Simulation of the TEEN and the SPIN Protocols in a Wireless Sensor Network Environment

Mohammed Omari, Noura Tiouririne, and Djamila Dahmani

***Abstract*— In WSN, sensor nodes have a limited transmission range. In fact, their processing and storage capabilities as well as their energy resources are also limited. However, regular routing protocols for wireless devices are not well suitable for sensors due to the constraints of their resources. These protocols ensure the duty of maintaining network routing and reliable single/multi-hop communication which requires considerable energy consumption. In this paper, we present our java-based simulators that implement two state-of-the-art wireless sensor network protocols: TEEN and SPIN. Our goal is to compare them vis-à-vis the energy consumption and the average throughput at each node. Our experimentations show a better performance with SPIN in terms of energy consumption protocol. On the other hand, the TEEN protocol shows a slightly higher throughput performance due to the absence of advertising messages as in most reactive routing protocols.**

***Index Terms*—Wireless sensor networks, cluster head, energy efficient protocol, SPIN, TEEN.**

I. Introduction

A wireless sensor network (WSN) is a network of simple sensing devices that are capable of capturing some parameter variations such as heat or weather, and able to transfer data to other devices. In fact, WSNs are widely investigated and considered as one of the most important technologies for the twenty first century [1]. With the explosion of WSNs there is an increasing demand for developing new protocols in order to efficiently convert physical phenomena into data which was difficult or sometimes impossible when conventional ways are used. As a matter of fact, the characteristic features of WSN are different from other ad hoc networks in terms of energy, processing speed, and physical storage.

In a typical environment, users can retrieve information of interest from a wireless sensor network by injecting queries and gathering results from base stations which behave as an interface between users and the network [2]. WSNs combine microsensor technology, low-power signal processing, computation, and low-cost wireless networking in a small system that provides distributed network and internet access to sensors and controls sensors that are deeply embedded in the environment [3]. Sensor nodes are dispersed over the area of interest and possess the capability of radio frequency communication, signal process, and communication protocols process. It allows hundreds to thousands of nodes to cooperate in the network to achieve a larger task.

A wireless sensor network is different from the conventional wireless networks. The small size of the sensor node limits battery capacity requiring every operation to be done efficiently. It also limits the radio transition range and suggests a small multi-hop transmission structure. Using several short range intermediate nodes to send a bit is much more energy-efficient than using one-long hop [3]. WSNs are influenced by different factors, such as fault tolerance, scalability, production costs, hardware constraints, network topology, physical environment, and power consumption. The network is usually organized using either a flat or hierarchical structure. One key challenge is how to handle network dynamics during the process of network discovery and organization. These dynamics include fluctuation in channel quality, failure of sensor nodes, variations in sensor node capabilities, and mobility or diffusion of the monitored entity. Autonomous adaptation of network discovery and organization protocols in light of such dynamics is the key to deliver proper system functionality [2].

The design of WSNs usually encounters more challenges than the traditional communication networks. For instance, maximizing the reliability may increase the network energy consumption substantially [4]. Hence, the network designers need to consider the tradeoff between reliability and energy consumption. In terms of transmission delay, sensor information must reach the sink within some deadline. Time delay is a very important QoS measurement since it influences on performance and stability of an industrial control system [5][6]. Note that controllers can usually tolerate a certain degree of packet losses and delay [5][7]. Hence, the maximization of the reliability and minimization of the delay are not the optimal design strategies since these strategies will significantly decrease the network lifetime for the control applications [8]. The type and amount of data to be transmitted is also an important issue in designing sensor networks [5]. In fact some environmental monitoring application protocols for WSNs operate in low traffic networks and cannot handle higher traffic loads [9][10]. Moreover, when the size of sensors increases, the calculations necessary to implement a protocol must be computationally light. These operations should be performed locally in order to avoid the heavy transmissions with a the base station [11][12].

In this paper we try to investigate two different protocols

M. Omari, associate professor, is with LDDI laboratory, at the University of Adrar, Adrar 01000, Algeria (phone: 213-49-967571; fax: 213-49-967572; e-mail: omari@univ-adrar.org).

