International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Analysis of Flooding and Directed Diffusion Protocol

Deepesh Dewangan[1], Srikant Singh[2]

[1]Department of Computer Science & Engineering, Kalinga University, Naya Raipur, Chhattisgarh-India
deepesh.dewangan@gmail.com

[2] Departmentof Computer Science & Engineering, Kalinga University, Naya Raipur, Chhattisgarh-India
srikant.singh6@gmail.com

Abstract: ***Wireless Sensor Networks (WSNs) consist of small nodes with sensing, computation, and wireless communications capabilities. Many routing, power management, and data dissemination protocols have been specifically designed for WSNs where energy awareness is an essential design issue. The focus, however, has been given to the routing protocols which might differ depending on the application and network architecture. This paper surveys the flooding and Directed Diffusion routing protocols for sensor networks and presents a classification for the various approaches pursued. The main categories explored in this paper are data-centric.Each routing protocol is described and discussed under the appropriate category. Moreover, protocols using contemporary methodologies such as network flow, average delay and energy are also discussed.***

Keywords: about Routing, WSN

1. Introduction

It is common to see embedded systems in several aspects of people‟s lives. Embedded systems control not only transportation and communication infrastructures, but also appliances in home and office environments (*e.g.*, refrigerators, VCRs, TVs, microwave ovens)[1][2][3]. Embedded systems are the first step toward ubiquitous computing whereby various computing elements are so seamlessly integrated into the environment that they will be invisible to common awareness. This vision is increasingly likely to be realized with recent advances in CMOS IC, wireless communication, and MEMS [Cen] technology[1][7]. Such advances have already led to dramatic reductions in size, power consumption, and circuitry cost. Various functions (*e.g.*, sensing, signal processing) can now be integrated into a single wireless node. Coordination and communication among such nodes will not only enable seamless computing but also revolutionize information technology, especially applications related to sensing and controlling physical environments. Small active devices or sensors can coordinate to perform larger sensing tasks (*i.e.*, Distributed micro-sensing tasks), which could not have been achieved with individual node capabilities.

Several thousands of such devices may be deployed in hostile dynamic environments (*e.g.*, toxic terrain) or in more benign, but less accessible, environments (*e.g.*, large complex industrial plants, aircraft interiors). In distributed micro-sensing systems, sensors can be dispersed throughout a terrain to collect information about the physical environment[1][5]. Users may be interested in receiving certain collected information from a particular region or from the entire network (*e.g.*, a periodic location of an animal in region *A*). The users may express their *interest* simply by querying information *sinks*, which could be any sensors nearby. The specified queries need to be delivered to the sensors in the specified region, which will be *tasked* to collect information. Once such sensors detect the animal, they might coordinate with one another to triangulate the animal‟s location and disseminate it back to the sinks or the users. In this thesis, we propose *directed diffusion* for efficient and robust dissemination of query and information in directed diffusion, a node requests data by sending interests for named data. Interest propagation leaves traces (or direction state) so that data, which match the interest, can be "drawn" toward that node. Intermediate nodes can process (*e.g.*, cache, aggregate, transform) or direct data and interests using application knowledge to conserve system resources (*e.g.*, energy, bandwidth). Using the directed diffusion paradigm, the example application might be implemented as follows. The query would be transformed into an interest and *diffused* (*e.g.*, broadcasted, geographically routed) toward nodes in region *A*. When a node in region *A* receives the interest, it activates its sensors to start collecting information about the animal. When the sensors detect the animal‟s location, the node disseminates the information by reversing the path of interest propagation. Intermediate nodes along the path might *aggregate* the data to reduce the total data size for energy savings.

Directed diffusion is dramatically different from IP-style communication whereby inter-node communication is layered on an end-to-end delivery service, and packets are associated with node identifiers of end-points. Unlike traditional networks, diffusion-based sensor networks are data-centric. In directed diffusion, communication primitives are expressed in terms of named data rather than node addresses. Attributes of the sensed event are used to name data in directed diffusion.

