very restricted computational and storage power.
- Node communication range is limited. In most cases nodes can directly communicate with immediate neighbors only.

- WSN consists of a large number of unreliable nodes.
- WSN must continue to operate at all times even when some of it nodes get physically destroyed at unpredictable times.
- WSN must continue to operate without interruption when new nodes are added to the network in order to replace the failed ones or extend the network.
- Nodes may temporarily stop processing due to power shortage and come back to life when power is restored.
- Information collected by nodes may be unavailable at some irregular times due to the bad quality of radio link.
- Node communication may require different paths at different times depending on the state of end- to-end link between communicating parts of the network.
- Nodes must be able to communicate with the rest of the world represented by traditional LAN.

V. PROTOCOLS FOR WSN

There are several protocols proposed for WSNs (Wireless Sensor Network). From [7], the MAC (Medium Access Control) layer reacts to this probabilistic reception information by adjusting the number of acknowledgments and/or retransmissions. It is observed that an optimal route discovery protocol cannot be based on a single retransmission by each node, because such a search may fail to reach the destination or find the optimal path. Gaining neighbor knowledge information with “hello” packets is not a trivial protocol. It is described the localized position-based routing protocols that aim to minimize the expected hop count (in case of hop-by-hop acknowledgments and fixed bit rate) or maximize the probability of delivery (when acknowledgments are not sent).An interesting open problem for future research is to consider physical-layer-based routing and broadcasting where nodes may adjust their transmission radii. Expected power consumption may then be considered a primary optimality measure. Further research should address other problems in the design of network layer protocols. For instance, if we consider a more dynamic and realistic channel model, such as multi-path fading, the estimated number of packets may suffer from large variance, and the described protocols may need some adjustments. More realistic interference models can be added, and transport layer protocols also need to be adjusted [39]. In [40], a survey of state-of-the-art routing techniques in WSNs is presented. Overall, the routing techniques were classified into three categories based on the underlying network structure: flit, hierarchical, and location-based routing. Furthermore, these protocols could be classified into multipath-based, query-based, negotiation-based, QoS based, and coherent-based depending on the protocol operation. Advantages and performance issues of each routing technique were highlighted [8].From [9], when compared with now classical MANETs (Mobile Ad hoc Networks), sensor networks have different characteristics, and present different design and engineering challenges. One of the main aspects of sensor networks is that the solutions tend to be very application specific. For this reason, a layered view like the one used in OSI imposes a large penalty, and implementations more geared toward the particular are desirable. Communication, which is the most energy-costly aspect of the network, can be organized in three fundamentally different ways: node-centric, data-centric, and position centric. Node-centric communication is the most popular and well understood paradigm, being currently used in the Internet. The other two, data-centric and position-centric, are more scalable, better adaptable to applications, and conceptually more appropriate in many cases, and therefore may successfully challenge the node-centric way of looking at the sensor networks. Data-centric approaches, on the other hand, tend to provide a top-to-bottom solution, as is the case with directed diffusion. In fact, directed diffusion solves only one problem, but solves it right. A new IEEE standard, 802.15.4, is aimed at low-power low-distance communication devices that may allow years of battery life.

VI. MODEL, FRAMEWORK AND SIMULATION TOOLS FOR WSN

Together with the development of simulation tools for WSN, their corresponding models have been introduced. The models include new components, not present in classical network simulators, as detailed power and energy consumption models or environment models. Widespread research on WSN has raised a race involving many simulation tools and frameworks. The selection of a simulation framework for any type of network is a task that is worth to spend enough time. Indeed, this is particularly true for wireless sensors nets, because of the diversity and complexity of the simulation scenarios, protocols, and elements involved. In such a heterogeneous scope, different evaluation tools achieve different goals. In a first step, existing WSN frameworks can be categorized in: (a) Specific add-ons to general purpose communication networks and (b) WSN frameworks built from scratch. Usually, the key properties to select suitable simulation environment are:

1) Reusability and availability.
2) Performance and scalability.
3) Support for rich-semantics scripting languages to define experiments and process results.
4) Graphical, debug and trace support.

