

Journal of Artificial Intelligence

ISSN 1994-5450

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

Design and Development of Routing Protocol for WSN Simulation in GloMoSim

¹R. Alageswaran, ²O.G. Kiruthiga, ²T.K. Keerthika and ³B. Prakash

¹Department of CSE, K.L.N. College of Engineering, Madurai, Tamil Nadu, India

²Student, Department of CSE, K.L.N. College of Engineering, Madurai, Tamil Nadu, India

³Linux L1 Engineer, WIPRO InfoTech, Chennai, India

Corresponding Author: R. Alageswaran, Department of CSE, K.L.N. College of Engineering, Madurai, Tamil Nadu, India Tel: 9787715335

ABSTRACT

Nowadays, Wireless Sensor Networks (WSN) are emerging because of the technological developments in wireless communication and the advancements in miniaturization of sensor hardware nodes. WSN is an *ad hoc* network with more number of nodes densely deployed. Sensor nodes cooperatively pass the data through the network to a sink node. The research area of WSNs is growing and testing of new protocols and mechanisms developed for WSN application in real environment is infeasible as the number of nodes in WSN application is more. The use of simulation environments is an alternative to test the performance of an application prior to actual deployment. Global Mobile Information System Simulator (GloMoSim) is a open source network simulation software which simulates wired and wireless networks. GloMoSim uses the OSI reference model by using a layered approach. It supports parallel programming language called Parsec which is used for parallel discrete event simulation. It offers basic functionality to simulate wireless networks and does not support any sensor specific features and protocols for WSN simulation. Hence, without any further efforts in terms of adding protocols with the layers of GloMoSim, no WSNs can be simulated using GloMoSim. In this paper, the design, implementation of WSN routing protocol Sensor Protocol for Information via Negotiation (SPIN) and interfacing in network layer of GloMoSim is proposed. This addition makes GloMoSim suitable for WSN simulations.

Key words: WSN, GloMoSim, AODV, SPIN, CSMA, MACA

INTRODUCTION

WSN applications are emerging because of the developments in wireless communication technology and miniaturization of the hardware. WSN consists of number of low-cost and low-power sensor nodes. These sensor nodes are used to monitor physical conditions, such as temperature, sound, vibration, pressure, motion, etc., Sink node is the master node which collects the data from nodes and to processes it. The nodes are limited in power, computational capacity and memory. Since sensor nodes have strong constraints on their energy usage, this data transfer needs to be energy-efficient to maximize network lifetime (Trigoni *et al.*, 2007). The applications of WSN include agriculture, target tracking, health care monitoring system, etc. Wireless sensor network is a type of *ad hoc* network with unique features (Akyildiz *et al.*, 2002). The number of nodes used in the sensor network application is higher than the number of nodes in a wireless *ad hoc* networks. Nodes are densely deployed and prone to failure. There is no fixed topology in wireless sensor

network. The nodes are not having global ID as the number of nodes is very high. Biagioni and Sasaki (2003) and Xu *et al.* (2006) noted that Application-specific deployment, random deployment and grid deployment are node deployment approaches used in WSN.

GloMoSim is a library based sequential and parallel simulator for simulating wireless network. GloMoSim supports parallel programming language called Parsec which is used for parallel discrete event simulation. GloMoSim supports various *ad hoc* network protocols in data link, transport and application Layers (Nuevo, 2003). The main limitations of GloMoSim with respect to WSN simulation is no sensor specific routing protocol support in network layer of GloMoSim.

WSNs differ greatly from traditional *ad hoc* wireless networks and therefore require the use of new types of network protocols that are energy-efficient (Van Hoesel *et al.*, 2004).

WORKING PRINCIPLE OF SPIN

Routing in WSNs is very challenging due to the inherent characteristics that distinguish wireless sensor networks from other wireless networks like mobile *ad hoc* networks or cellular networks (Al-Karaki and Kamal, 2004). SPIN (Kulik *et al.*, 2002) is a routing protocol used to send information in a WSN application. The working principle of SPIN is shown in Fig. 1. SPIN protocol assigns a name to the data, called Meta-data and meta-data negotiation is done prior to data transmission. This negotiation reduces redundant data sent which consumes more energy in the network. Three types of messages used in SPIN are Data Advertisement (ADV), Data Request (REQ), Sending actual data (DATA).