N. Tiouririne, master student, is with the University of Adrar, Adrar 01000, Algeria (e-mail: tiouririne88@gmail.com).

D. Dahmani, master student, is with the University of Adrar, Adrar 01000, Algeria (e-mail: djammoula@gmail.com).

that are used in the WSN domain. Our goal is to implement these protocols in a simulation environment and compare both of them in terms of throughput and energy consumption. Hence, the rest of this paper is organized as follows. In Section 2 and Section 3, we present a brief description the two selected WSN protocols (TEEN and SPIN) with a brief description of each protocol. In Section 4, we layout the steps of building our java-based simulators that implement both protocols, in addition to the set of varying parameters that conduct the simulations, as well as an interpretation of the experimental results. Section 5 is the conclusion.

II. Threshold sensitive Energy Efficient sensor Network protocol (TEEN)

WSNs are divided into two categories: proactive networks and reactive networks. In proactive networks, the nodes periodically switch on their sensors and transmitters, sense the environment and transmit the data of interest, hence providing a snapshot of the relevant parameters at regular intervals. These networks are well suited for applications requiring periodic data monitoring [13]. On the contrary, nodes in reactive networks react immediately to sudden and drastic changes in the value of a sensed attribute. These networks are well suited for time critical applications [17].

The TEEN protocol is applied in reactive networks. In terms of energy efficiency, TEEN protocol has been observed to outperform existing conventional sensor network protocols. The traditional routing protocols defined for wireless ad hoc networks [14][15] are not well suited due to the following reasons:

1- Sensor networks are “data centric” it means, unlike traditional networks where data is requested from a specific node, data is requested based on certain attributes.

2- The requirements of the network change with the application and so, it is application-specific [16].

3- Adjacent nodes may have similar data. So, rather than sending data separately from each node to the requesting node, it is desirable to aggregate similar data and send it.

4- In traditional wired and wireless networks, each node is given a unique id, used for routing. This cannot be effectively used in sensor networks simply because it implies large ids [17], which might be substantially larger than the actual data being transmitted.

A. *Clustering*

The TEEN protocol is applied with certain hierarchy in the network [18]. It consists of a base station (BS), away from the nodes, through which the end user can access data from the sensor network. All the nodes in the network are homogeneous and begin with the same initial energy. The BS can transmit with high power to all the nodes because it has a constant power supply and has no energy constraints. So, there is no need for routing from the base station to any specific node. Therefore, the nodes cannot always reply to the BS directly due to their power constraints, resulting in asymmetric communication. This model uses a hierarchical clustering scheme as shown in Fig. 3.

B. *Energy Saving:*

Each cluster has a cluster head which collects data from its cluster members, and sends the aggregated information to the BS or an upper level cluster head [13]. This pattern is repeated to form a hierarchy of clusters with the uppermost level cluster nodes reporting directly to the base station. The base station forms the root of this hierarchy and supervises the entire network. This hierarchy imposes that all the nodes transmit only to their immediate cluster head, i.e., only the cluster head needs to perform additional computations, which makes the protocol very efficient in terms of energy consumption.

III. Sensor Protocol for Information via Negotiation (SPIN)

SPIN is a proactive protocol that was designed to enable data-centric information dissemination in sensor networks [17]. Rather than blindly broadcasting sensor data throughout the network, nodes receiving or generating data first advertise this data through short ADV messages (advertise messages) as shown in Fig. 4 (a). The ADV messages simply consist of an application-specific meta-data description of the data itself. This meta-data can describe such aspects as the type of data and the location of its origin. Nodes that are interested in this data request the data from the ADV sender through REQ messages. Finally, the data is disseminated to the interested nodes through DATA messages that contain the data as shown in Fig. 4 (b).

The advantage of SPIN over blind flooding or gossiping data dissemination methods is that it avoids three costly problems: implosion, overlap and resource blindness. Implosion occurs in highly connected networks that employ flooding and thus each sensor receives many redundant copies of data. For large data messages, this wastes considerable energy. Short ADV messages helps in reducing the costly and unnecessary transfer of data messages due to the redundant nature of sensor transmission. Hence, two sensors with some common data will both send their data, causing redundancy in data transmission and thus SPIN is able to solve this problem by naming data so that sensors only request the data or parts of data they are interested in receiving. So with SPIN, there are mechanisms whereby a sensor that is running low on energy will not advertise its data in order to save its limited energy resources. Thus SPIN solves the resource blindness problem by having sensors make decisions based on the current level of available resources [17].