2. Sensor Nodes

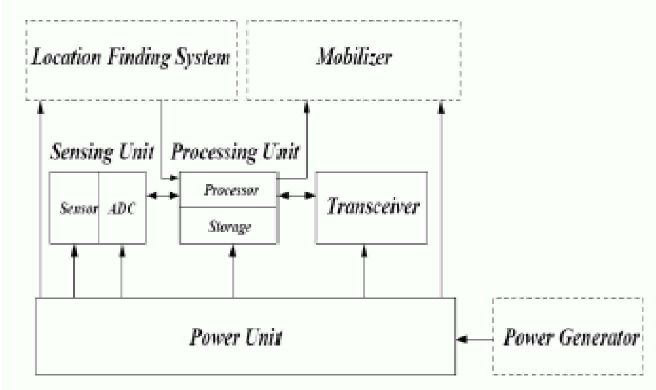

Figure 1: Wireless sensor node

Many recent advances in chip integration technology will enable matchbox sized sensor nodes equipped with a battery, a power-conserving CPU (several hundred MHz), a memory (several tens of Mbytes) , a wireless device using a diversity coding scheme , and an energy efficient MAC (*e.g.*, TDMA). Each node will be capable of running a stripped-down version of a modern operating system (*e.g.*, Windows CE, uCLinux). Sensors (*e.g.*, seismic geophones, infrared dipoles, electrets microphones for acoustic sensing) will be integrated into each node for monitoring various physical conditions (*e.g.*, temperature, pressure, humidity, motion, noise, light). Each node will possibly contain a fully functional GPS receiver and an analog-to-digital conversion system, which can produce ≤ 70 ksamples per second at ≤ 12-bit resolution .For power conservation reasons, some common signal processing functions may be separated from the main unit and moved to a low power ASIC so that the main processor needs to be woken up only when interesting events occur .

3. Wireless Sensor Network model

Unlike their ancestor ad-hoc networks, WSNs are resource limited, they are deployed densely, they are prone to failures, the number of nodes in WSNs is several orders higher than that of ad hoc networks, WSN network topology is constantly changing, WSNs use broadcast communication mediums and finally sensor nodes don"t have a global identification tags [8]. The major components of a typical sensor network is shown in Figure 2. are

Sensor Field: A sensor field can be considered as the area in which the nodes are placed.**Sensor Nodes:** Sensors nodes are the heart of the network. They are in-charge of collecting data and routing this information back to a sink.

Sink: A sink is a sensor node with the specific task of receiving, processing and storing data from the other sensor nodes. They serve to reduce the total number of messages that need to be sent, hence reducing the overall energy requirements of the network. Sinks are also known as data aggregation points.

Task Manager: The task manager also known as base station is a centralized point of control within the network, which extracts information from the network and disseminates control information back into the network. It also serves as a gateway to other networks, a powerful data processing and storage centre and an access point for a human interface. The base station is either a laptop or a workstation. Data is streamed to these workstations either via the internet, wireless channels, satellite etc. So, hundreds to several thousand nodes are deployed throughout a sensor field to create a wireless multi-hop network.

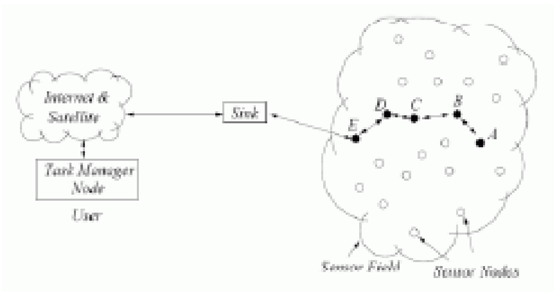

Figure 2: Wireless sensor network

3. Flooding Protocol

Flooding protocols the broadcast of messages is a frequently used operation to spread information to the whole network It is the simplest building block used by network algorithms and is often required by higher level protocols such as most routing algorithms. For this reason, it is important for the broadcast to be implemented in the most efficient way. Its performance is likely to affect the global efficiency of any protocol using it. Flooding is an old technique that can also be used for routing in sensor networks. In flooding, each node receiving a data or management packet repeats it by broadcasting, unless a maximum number of hops for the packet is reached or the destination of the packet is the node itself. Flooding is a reactive technique, and it does not require costly topology maintenance and complex route discovery algorithms

Flooding Example
A rather direct and simple way to implement broadcast is to flood the message over the network. When a node initiates a broadcast, it transmits the message to its neighborhood. By neighborhood, we mean all the nodes within the sender's transmission range. Then, when the message is received for the first time, the recipient re-broadcasts it. An example is shown in Figure 4. with a network composed of fie mobile nodes labeled from A to E.