GloMoSim supports various *ad hoc* network protocols in data link, transport and application layers. The reasons for using GloMoSim for this work are as follows: (1) GloMoSim is a open source software. (2) It is easy to simulate wireless *ad hoc* networks in GloMoSim. (3) It supports scalability which is considered as an important design issue in WSN application development. It allows the simulation up to thousand nodes. The current version of GloMoSim does not offer any sensor network specific network layer features, such as routing protocol support, in the default package. In order to make GloMoSim suitable for WSN application simulations, WSN routing protocol SPIN is implemented through four modules and interfaced with network layer of GloMoSim.

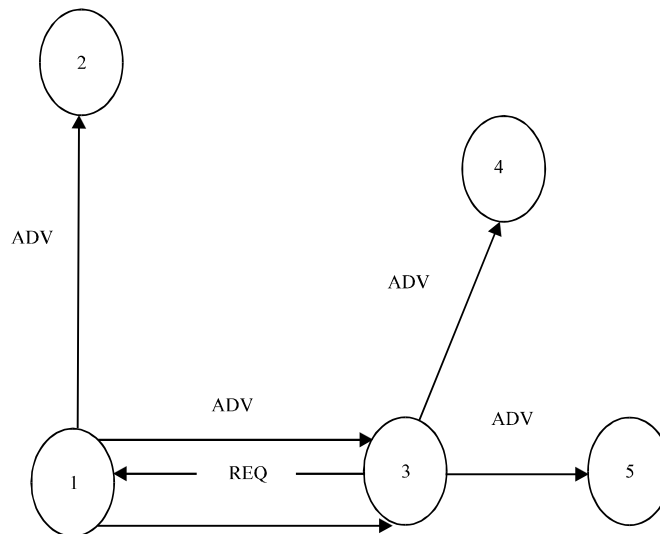


Fig. 1: Working principle of SPIN

ALGORITHMS

The assumptions considered in this paper are: (1) Network consists of homogeneous nodes. (2) Nodes are distributed randomly. The header file `spin.h` and the source file `spin.pc` files are created. This header and source files are appended with the existing routing protocol in the network layer of GloMoSim. SPIN is implemented through four modules given below:

- Data advertisement
- Data request
- Data message
- Interfacing Network and MAC Layer

Data advertisement: The steps for data advertisement is given in Algorithm 1. To overcome the situation of sending non-useful and redundant information in the network the node advertise the data using ADV message containing metadata a priori to transmission of the actual data. The size of the metadata is smaller than the size of actual data in SPIN in order to reduce overheads. The nodes send the ADV message and all nodes within the transmission range receive the ADV message.

Algorithm 1:

1. Create an instance for the GloMoRoutingSPIN.
 2. Declare the "AdvPkt" as "SPIN_ADV_Packet *AdvPkt;" and assign "int PktSize = size of (SPIN_ADV_Packet);"
 3. Allocate memory for message.
 4. Assign PktType as SPIN_ADV
 5. Assign the bcstId by calling RoutingSPINGetBcastId (node);
 6. Assign SrcNo as Node_No;
 7. Send the message to its neighbor node within the transmission range without an IP Header.
 8. Add IP Header to the message.
 9. Increment the variable "NumAdvsent".
 10. Set the timer for the destination
-

Implementation of data request: On receiving an ADV message, each node checks whether it has already received or requested the advertised data. If not, it waits for certain amount of predefined time and initiated the REQ message specifying original advertiser in the header of the message as shown in Algorithm 2.

Algorithm 2:

1. On receiving ADV message neighbor nodes will check
if ((current->adv == adv) and (current->reqsent != 0))
 return(TRUE);
else
 return (FALSE);
 2. Insert the details of packet sent in sent table and increase the TTL of that node
 3. Initiate the Request message as "RoutingSPINInitiateREQ(node, msg, node->nodeno);"
 4. Declare the "ReqPkt" as "SPIN_REQ_Packet *reqPkt;" and assign "int PktSize = sizeof(SPIN_REQ_Packet);".
 5. Allocate the space for message.
 6. Assign PktType as SPIN_REQ,
-

Algorithm 2: continued

7. Get the broadcasted and assign to `beastId`.
 8. Assign the `SrcNo` of `AdvPkt` to destination of `ReqPkt`.
 9. Send the request message to originator of the adv message.
 10. Increase the variable "NumRequestSent" and Set timer for the destination
-

Implementation of data message: The steps for implementing data message is given in Algorithm 3. On receiving the REQ message, the node will check whether it is original advertiser. If so, it will respond and send the actual data to the requesting node otherwise it will discard the REQ message. The negotiation mechanism ensures elimination of redundant data.