IV. Simulations, Results, and Analysis

In this section, we present the different simulation parameters as well as the experimental results that conduct us to evaluate the performance of TEEN and SPIN protocols. As a programming language, we used the Java Development Kit (JDK) in order to build our simulators. Java is a famous object oriented programming language that acquires a large library for building graphical interfaces as well as thread manipulation. Our goal is to compare the performance of TEEN and SPIN protocols based on the average energy

consumption and throughput versus the size of the network.

In fact, we altered a platform simulator that was originally designed by Soliman and Omari (SDES simulator) [19] to simulate security protocols. In the conducted simulation, the experiments have been performed on 256 sensors nodes, two cluster heads, and two bases stations. In both protocols, all the sensor nodes start with the same amount of energy.

A. Simulation of an experimental environment using threads:

Every sensor node is entitled to send data to the upper level (cluster or principal node) or to the base. Clusters of sensor nodes are formed independently. In order to simulate a real WSN, we implemented sensor nodes using concurrent programming where each sensor node is represented by a thread that simulates node's dynamics in terms of sending, receiving and aggregating data. The `run()` function of each sensor thread describes the life of a sensor:

```
/* create sensor nodes thread */
nodeThreads = new
NodeThread[simulationParameters.getNumberOfNodes()];
for(int i = 0; i <
  simulationParameters.getNumberOfNodes(); i++) {
    nodeThreads[i] = new
    NodeThread(simulationParameters, nodeInfo[i],
      baseStationInfo, nodesTable);
    }

/* starting sensor nodes threads */
for(int i = 0; i <
  simulationParameters.getNumberOfNodes(); i++) {
      nodeThreads[i].start();
}

/* start running a sensor node's thread */
public class Sensor Nodes Thread extends Thread {
    public void run() {
      /* associate with the nearest base station
      and register */
      associateWithNearestBaseStation();

      try {
          /* register to the base station */
          register();
        } catch(Exception e) {
        }

      while (true) {
        try {
          /* generate messages */
          generateMessages();
        } catch(Exception e) {
        }
      }
    }
}
```

Similarly, The `run()` function of each base station thread describes the life of a base station:

```
/* create base station threads */
baseStationThreads = new
BaseStationThread[simulationParameters.getNumberOf
  BaseStations()];
for(int i = 0; i <
  simulationParameters.getNumberOfBaseStations();
    i++) {
  baseStationThreads[i] = new
  BaseStationThread(simulationParameters,
  baseStationInfo[i], baseStationsTable);
}

/* start base station threads */
for(int i = 0; i <
  simulationParameters.getNumberOfBaseStations();
    i++)
  baseStationThreads[i].start();

/* start running a base station's thread  */
public class BaseStationThread extends Thread {
  public void run() {
    while (true) {
      byte dataToSend[] = new byte[Constant.P_SIZE];
      byte receivedData[] = new
                        byte[Constant.P_SIZE];

      /* wait for packets */
      receivedData = waitForPackets();

      /* analyzing different head packets */
      analyzePacket(receivedData);

      /* process packet */
      try {
        dataToSend = processPacket(receivedData);
      } catch (Exception e) {
      }
      /* send packet back */
      replyPacket(dataToSend);
    }
  }
}
```

B. Simulation interface:

Fig. 1 shows the simulator interface of the TEEN protocol (SPIN also has a similar interface). The input parameters used in our simulations are set as shown in Table II.

Once the simulator is launched, two tables appear: one for the cluster heads and one for the sensor nodes (Fig. 2). These tables show a real time description of network communication in terms of sent and received messages.

The second interface (Fig. 2) presents the current sent and received data, and amount of consumed energy. Once the simulation is completed (or stopped by the user), the simulator automatically generates a trace file that includes the cumulative information related to sent and received data, and consumed energy.

C. Hardware specification

Our simulation of the TEEN and the SPIN protocols were conducted using the hardware specification as shown in Table I:

TABLE I
HARDWARE SPECIFICATIONS OF THE SIMULATION MACHINE

Hardware component	Characteristic
Memory RAM	2 GB
Microprocessor	Intel Pentium Dual CPU T3400 @2.16 GHz 2.17 GHz

D. Results and Analysis:

In this section, we will lay out the steps to compare the two selected protocols (TEEN and SPIN). For each simulation instance we calculate the amount of received data of all nodes, and then divide it by the number of nodes in order to get the proportional data per each node. These experiments were run several times to gain high simulation confidence. The experiments are also run with deferent probabilistic distributions of packet generation (uniform, exponential, gamma, logarithmic and pareto) in order to get more solid and accurate results. The simulation time was set to several minutes for all our conducted experiments. Table II shows the different parameters used in the simulation along with their range values:

TABLE II
INPUT PARAMETERS FOR CONDUCTED SIMULATIONS OF TEEN AND SPIN PROTOCOLS

Parameter	Value	Observation
Number of Base Stations	2	
Number of Cluster Heads	2	
Number of Sensor Nodes	10 to 256	
Zone Height	100	
Zone Width	100	
Initial Energy	1000 (Unit)	
Probabilistic Distribution of Packet Generation	Variable	Uniform, Gamma, Exponential, pareto, and logarithmic

Figures 5, 7 and 9 show a better performance in energy consumption in SPIN compared to TEEN protocol. We can clearly see that the energy consumption increases in both protocols when the number of sensor nodes increases; however, SPIN shows a lower magnitude of consumption. In fact, the reduction of transmission volume (from data to advertising only) in SPIN made it save a lot of unnecessary transmissions, especially when the cluster head or the base station detects redundancy between adjacent senor nodes.

Figures 6, 8, and 10 show that both SPIN and TEEN protocols maintain the same sensor throughput in small networks. However, when the number of integrated sensors increases, the average throughput per sensor node starts decreasing (uniform distribution) though the TEEN protocol slightly maintained higher throughput. This is due to the extra messages of advertising in SPIN that decreases somehow the throughput performance, yet it is still insignificant.

V. CONCLUSION

WSNs became very popular in the recent years due to advances interest in sensing, communication, and computation. In order to make wireless sensor networks practically useful, efficient network protocols must be developed with a main focus on low power consumption, scalability, and low latency.

There are several protocols in the literature that were proposed for WSNs communication. In this paper, we selected two famous protocols (TEEN and SPIN) in order to evaluate the performance of proactive and reactive WSNs. Moreover, we have highlighted the advantages of each protocol which raised the need for simulation and comparison in terms of consumed energy and average throughput. Therefore, we built our own simulators using the friendly-user java platform. Extensive simulations show that the SPIN protocol saves a lot of energy due to its advertising that precedes sending data. On the other hand, advertising packets made the SPIN protocol less performing in terms of throughput; yet, the throughput difference was insignificant. Hence, our experimentations generally show a clear advantage of SPIN over TEEN when both energy consumption and throughput are considered.

REFERENCES

[1] "21 Ideas for the 21st Century," Business Week, pp. 78– 167, Aug. 30, 1999.

[2] Y. Yu, V. K Prasanna, and B. Krishnamachari. "Introduction to Wireless Sensor Networks: Information Processing and Routing in Wireless Sensor Networks, Chapter 1," *World Scientific Publishing Co. Ptc. Ltd*, pp. 1-21, 2006.

[3] J. Rabaey, M. Josie Ammer, J. L. Da Silva, D. Patel and S. Roundy, "PicoRadio Supports Ad Hoc Ultra-Low-Power Wireless Networking," *Computer Magazine*, 2000.

[4] A. Willig, "Recent and emerging topics in wireless industrial communication," *IEEE Transactions on Industrial Informatics*, vol. 4, no. 2, pp. 102–124, 2008;

[5] J. R. Moyne and D. M. Tilbury, "The emergence of industrial control networks for manufacturing control, diagnostics, and safety data," *Proceedings of the IEEE*, 2007.

[6] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, 2007.

[7] W. Zhang and M. S. Braniky and S. M. Phillips, "Stability of Networked Control Systems," *IEEE Control Systems Magazine*, 2001.

[8] A. Speranzon, C. Fischione, K. H. Johansson, and A. Sangiovanni-Vincentelli, "A distributed minimum variance estimator for sensor networks," IEEE Journal on Selected Areas in Communications, *Special Issue on Control and Communication*, vol. 26, pp. 609–621, May 2008.

[9] N. Burri, P. von Rickenbach, and R. Wattenhofer, "Dozer: ultra-low power data gathering in sensor networks," *in ACM/IEEE IPSN*, 2007.

[10] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh., "Fidelity and yield in a volcano monitoring sensor network," *in USENIX OSDI*, 2006.

[11] A. G. A. Elrahim, H. A. Elsayed, and S. H. Elramly. "A new Routing Protocol for Mobility in Wireless Sensor Networks," *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT)*, February 2011.