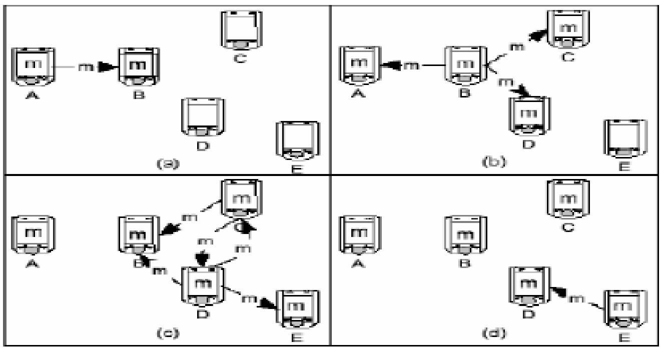

Figure 3: Example of flooding

Node A initiates a broadcast by flooding a message m to its surrounding nodes. In step (a), A floods m to its single neighbor B. Then, in step (b), node B, which has received m for the first time, rebroadcasts m to nodes C and D and so on.

In (c), m is completely flooded through the whole network and delivered to every node. In this example, at least three steps are required in order to reach node E on the right. The last step (d) is useless as every neighbor of E has already received m. This technique has an important drawback : it leads obviously to an overhead of flooded messages in the network. With ideal conditions (i.e. all node receive the broadcast) in a network of N nodes, a single broadcast will generate exactly N copies of itself which are likely to increase the probability of collisions. Moreover, most nodes will receive the same message several times keeping the shared medium unnecessarily busy[37].

4. Directed Diffusion Protocol

Direct Diffusion is the data centric protocol [21][23]. It consists of several elements: interests, data messages, gradients, and reinforcements. First, sink node requests data by sending interests. An interest message is a query or an interrogation, which specifies what a user wants to its neighbors for named data. The data is named using attribute-value pairs and it is the collected or processed information of a phenomenon that matches an interest of a user. The interests are flooded over the whole network by the sink. Such data can be an *event, which* is a short description of the sensed phenomenon. Whenever a node receives an interest, it will check whether the interest exists or new one. If it is a new interest, the sensor node will set up a gradient toward the sender to "draw" down data that matches the interest. Each pair of neighboring nodes will establish a gradient to each other. After the gradient establishment stage, the source node begins to send the related data that matches the interest to the sink. The data are generally broadcasted to all its gradient neighbors. Events are propagated toward the interest originators along multiple gradient paths. The sensor network reinforces one or a small number of these paths. The reinforcement scheme in directed diffusion is generally designed for minimum delay or maximum number of packets received during a certain period of time as shown in Figure 4.

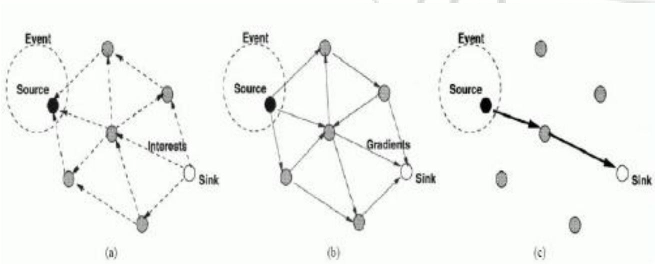

Figure 4: Directed Diffusion, a) Interest Propagation, b) Initial gradient setup, c) Data delivery.

In directed diffusion, tasks are described or named using attribute-value pairs [21][23]. For the attribute-value based naming scheme, each attribute is associated with a value range. The attribute value can be any subset of its range. A simplified description of the four wheeler tracking task might be:

type = four-wheeler// detect four wheeler‟s location
interval = 10 ms // send back events every 10 ms
duration = 10 minutes // for the next 10 minutes
rect = [-100, 100, 200, 400] // from sensor nodes within rectangle

There are other choices for an arrangement of attribute-value pairs and other naming schemes. For example, a sensor that detects a four wheeler might generate the following data:
type = four-wheeler // detect four wheeler‟s location
instance = car // instance of this type
location = [125, 220] // estimated location
intensity = 0.6 // signal amplitude measure
confidence = 0.85 // confidence in the estimate
timestamp = 01:20:40 // event generate time

The interest is usually injected into the network at the sink. For each active task, the sink periodically broadcasts an interest to all its neighbors. The interval attribute specifies an event data rate. Since the location of the sources is not precisely known, interests must necessarily be diffused over a broader section of the sensor network than that covered by the potential sources. As a result, if the sink had chosen a higher initial data rate, higher energy consumption might have resulted from the wider dissemination of sensor data. The desired higher data rate can be achieved by reinforcement.