Algorithm 3:

1. On receiving the REQ message, Check for original advertiser
if (`AdvPkt_SrcNo = ReqPkt_dest`)
{
 Route original data to the requesting node;
 Increment the variable `stats.NumDataSent`;
 }
 else
 Discard the REQ message;
 2. Update the routing table and set timer to destination node.
 3. Check for assertion
 `assert(node->nodeno != ANY_DEST)`;
 4. `SendPacketToMacLayer`;
 5. Increment `Stats.umDataTxed`;
 6. Update the lifetime and set timer of the node
-

Interfacing network layer with MAC layer: In order to proper functioning of routing protocol in network layer it is mandatory to interface network layer with MAC layer. After implementing SPIN in network layer, the following functions in GloMoSim are called for interfacing network and MAC layer:

- **NetworkIpSendRawMessage:** Adds an IP header to a packet and calls function `RoutePacketAndSendToMac` to add routing information to the packet.
- **NetworkIpSendRawMessageToMacLayer:** Adds an IP header to a packet and sends the packet to the MAC Layer.
- **NetworkIpSendPacketToMacLayer:** Sends an IP packet to the MAC Layer (Nuevo, 2003).

RESULTS AND DISCUSSION

The simulation output of GloMoSim i.e., `glomo.stat` with AODV routing protocol and SPIN protocol are shown in Fig. 2 and 3, respectively. From Fig. 2, it is seen that `glomo.stat` file generated using AODV protocol contains the routing statistics of each node such as number of route requested, number of reply packets sent, throughput, etc. From Fig. 3, it is evident that `glomo.stat` file with SPIN as routing protocol in network layer contains routing statistics of each node such as number of data advertised, requested, number of packets sent, throughput, etc.

```

C:\WINDOWS\system32\cmd.exe - edit glomo.stat
File Edit Search View Options Help
C:\glomosim\glomosim\bin\glomo.stat
Node: 0, Layer: RoutingAdv, Number of Route Errors (RERR) Re-sent = 1
Node: 0, Layer: RoutingAdv, Number of CTRL Packets Txed = 2
Node: 0, Layer: RoutingAdv, Number of Routes Selected = 0
Node: 0, Layer: RoutingAdv, Number of Hop Counts = 0
Node: 0, Layer: RoutingAdv, Number of Data Txed = 0
Node: 0, Layer: RoutingAdv, Number of Data Packets Originated = 0
Node: 0, Layer: RoutingAdv, Number of Data Packets Received = 8
Node: 0, Layer: RoutingAdv, Number of Packets Dropped or Left waitin
Node: 0, Layer: RoutingAdv, Number of Broken Links = 0
Node: 0, Layer: RoutingAdv, Number of Broken Link Retries = 0
Node: 0, Layer: NetworkIp, Number of Packet Attempted to be Sent to
Node: 0, Layer: NetworkIp, Number of Packets Routed For Another Nod
Node: 0, Layer: NetworkIp, Number of Packets Delivered To this Node
Node: 0, Layer: NetworkIp, Total of the TTL's of Delivered Packets:
Node: 0, Layer: NetworkIp, Average Hop Count Assuming 64 Initial TTL
Node: 0, Layer: NetworkIp, Number Fragments dropped because Node wa
Node: 0, Layer: NetworkIp, Number Fragments dropped because TTL exp
Node: 0, Layer: AppChrServer, (0) Client address: 2
Node: 0, Layer: AppChrServer, (0) First packet received at [s]: 2.0102
Node: 0, Layer: AppChrServer, (0) Last packet received at [s]: 2.02072
Node: 0, Layer: AppChrServer, (0) Average end-to-end delay [s]: 0.0079
Node: 0, Layer: AppChrServer, (0) Session status: Closed
F1=Help | Line:25 Col:1
    
```