[12] Y. SABRI, N. EL KAMOUN. "A Distributed Method for Localization in Large-Scale Sensor Networks based on Graham's scan," *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT)*, January 2012.

[13] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva. "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *In Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, ACM, October 1998.

[14] E. M. Royer and C.-K. Toh. "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks". *In IEEE Personal Communications Magazine*, pages 46–55, April 1999.

[15] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. "Next Century Challenges: Scalable Coordination inWireless Networks," *In Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, pages 263–270, 1999.

[16] J. Elson and D. Estrin. "An Address-Free Architecture for Dynamic Sensor Networks," *Technical Report 00-724*, Computer Science Department, USC, January 2000.

[17] Kemal Akkaya, Mohamed Younis. "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, May 2005, Vol. 3, 3, pp. 325-349.

[18] W. Heinzelman, J. Kulik, and H. Balakrishnan. "Adaptive protocols for information dissemination in wireless sensor networks," *In Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom),* 1999.

[19] H. S. Soliman and M. Omari. "An Efficient Application of a Dynamic Crypto System in Mobile Wireless Security," *in the proceedings of IEEE Wireless Communications and Networking Conference*, Atlanta, Georgia, March 2004.

Mohammed Omari received his B.E degree from the University of Es-Senia Oran (Algeria) in 1995 and the M.S. degree from New Mexico Tech (New Mexico, USA) in 2002, as well as the Ph.D. degree in 2005. His major is in computer science. He is currently an Associate Professor at the computer science department, and a unit president at the LDDI laboratory (University of Adrar, Algeria). His research interests include network security, cryptography, sensors and ad hoc protocols, image processing, neural networks, and genetic algorithms.

Noura Tiouririne received her B.S degree in computer science from the University of Adrar, Algeria, in 2011, and she is currently a master student at the same university. Her research interests are in ad hoc networks and their performance analysis.

Djamila Dahmani received her B.S degree in computer science from the University of Adrar, Algeria, in 2011, and she is currently a master student at the same university. Her research interests are in ad hoc networks and their performance analysis.

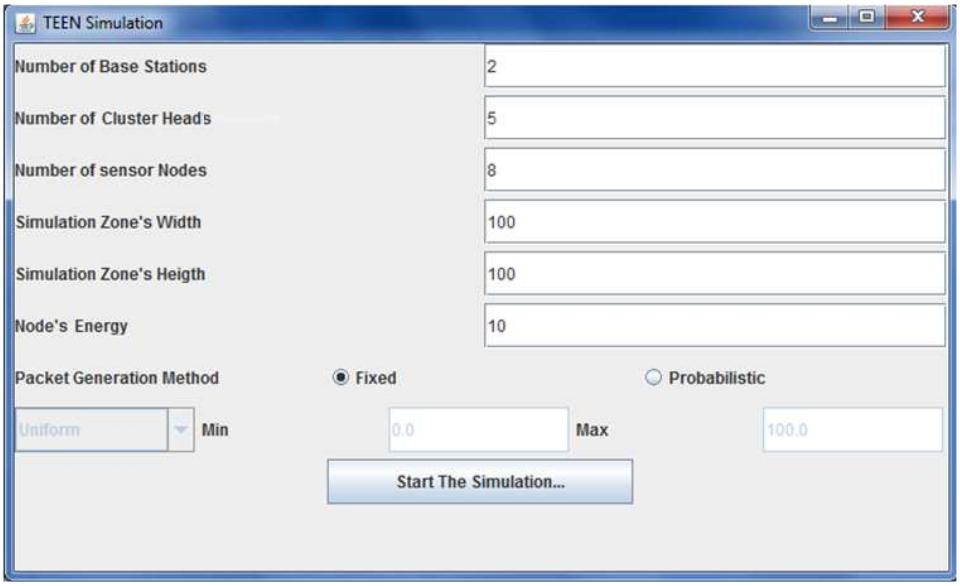

Fig. 1. Simulator interface of the TEEN protocol.