The simplest alternative is to rebroadcast the interest to all neighbors, which is equivalent to flooding the interest. This alternative is reasonable in the absence of information about the sensor nodes that can satisfy the interest. Given that interests are flooded, all nodes establish gradients, as shown in Figure 6. Unlike the simplified description in Figure 4. every pair of neighboring nodes establishes a gradient toward each other, as a crucial consequence of local interactions. An interest does not contain information about a sink. Therefore, when a node receives an interest, it is impossible for the node to determine whether the interest is delivered back to the node because there is a loop, the interest is delivered using another path, or the identical interest is newly generated from another sink. Such bi-directional gradients can cause a node to receive one copy of low data rate events from each of its neighbors. However, this technique can enable fast recovery from failed paths or reinforcement of empirically better paths and does not incur persistent loops.

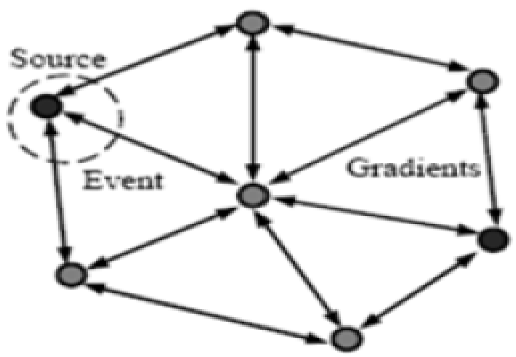

Figure 5: Gradient Establishment

5. Network Simulator (NS2)

Simulation can be defined as "Imitating or estimating how events might occur in a real situation". Network Simulator

(NS2) is simulation software to build network model and run on computer and analyze it.

NS [25][26] is an event driven network simulator developed at University of California at Berkeley, USA, as a REAL network simulator projects in 1989 and was developed at with cooperation of several organizations. Now, it is a VINT project supported by DARPA. NS is a discrete event network simulator where the timing of events is maintained by a scheduler and able to simulate various types of network such as LAN and WPAN according to the programming scripts written by the user. Besides that, it also implements variety of applications, protocols such as TCP and UDP, network elements such as signal strength, traffic models such as FTP and CBR, router queue management mechanisms such as Drop Tail and many more.

There are two languages used in NS2 C++ and OTcl (an object oriented extension of Tcl). The compiled C++ programming hierarchy makes the simulation efficient and execution times faster. The OTcl script which written by the users the network models with their own specific topology, protocols and all requirements need. The form of output produce by the simulator also can be set using OTcl. The OTcl script is written which creating an event scheduler objects and network component object with network setup helping modules. The simulation results produce after running the scripts can be use either for simulation analysis or as an input to graphical software called Network Animation (NAM). NS2 is an event driven network simulator, which can be implemented in Linux-based platform. The NS2 files (recommended to download a piece of file which includes all the needed files called ns-allinone-2.xx from http://www.isi.edu/nsnam/ns/[27]) must be downloaded into any media storage, most preferred is inside the computer itself where the NS2 is going to be installed. Since, we are using NS 2.33. It is not recommend logging in as a root because installation at root may interfere with any important Linux files.

6. Simulation

Simulation of routing protocols namely: Flooding and Directed Diffusion has been carried in NS2 to evaluate the performance. The analysis is being done on the basis of simulation results of *.nam file and the *.tr file. We have also evaluated the performance of the protocol. In the NS2-allinone package NAM is a built-in program. NAM helps us to see the flow of message between the nodes and works like an animator for the working of the program. It also shows the packets dropping or reaching to the destination properly with time reflection. When the TCL file is written, NAM is invoked inside that file. With the help of 2D graphs we have tried to analyze the simulation with different simulation time. The scripts for the NAM is stored as *.nam and for trace graph *.tr is used. The simulation has been mainly divided in three parts:

1) Simulation of flooding protocol
2) Simulation of Directed Diffusion protocol
3) The comparison between Flooding, Directed Diffusion and is performed on the common factors like, Average Delay and Energy Consumption in the different network size over different simulation time.