Fig. 2: Content of Glomo.stat file with AODV protocol in network layer

```

C:\WINDOWS\system32\cmd.exe - edit glomo.stat
File Edit Search View Options Help
C:\glomosim\glomosim\bin\glomo.stat
Node: 0, Layer: NetworkIp, Number of Packet Attempted to be Sent to
Node: 0, Layer: NetworkIp, Number of Packets Routed For Another Nod
Node: 0, Layer: NetworkIp, Number of Packets Delivered To this Node
Node: 0, Layer: NetworkIp, Total of the TTL's of Delivered Packets:
Node: 0, Layer: NetworkIp, Number Fragments dropped because Node wa
Node: 0, Layer: NetworkIp, Number Fragments dropped because TTL exp
Node: 1, Layer: RoutingSPIN, Number of advertisement Txed = 4
Node: 1, Layer: RoutingSPIN, Number of Data Packets Originated = 0
Node: 1, Layer: RoutingSPIN, Number of request= 4
Node: 1, Layer: RoutingSPIN, Number of Data Txed = 0
Node: 1, Layer: RoutingSPIN, Number of Data Packets Received = 0
Node: 1, Layer: NetworkIp, Number of Packet Attempted to be Sent to
Node: 1, Layer: NetworkIp, Number of Packets Routed For Another Nod
Node: 1, Layer: NetworkIp, Number of Packets Delivered To this Node
Node: 1, Layer: NetworkIp, Total of the TTL's of Delivered Packets:
Node: 1, Layer: NetworkIp, Number Fragments dropped because Node wa
Node: 1, Layer: NetworkIp, Number Fragments dropped because TTL exp
Node: 1, Layer: AppChrClient, (0) Server address: 0
Node: 1, Layer: AppChrClient, (0) First packet sent at [s]: 0.00000000
Node: 1, Layer: AppChrClient, (0) Last packet sent at [s]: 0.015000000
Node: 1, Layer: AppChrClient, (0) Session status: Closed
Node: 1, Layer: AppChrClient, (0) Total number of bytes sent: 2048
F1=Help | Line:27 Col:1
    
```

Fig. 3: Content of Glomo.stat file with SPIN Protocol in network layer

On comparing AODV and SPIN, SPIN avoids the transfer of redundant data via negotiation which reduces the depletion of energy used for transmission. This ensures that only interested nodes will spend energy to receive data. AODV discovers the routes and transmits the data through the efficient path within a simulation time, whereas SPIN starts negotiation of data as soon as simulation starts thus it is able to send more number of data than AODV within the simulation time. The MAC protocol supported by GloMoSim is used for testing the performance of SPIN protocol. By including SPIN protocol in GloMoSim, the use of GloMoSim is extended to wireless sensor network applications.

CONCLUSION

For WSN applications, it is essential to use a protocol which is scalable, consumes less energy and adaptable, robust in terms of reliable packet delivery and predictable with bounded delay. But energy efficiency stands among the top attributes given the limited battery energy of sensor node in unattended environments. GloMoSim is simulator software without routing support for WSN. In this study, to make GloMoSim suitable for WSN simulation, WSN network layer routing protocol, SPIN is developed and interfaced with network layer of GloMoSim. The performance of SPIN in GloMoSim is tested by simulating a WSN with existing routing protocol supported by GloMoSim, AODV and the new protocol added in network layer of GloMoSim, SPIN. From the results it is understood that SPIN performs well in WSN applications than AODV by avoiding redundant information in the network which conserves the battery energy.

REFERENCES

- Akyildiz, I.F., W. Su, Y. Sankarasubramaniam and E. Cayirci, 2002. A survey on sensor networks. *IEEE Commun. Magazine*, 40: 102-114.
- Al-Karaki, J.N. and A.E. Kamal, 2004. Routing in wireless sensor networks: A survey. *IEEE Wireless Commun.*, 11: 6-28.
- Biagioni, E.S. and G. Sasaki, 2003. Wireless sensor placement for reliable and efficient data collection. *Syst. Sci.*, 5: 10-16.
- Kulik, J., W.R. Heinzelman and H. Balakrishnan, 2002. Negotiation-based protocols for disseminating information in wireless sensor networks. *Wireless Networks*, 8: 169-185.
- Nuevo, J., 2003. A comprehensible GloMoSim tutorial. <http://www.cs.virginia.edu/~jx9n/courses/cs656/glomoman.pdf>
- Trigoni, N., Y. Yao, A. Demers, J. Gehrke and R. Rajaraman, 2007. Wave scheduling and routing in sensor networks. *ACM Trans. Sensor Networks*, Vol. 3.
- Van Hoesel, L., T. Nieberg, J. Wu and P.J.M. Havinga, 2004. Prolonging the lifetime of wireless sensor networks by cross-layer interaction. *IEEE Wireless Commun.*, 11: 78-86.
- Xu, K., G. Takahara and H. Hassanein, 2006. On the robustness of grid-Based deployment in wireless sensor networks. *Proceedings of the International conference on Wireless Communications and Mobile Computing*, July 3-6, 2006, Canada.