TEEN Simulator

Cluster Nodes Table:

Base ID	X	Y	Registered No...	Associated No...	Registered Me...	Associated Me...	Data Messages	Nodes To Nod...	Total Messages
0	25	25	0	0	138	138	137	256	669
1	75	25	0	0	118	118	117	0	353

Sensor Nodes Table:

Node ID	Nodes Cluster ...	X	Y	Registration Me...	Association Mes...	Energy	Data Messages	Total Messages
0	0	47	92	1	1	7	1	3
1	0	25	38	1	1	7	1	3
2	0	6	72	1	1	7	1	3
3	0	35	92	1	1	7	1	3
4	1	57	87	1	1	7	1	3
5	0	28	80	1	1	7	1	3
6	1	85	60	1	1	7	1	3
7	0	2	73	1	1	7	1	3
8	0	6	79	1	1	7	1	3
9	0	46	3	1	1	7	1	3
10	1	77	65	1	1	7	1	3
11	0	28	93	1	1	7	1	3
12	1	56	7	1	1	7	1	3
13	0	0	84	1	1	7	1	3
14	1	60	3	1	1	7	1	3
15	0	49	47	1	1	7	1	3
16	0	50	61	1	1	7	1	3
17	0	16	92	1	1	7	1	3
18	0	46	59	1	1	7	1	3
19	1	98	99	1	1	7	1	3
20	1	98	23	1	1	7	1	3
21	1	76	38	1	1	7	1	3
22	1	65	75	1	1	7	1	3
23	0	35	26	1	1	7	1	3
24	0	38	18	1	1	7	1	3

Stop The Simulation

Fig. 2. Running of the TEEN simulator.

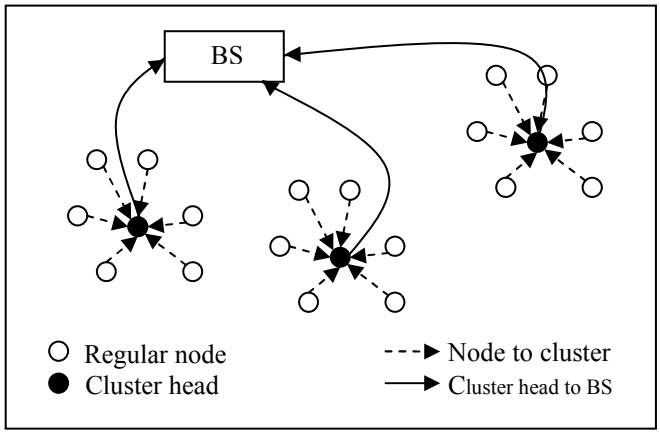

Fig. 3. Cluster model of the TEEN protocol.

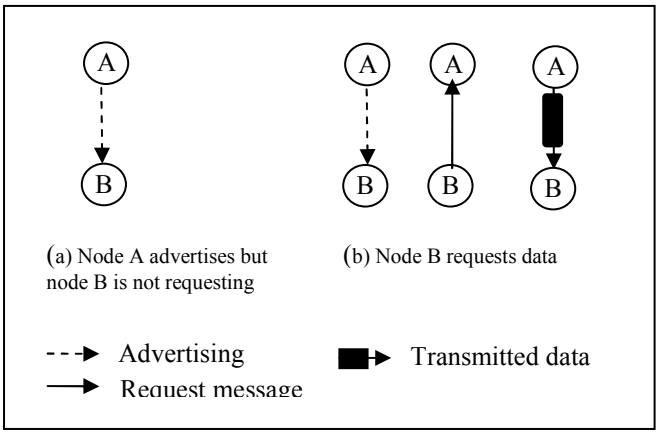

Fig. 4. Advertising and requesting property of the SPIN protocol.

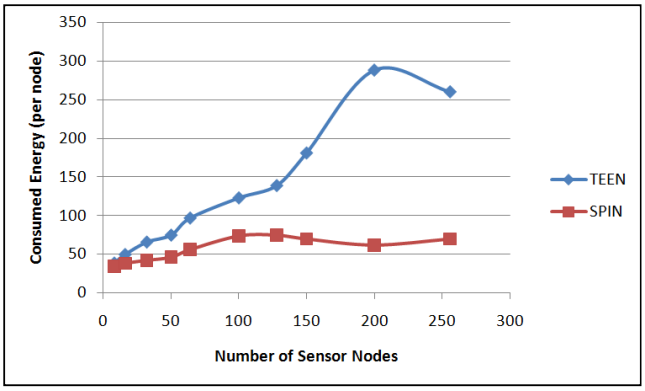

Fig. 5. Consumed energy comparison using exponential distribution.