A simulation framework usually consists of a basic simulation library, a utility library, and some scripting support. The actual form the package is deployed depends on the implementation. Some packages provide tools that translate model scripts into objects in the implementation language to be compiled afterwards. Other packages bind library and scripting so that simulation objects can be instantiated from a script. Others provide a visual interface. A broad variety of different simulation tools are used to simulate key characteristics of Wireless Sensor Networks. They range from emulator originated tools like Avrora andTOSSIM to wireless and mobile communication simulation environments, like OMNeT++,OPNET and NS-2. Each of these classes and tools has its specific advantages and disadvantages and often the selection of the tool is mainly based on the experience of the researcher rather than on rational arguments. An overview of the different tools and simulation environments with their particular pros and cons has been established by the CRUISE project [10] and is given in [11].

The most relevant simulation environments used to study WSN are introduced and their main features and implementation issues are also described.

• *NS-2 [12]:* Discrete event simulator developed in C++.NS-2 is one of the most popular non-specific network simulators, and supports a wide range of protocols in all layers. It uses OTcl [13] as configuration and script interface.NS-2 is the paradigm of reusability. It provides the most complete support of communication protocol models, among non-commercial packages. Regarding WSN, NS-2 includes ad-hoc and WSN specific protocols such as directed diffusion [14] or SMAC [15]. Also, several projects intend to provide WSN support to NS-2 such as SensorSim [16] and NRL [17]. Both are extensions of NS-2 to support WSN modeling. However, SensorSim seems to be no longer available at [18]. NS-2 can comfortably model wired network topologies up to 1,000 nodes or above with some optimizations. This experiment size can be kept for wireless topologies using some new optimizations [19]. A disadvantage of NS-2 is that it provides poor graphical support, via Nam. This application just reproduces a NS-2 trace.NS-2 has been an essential testing tool for network research and, so, one could expect that the new *conventional* protocols will be added to future releases.

• *OMNET++ [20]:* Modular discrete event simulator implemented in C++. Getting started with it is quite simple, due to its clean design. OMNET++ also provides a powerful GUI library for animation and tracing and debugging support. Its major drawback is the lack of available protocols in its library, compared to other simulators. However, OMNET++ is becoming a popular tool and its lack of models is being cut down by recent contributions. For instance, a mobility framework has recently been released for OMNET++ [21], and it can be used as a starting point for WSN modeling.

Additionally, several new proposals for localization and MAC protocols for WSN have been developed with OMNET++, under the Consensus project [22], and the software is publicly available. Nevertheless, most of the available models have been developed by independent research groups and don't share a common interface, what makes difficult to combine them.

• *J-Sim [23]:* A component-based simulation environment developed entirely in Java. It provides real-time process based simulation. The main benefit of J-Sim is its considerable list of supported protocols, including a WSN simulation framework with a very detailed model of WSNs, and a implementation of localization, routing and data diffusion WSN algorithms [24]. J-Sim models are easily reusable and interchangeable offering the maximum flexibility. Additionally, it provides a GUI library for animation, tracing and debugging support and a script interface, named Jacl [25].

• *NCTUns2.0 [26]:* Discrete event simulator whose engineis embedded in the kernel of a UNIX machine. The actual network layer packets are tunnellzed through virtual interfaces that simulate lower layers and physical devices. This notable feature allows simulations to be fed with real program data sources. A useful GUI is available in addition to a high number of protocols and network devices, including wireless LAN. Unfortunately, no specific designs for WSN are included. On one hand, the close relationship between the simulation engine of NCTUns2.0 and the Linux kernel machine seems a difficulty (adding WSN simulation modules to this architecture is not a straightforward task). But, on the other hand, real sensor data can be easily plug into simulated devices, protocols and actual applications, just by installing these sensors in the machine.NCTUns2.0 also has worthy graphical edition capabilities.