Simulation Model: Our routing protocol is implemented in the NS2 network simulator. In all our simulations, we consider a square sensor field, of size L. Inside the field, M static sensor nodes are deployed randomly. The value of M is varied from 15 to 25. Each node has a fixed radio range of 40 meters. The positions of the source and sink nodes are located far from each other. Such settings facilitate our evaluation of the protocol where the routing path has to traverse a large area in the sensor field. The data packet size is 64 bytes. We use an event-driven wireless sensor network in our experiment. After the route search phase, each source node generates data packets and sends them to the sinks through the network with a fixed rate. We adopt the NS2 radio energy model and assign each node with the same initial energy level of 10 Joule at the beginning of each simulation in order to keep the simulation time within a reasonable time period. We further assume that each sensor node carries an Omni antenna and the energy consumptions for idle time, transmission and reception are 35 mW, 660 mW, and 395 mW respectively. The energy dissipation for data processing in the node is neglected in our simulations. We adopt the IEEE 802.15 MAC layer provided in the NS-2 with a bandwidth of 1.6 Mbps. Table 1. below shows the Network parameter during simulation.

Table 1: Network parameter in simulation

Parameters	**Value**
Number of Nodes	15
Data Packet Size	64bytes
Idle Power	35mW
Receive Power	395mW
Transmit Power	660mW
Node Initial energy	10joule
Node Radio range	40m
MAC Protocol	IEEE 802.15

We use a number of metrics to evaluate the performance of our protocol and compare with other existing schemes. The average delay measures the average time spent to relay data packets from the source node to the sink node. It is calculated as follows:

Average delay =
$\Sigma(Timepacketarrive@dest - Time\ packetsent@source)/Total\ Number\ of\ Connection\ Pairs$

Eq. 4.1

Where Timepacketarrive@dest is the time at which packet arrives at destination and Timepacketsent@source is the time at which source sends the packet arrives at destination and Timepacketsent@source is the time at which source sends the packet.

The node energy consumption measures the average energy dissipated by the node in order to transmit a data packet from the source to the sink. It is calculated as follows:
Node energy consumption =
$\Sigma(e_{i,init}\ _{Mi=1} - e_{i,res})/M\Sigma dataN_{jSj=1}$

Eq. 4.2

Where M is the number of nodes in the network, ei,init and ei,res are respectively the initial and residual energy levels of node i, S is the number of sink nodes and dataNj is the number of data packets received by sink j. The average node energy measures the average energy level of all nodes in the network after the data transmission has been started for a certain amount of time. It gives an indication of the network state in terms of energy consumption.

7. Result Performance Evolution & Analysis

Average delay:
Average packet delay measures the average one way latency observed between transmitting an event and receiving it at sink. Ideally Average Delay should have a rather constant value .Table 2 shows the average packet delay for different routing protocols under different topology settings. The Directed Diffusion has the shortest delay compared to Flooding. In Flooding the end to end delay is increasing continuously because all the nodes on the straight path have lost all the energy. So to reach the destination a comparatively long path is being followed by the message which implies more end to end delay. The delay at network size 15 for flooding is .21sec. and the delay at network size 15 for Directed Diffusion is 0.021 This shows that the delay of directed diffusion is shorter than flooding.

Table 2

No. of Nodes	Flooding	Directed Diffusion
15	0.213836	0.021400

Node Energy Consumption:
The next metric that we study is the node energy consumption. Table. 3 show the simulation results of different topology settings. At network size of 15, node energy consumption is .019% for flooding and 0.003% for directed diffusion. Thus we can observe that there is lower node energy consumption in case of Directed Diffusion routing protocol over the flooding protocols. The flooding is the most costly protocol; by adding a simple mechanism of retransmission probability control on top of flooding, the directed transmission improves the energy efficiency. The Directed Diffusion routing obtains further improvement in energy consumption by effectively reducing redundant information spread and saves the energy. We can observe that Directed Diffusion routing protocol can still maintain its node energy consumption at a low level even when the network size increases.