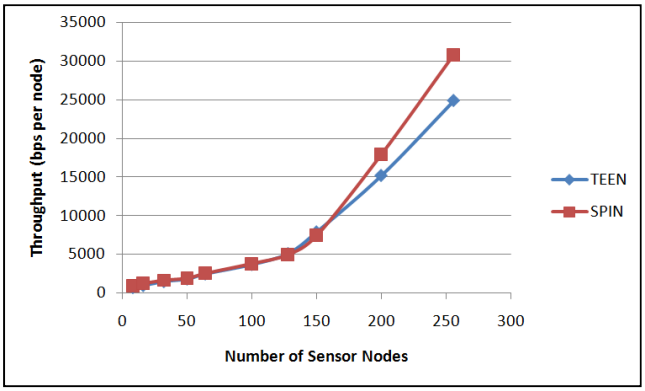

Fig. 6. Throughput comparison using exponential distribution.

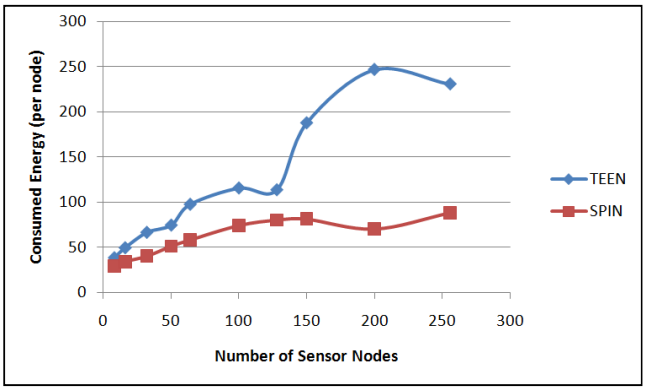

Fig. 7. Consumed energy comparison using uniform distribution.

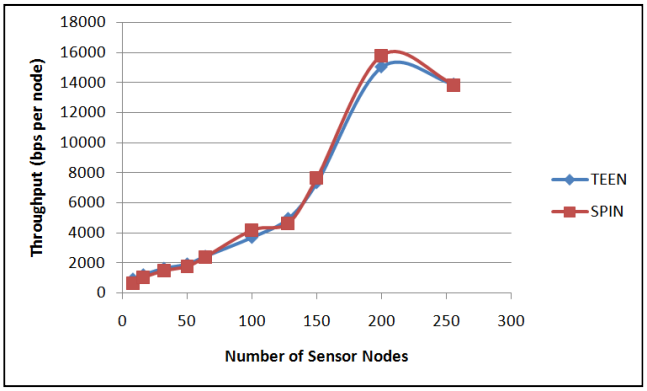

Fig. 8. Throughput comparison using uniform distribution.

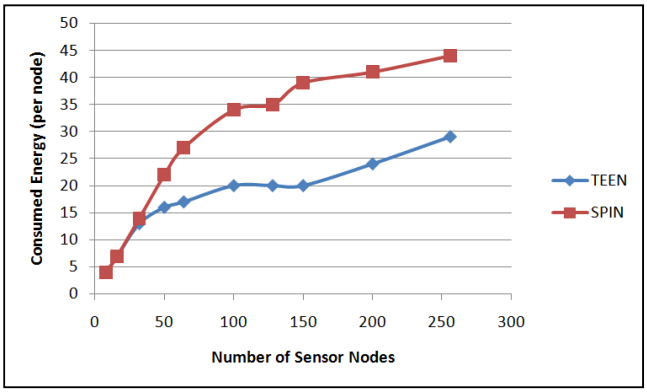

Fig. 9. Consumed energy comparison using logarithmic distribution.

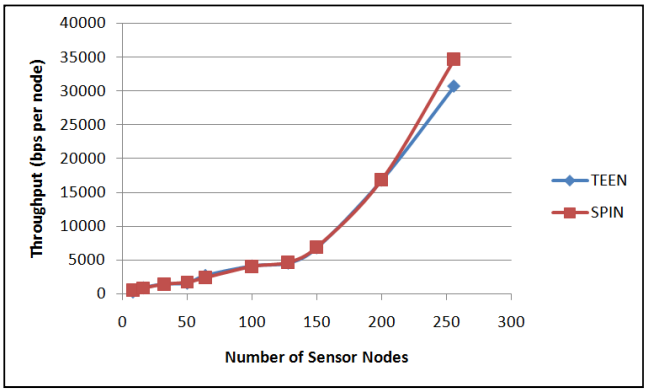

Fig. 10. Throughput comparison using logarithmic distribution.