• *JiST/SWANS [27]:* Discrete event simulation framework that embeds the simulation engine in the Java byte code. Models are implemented in Java and compiled. Then, byte codes are rewritten to introduce simulation semantics. Afterwards, they are executed on a standard JVM. This implementation allows the use of unmodified existing Java software in the simulation, as occurs with NCTUns2.0 and UNIX programs. The main drawback of JiST tool, is the lack of enough protocol models. At the moment it only provides an *ad-hoc* network simulator called SWANS, built atop JiST engine, and with a reduced protocol support. The only graphical aid is an event logger. Python [28] is used as a scripting engine. JiST claims to scale to networks of 106 wireless nodes with two and one order of magnitude better performance (execution time) than NS-2 and GloMoSim respectively. It has been also shown that it outperforms Glo-MoSim and NS-2 in event throughput and memory consumption, despite/ being built with Java. Parsec is a simulation language derived from C that adds semantics for creating simulation entities and message communication on variety of parallel architectures. Taking advantage of parallelization, it has been shown to scale to 10,000 nodes [29].

• *TOSSIM [30]:* Bit-level discrete event simulator and emulator of TinyOS, i.e. for each transmitted or received bit an event is generated instead of one per packet. This is possible because of the reduced data rate (around 40 kbps) of the wireless interface. TOSSIM simulates the execution of nesC code on a TinyOS/MICA, allowing emulation of actual hardware by mapping hardware interruptions to discrete events. A simulated radio model is also provided. Emulated hardware components are compiled together with real TinyOS components using the nesC compiler. Thus, an executable with real TinyOS applications over a simulated physical layer is obtained. Additionally, there are also several communication services that provide away to feed data from external sources. The result is a high fidelity simulator and emulator of a network of TinyOS/MICA nodes. The goal of TOSSIM is to study the behavior of TinyOS and its applications rather than performance metrics of some new protocol. Hence, it has some limitations, for instance, it does not capture energy consumption. Another drawback of this framework is that every node must run the same code. Therefore, TOSSIM cannot be used to evaluate some types of heterogeneous applications. TOSSIM can handle simulations around a thousand of Motes. It is limited by its bit-level granularity: Performance degrades as traffic increases. Channel sampling is also simulated at bit level and consequently the use of a CSMA protocol causes more overhead than would do a TDMA one.

• *Prowler/JProwler [31]:* A discrete event simulator running under MATLAB intended to optimize network parameters. Prowler is a version of Prowler developed in Java.

• *SNAP [32]:* SNAP is defined as an integrated hardware simulation-and deployment platform. It is a microprocessor that cans be used in two ways: (1) As the core of a deployed sensor or (2) as part of an array of processors that performs parallel simulation. Again, "real" code for sensors can be simulated. By combining arrays of SNAPs (called Network on a Chip), it is claimed to be able to simulate networks on the order of 100,000 nodes.