Table 3

No. of Nodes	Flooding	Directed Diffusion
15	0.019968	0.003793

8. Conclusion

Routing is a significant issue in Wireless Sensor Networks. In this project we have done an intensive and detailed study on the routing protocol in wireless sensor networks. An attempt has been made to implement Directed Diffusion routing protocols using NS-2(Network Simulator-2). Also a detailed comparison of the performances of these protocols with Flooding routing protocol has been carried out. We have chosen two metrics to analyze the performances and compare these protocols. The results of comparison have been presented in the form of tables. From the results, we concluded that the Directed Diffusion routing protocol has the shortest delay compared to Flooding Protocol. In Directed Diffusion protocol the end to end delay is not increasing continuously. Directed Diffusion is data centric so there is no need for a node addressing mechanism. Directed diffusion can reduce the bandwidth needed for sensor networks. Each node is assumed to do caching and sensing. Directed diffusion is energy efficient since it is on demand and there is no need to maintain global network topology. We also observed that Directed Diffusion routing effectively reduces redundant information spread and saves the energy and thus has lower node energy consumption over Flooding Protocol. These results indicate that the Directed Diffusion algorithm is more efficiency in energy than the Flooding Protocol and perform much better than the traditional Flooding scheme

9. Future Scope

In this thesis, we compared Directed diffusion routing protocol and Flooding protocol for wireless sensor networks. Further research work is required to enhance the performance of Directed diffusion protocol. The Directed Diffusion routing protocol as proposed applies for static sensor nodes and sleeping nodes also. It will be useful to enhance the protocol to support nodes with limited mobility, as they are able to better adapt to the environment. A location update mechanism is required to allow each node to be aware of its own and its neighbors‟ positions constantly. It is a challenge to balance between the node energy consumption and the additional maintenance efforts that keeps the node coordinates
Up-dated.

References

[1] I. Howitt and J.A. Gutierrez, “IEEE 802.15.4 low rate-wireless personal area network coexistence issues”, Wireless Communications and Networking 2003 pp 1481-1486

[2] Avokh, A. Mirjalily, G., “Dynamic Balanced Spanning Tree (DBST) for data aggregation in wireless Sensor Networks”, 5th International Symposium on Telecommunications (IST, 2010), Tehron, Iran‟ Dec 2010, pp 391-396

[3] YiPing Chen, Yu-Zhong Chen, “A Novel Energy-Efficient Routing Algorithm for Wireless Sensor Networks” International conference on Machine Learning and Cybernetics (ICMLC), 2010 pp 1031-1036

[4] Sha chao, Wang ru-chan, “Energy efficient clustering algorithm for data aggregation in wireless sensor networks”, The Journal of china Universities of posts and Telecommunications „„ volume 17, December 2010, pp 104-109

[5] Md. Zeynali, Leile Md khanli, Amir Mollanejad, “EDARP: Novel Energy and Distance-aware Rotting protocol in Wireless Sensor Network”, Proceeding of the 2nd International Conference on International Science: Informational Technology, culture and human, 2009

[6] Jun Yang, Deyon Zhang, Yunyi Zhang, “An Energy-Efficient Data Gathering Protocol for wireless Sensor Networks”, Eighth IEEE/ACIS International

conference on computer and information science, Shanghai, June- 2009

[7] S. Tilak et al., "A Taxonomy of Wireless Microsensor Network Models," *in ACM Mobile Computing and Communications Review (MC2R)*, June 2002.

[8] Sha chao, Wang ru-chan, "Energy efficient clustering algorithm for data aggregation in wireless sensor networks", The Journal of china Universities of posts and Telecommunications ,,, volume 17, December 2010, pp 104-109

[9] J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey*," IEEE Wireless Communications*, vol. 11, issue 6, pp. 6-28, Dec. 2004.

[10] M. Younis, M. Youssef and K. Arisha, "Energy-Aware Routing in Cluster-Based Sensor Networks*", in the Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS2002)*, Fort Worth, TX, October 2002.