VII. STANDARDS OF WSN

From [33], while most ongoing work in IEEE 802wireless working groups is geared to increase data rates, throughput, and QoS, the 802.15.4 LR-WPAN (Low rate-Wireless Personal Area Network) task group is aiming for other goals. The focus of 802.15.4 is on very low power consumption, very low cost, and low data rate to connect devices that previously have not been networked, and to allow applications that cannot use current wireless specifications. Working within a standards organization to develop a wireless solution has the advantage of bringing developers and users of such a technology together in order to define a better solution. The work also fosters high-level connectivity to other types of networks and enables low-volume products that do not justify a proprietary solution to be wirelessly connected. Two physical layer specifications were chosen to cover the 2.4 GHz worldwide band and the combination of the868 MHz band in Europe, the 902 MHz band in Australia, and the 915 MHz band in the United States. Both physical layers are direct sequence spread spectrum (DSSS). The efforts of the IEEE 802.15.4 task group will bring one step closer to the goal of a wirelessly connected world [34].Coexistence among diverse collocated devices in the 2.4 GHz band is an important issue in order to ensure that each wireless service maintains its desired performance requirements. On the other hand, from , the IEEE 1451, a family of Smart Transducer Interface Standards, describes a set of open, common, network-independent communication interfaces for connecting transducers (sensors or actuators)to microprocessors, instrumentation systems, and control/field networks. The key feature of these standards is the definition of a TEDS (Transducer Electronic Data Sheet). The TEDS is a memory device attached to the transducer, which stores transducer identification, calibration, correction data, and manufacture-related information. The goal of IEEE 1451 is to allow the access of transducer data through a common set of interfaces whether the transducers are connected to systems or networks via a wired or wireless means. The family of IEEE 1451 standards is sponsored by the IEEE Instrumentation and Measurement Society's SensorTechnology Technical Committee. IEEE P1451.5 defines a transducer-to-NCAP (NetworkCapable application Processor) interface and TEDS for wireless transducers. Wireless standards such as 802.11(WiFi), 802.15.1 (Bluetooth), 802.15.4 (ZigBee) are being considered as some of the physical interfaces.

VIII. REFERENCES

[1] G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J.Sallai, K. Frampton, Sensor network-based countersniper system, in:Proceedings of the Second International Conference on Embedded Networked Sensor Systems (Sensys), Baltimore, MD, 2004.

[2] J. Yick, B. Mukherjee, D. Ghosal, Analysis of a Prediction-based Mobility Adaptive Tracking Algorithm, in: Proceedings of the IEEE Second International Conference on Broadband Networks (BROADNETS), Boston, 2005.

[3] M. Castillo-Effen, D.H. Quintela, R. Jordan, W. Westhoff, W. Moreno, Wireless sensor networks for flash-flood alerting, in: Proceedings of the Fifth IEEE International Caracas Conference on Devices, Circuits, and Systems, Dominican Republic, 2004.

[4] T. Gao, D. Greenspan, M. Welsh, R.R. Juang, A. Alm, Vital signs monitoring and patient tracking over a wireless network, in: Proceedings of the 27th IEEE EMBS Annual International Conference, 2005.

[5] K. Lorincz, D. Malan, T.R.F. Fulford-Jones, A. Nawoj, A. Clavel, V.Shnayder, G. Mainland, M. Welsh, S. Moulton, Sensor networks for emergency response: challenges and opportunities, Pervasive Computing for First Response (Special Issue), IEEE Pervasive Computing, October–December 2004.

[6] G. Wener-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, M. Walsh, Deploying a wireless sensor network on an active volcano, Data-Driven Applications in Sensor Networks (Special Issue), IEEE Internet Computing, March/April 2006.

[7] Ivan Stojmenovic, Amiya Nayak, Johnson Kuruvila, "DesignGuidelines for Routing Protocols in Ad Hoc and Sensor Networkswith a Realistic Physical Layer"; *IEEE Communications Magazine,*pp. 101-106, March 2005.

[8] Jamal N. Al-Karaki, Ahmed E. Kamal; "Routing Techniques inWireless Sensor Networks: A Survey"; *IEEE WirelessCommunications*, pp. 6-28, December 2004.

[9] Dragos, Niculescu; "Communication Paradigms for SensorNetworks"; *IEEE Communications Magazine*, pp. 116-122, March2005.

[10] CRUISE Network of Excellence Web Portal, http://www.ist-cruise.eu, last accessed October 2nd, 2007.

[11] http://www.ist-cruise.eu /cruise/Public%20documents /wp123-sn-simulation-tool-knowledgebase/, last accessed April 1st, 2007 *2.1 – OMNeT++*

[12] The Network Simulator, NS–2 [Online]. Available:http://www.isi.edu/nsnam/ns/

[13] MIT Object Tcl. [Online].Available: http: //bmrc.berkeley.edu/research/cmt/cmtdoc/otcl

[14] C. Intanagonwiwat, R. Govidan, D. Estrin, J. Heidemann,F. Silva, "Directed diffusion for wireless sensor networking."*IEEE/ACM Transactions on Networking*, vol. 11,issue 1, pp. 2–16, February 2003.