[11] D. Ganesan, R. Govindan, S. Shenker, and D.Estrin, "Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks," *in Proc. of the Second ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mo-biHoc 2001*), Long Beach, CA, Oct. 2001

[12] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks," *in the Proceedings of the First Workshop on Sensor Networks and Applications (WSNA)*, Atlanta, GA, October 2002.

[13] Sangho Yi, Junyoung Heo "PEACH: power efficient & Adaptive clustering hierarchy protocol for wireless sensor network". Publisher Butterworth-Heinemann USA 2007

[14] S.S. Satapathy and N. Sarma," TREEPSI: tree based energy efficient protocol for sensor information," Wireless and Optical Communications Networks 2006 IFIP International Conference 2006.

[15] Lindsey and Cauligi S. Raghavendra, "PEGASIS: Power- Efficient Gathering in Sensor Information Systems", Aerospace conference proceedings, IEEE, 2002, pp 1-10.

[16] C. Schurgers and M.B. Srivastava, "Energy efficient routing in wireless sensor networks," *in the MILCOM Proceedings on Communications for Network-Centric Operations: Creating the Information Force*, McLean, VA, 2001.

[17] K. Akkaya and M. Younis, "An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks*," in the Proceedings of the IEEE Workshop on Mobile and Wireless Networks (MWN 2003*), Providence, Rhode Island, May 2003.

[18] Y. Yao and J. Gehrke, "The cougar approach to in-network query processing in sensor networks," *in SIGMOD Record*, September 2002.

[19] K. Sohrabi, et al., "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, Vol. 7, No. 5, pp. 16-27, October 2000.

[20] Shuaib, A.H.; Mahmoodi, T.; Aghvami, A.H.; A timed Petri Net model for the IEEE 802.15.4 CSMA-CA process, Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium , Page(s): 1204 – 1210,2009

[21] L. Zhao, G. Liu, J Chen, Z. Zhang"Flooding and Directed Diffusion Routing Algorithm in Wireless Sensor Networks*",Ninth International Conference on Hybrid Intelligent Systems*,2009.

[22] J. Ding, K. Sivalingam, R. Kashyapa, L. J. Chuan, "A multi-layered architecture and protocols for large-scale wireless sensor networks", *IEEE 58th Vehicular Technology Conference, 2003, VTC 2003-Fall 2003*, Volume: 3, Pages:1443 - 1447, 6-9 Oct. 2003.

[23] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," *Proceedings of ACM MobiCom '00*, Boston, MA, pp. 56-67, 2000.

[24] B. Wei, H.Y. Hu and W. Fu, "An Improved LEACH Protocol for Data Gathering and Aggregation in Wireless Sensor Networks," *Proceedings of the International Conference on Computer and Electrical Engineering*, pp.398-40 I, December 2008.

[25] Y. Bin, S. Bin, W. Nuan, NS2 And Network Simulation .Beijing: Posts & Telecom Press, 2007.

[26] L. L. Y. Ping, Routing Protocols Simulation Based on NS2, *Ship Electronic Engineering [J]*,.4 Vo.l 28 No. 4:132-134,2008

[27] C.C. Shen, C. Srisathapornphat, C. Jaikaeo:" Sensor Information Networking Architecture and Applications", *IEEE Personal Communications*, pp.52-59 (Aug. 2001).

Author Profile

Deepesh Dewangan received the MCA degree in Computer Application from Shankaracharya Bhilai(CG) in 2008 and M.Tech degree in Computer Science & Engineering from Rungta College of Engineering & Technology in 2013, respectively. During 2009-2011, he was teaching as Assistant Professor in Maharaja Agrasen International College, Raipur(CG), and During 2012-1014 he was teaching as Assistant Professor in Kruti Institute of Technology & Engineering, Raipur(CG). Now, He is Working with Kalinga University, Naya Raipur as an Assistant Professsor in CSE Department.

Srikant Singh received the MCA degree in Computer Application from Indira Gandhi National Open University , New Delhi in 2010 and have an specialization in J2SE. During 2010-2012 he was working with Future Point a SMU Study Centre as Lead Faculty, he During 2012-13, he was working with UMC, Raigarh(CG) as Assistant Professor of Computer Science and During 2013-14, he was working with DSCET, Durg(CG) as Assistant Professor of Computer Science. Now, He is Working with Kalinga University, Naya Raipur as an Assistant Professsor in CSE Department.