[15] W. Ye, J. Heidemann, D. Estrin, "Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks." *ACM/IEEE Transactions on Networking*, vol. 12, pp. 493–506, 2004.

[16] S. Park, A. Savvides, M. B. Srivastava. "SensorSim: A Simulation Framework for Sensor Networks." In *Proc.ACM Modeling, Analysis and Simulation of Wireles andMobile Systems (MSWiM 2000)*, Boston, MA, pp. 104–111, August 2000

[17] NRL's Sensor Network Extension to NS-2 [Online]. Available:http://nrlsensorsim.pf.itd.nrl.navy.mil/

[18] SensorSim: A simulation framework for sensor networks.[Online].Available: http://nesl.ee.ucla.edu/projects/sensorsim/

[19] V. Naoumov, T. Gross, "Simulation of Large Ad Hoc Networks." In *Proc. ACM Modeling, Analysis and Simulation of Wireles and Mobile Systems (MSWiM 2003)*, San Diego, CA, pp. 50–57, 2003.

[20] OMNET++ discrete event simulator. [Online]. Available: http://www.omnetpp.org

[21] Mobility Framework for OMNET++. [Online]. Available: http://mobility-w.sourceforge.net

[22] Consensus: Collaborative Sensor Networks [Online]. Available: http://www.consensus.tudelft.nl

[23] J-Sim [Online]. Available: http://www.j-sim.org

[24] A. Sobeih, W. Chen, J. C. Hou, L. Kung, N. Li, H. Lim, H. Tyan, H. Zhang, "J-Sim: A simulation andemulation environment for wireless sensor networks." In *Proc. Annual imulation Symposium (ANSS 2005)*, SanDiego, CA, pp. 175–187, April 2005.

[25] Jacl, Java implementation of Tcl8.x. [Online].Available: http://www.tcl.tk/software/java

[26] NCTUns 2.0 Network Simulator and Emulator. [Online].Available: http://nsl.csie.nctu.edu.tw/nctuns.html

[27] R. Barr, Z. J. Haas, R. van Renesse, "JiST: Embedding Simulation Time into a Virtual Machine." In *Proc. 5thEUROSIM Congress on Modeling and Simulation*, Paris,France, september 2004

[28] Jython. [Online]. Available: http://www.jython.org

[29] M. Takai, R. Bagrodia, K. Tang, M. Gerla, "Efficient Wireless Networks Simulations with Detailed Propagations Models." *Kluwer Wireless Networks*, 7, pp. 297–305, 2001.

[30] P. Levis, N. Lee, M. Welsg, D. Culler, "TOSSIM:Accurate and Scalable Simulation of Entire TinyOS Applications." In *Proc. 1st ACM Int. Conf.*

Embedded Networked Sensor Systems (SenSys), Los Angeles, CA, pp. 126–137, 2003.

[31] PROWLER: Probabilistic Wireless Network Simulator.[Online].Available: http://www.isis.vanderbilt.edu/projects/nest/prowler

[32] C. Kelly, V. Ekanayake, R. Manohar, "SNAP: A Sensor-Network Asynchronous Processor." In *Proc. 9th ACM Int. Symposium on Asynchronous Circuits and Systems*,Washington, DC, pp. 24, 2003.

[33] José A. Gutierrez, Marco Naeve, Ed Callaway, Monique Bourgeois,Vinay Mitter, Bob Heile, "IEEE 802.15.4: A Developing Standardfor Low-Power Low-Cost Wireless Personal Area Networks"; *IEEENetwork*, pp. 12-19, September/October 2001.

[34] (2006, October). IEEE 1451 Family of Smart Transducer InterfaceStandards [Online]. Available:http://grouper.ieee.org/groups/1451/0/body %20frame_files/Familyof-51_handout.htm