International Journal of New Technology and Research (IJNTR)
ISSN:2454-4116, Volume-2, Issue-9, September 2016 Pages 01-05

Efficient Routing in Wireless Sensor Networks

Suneet Gupta, Kapil Kapoor

***Abstract*— Wireless Sensors requires energy for communication,sensing and processing. Mechanisms are discovered to reduce the communication between the sensors .Multicasting in Wireless Network is the technique where there is one sender and multiple receivers. That means there is no need of broadcasting . But my proposed model is using mulicasting and it also uses the inherent broadcasting property of Wireless links.This model is based on demand based protocol strategy where the initiation for receiving the data starts from the destination.This model assumes that a unique id is given to all the sensors. Another assumption in my model is that there are multiple receivers and only one source node. Each receiver is using a different session for searching and receiving information from the source node.Sink node which has interest in receiving data from source will broadcast a search message and then nodes in the communication range of sink will supply a feedback message which contains the id of sensor node sending the feedback message. After this step selection of path or reservation will take place.Path selection is done according to the id of the source node. An id of sensor node is selected which is nearest to the source node id. Along with this an entry is made in the sensor node whose id is selected.This is achieved by utilizing the feedbacks send by sensor nodes.According to my model, reservation means that when a sensor node finds some already present entry in some other sensor node, And if that entry belongs to some other receiver then sensor node will make an entry in other sensor node so that when the information is transmitted from the source node to other receiver then it will be received by the sensor node which has done its registration. Registrations are performed so that only least number of sessions related to other receivers will be active and less number of nodes to be used while sending data from the source node. Simulation showed less number of nodes are used while the information is coming from source node.**

***Index Terms* - Wireless Sensors, sensor node.**

I. Introduction

A Wireless sensor network (WSN) consists of spatially distributed autonomous sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants.The development of wireless sensor networks was motivated by military applications such as battlefield surveillance. They are now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control. In addition to one or more sensors, each node in a sensor network is typically equipped with a radio transceiver or other wireless communications device, a small microcontroller, and an energy source, usually a battery. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from hundreds of dollars to a few pennies, depending on the size of the sensor network and the complexity required of individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and bandwidth. A sensor network normally constitutes a wireless ad-hoc network, meaning that each sensor supports a multi-hop routing algorithm (several nodes may forward data packets to the base station).A communication session is achieved either through single-hop transmission if the recipient is within the transmission range of the source node otherwise.the network is called as multi-hop packet radio network.

Suneet Gupta, AGISPET, Chailchowk, Mandi
Kapil Kapoor, AGISPET, Chailchowk, Mandi

II. Literature Review

Multicast potentially optimises bandwidth consumption and node resources, when several users simultaneously participate in a communication session.Nevertheless, contrary to the expectations, IP multicast has not experienced widespread deployment, with the exception of IPTV. On the other hand, emerging Wireless Sensor Network (WSN) applications could greatly benefit from multicast and constitute another field where multicast can be an effective and efficient technique.

Wireless sensor networks are embedded networks that are highly restricted to energy, bandwidth and processing power. Even though wireless sensor networks operate with limited resources,sensor nodes are required to do more functions compared to nodes in other data networks. Sensor nodes work as both hosts for processing sensed data and routers for forwarding packets. To decrease such a workload in sensor nodes, a multicasting method can be proposed which minimizes the number of transmission and forwarding in each sensor node.Wireless Sensor Networks (WSNs) consist of battery-operated sensor nodes with limited processing capability. Such resource constrained characteristics distinguished WSNs from other data networks.Traditional networking involves communications between two end systems.However, important emerging applications like has several limitations when applied to mobile Recent advances in wireless communications. IPTV, remote teaching or videoconference,require simultaneous communication between groups of users.Multicast protocols can offer several benefits. The use of a set of point-to-point channels to support a virtual multicast environment results in a complex and inefficient process,mainly in wide area networks. When a source needs to transmit a message to n receivers using

point-to-point channels, it is necessary to transmit the same message n times. In the case of IPTV, where the number of receivers is extremely high, this is not only technologically impossible but also the required resources are prohibitive.The emergence of applications with inherent multicast requirements led to the development of native multicast protocols. In the case of IP networks,multicast support was typically based on the Internet Group Management Protocol to announce hosts interested in receiving multicast information, and on Protocol-Independent Multicast – Sparse Mode (PIM-SM), Multicast Border Gateway Protocol (MBGP) and Multicast Source Discovery Protocol (MSDP) to route multicast messages between core routers. With the increasing demand for multicast support, new protocols were proposed. The most promising protocol is the Source-Specific Multicast (SSM) protocols were proposed. The most promising protocol is the Source-Specific Multicast (SSM) protocol. According to this protocol, when a host decides to join a multicast group it is necessary to specify not only the IP multicast address, as usual, but also the source address or a list of source addresses that the node joining the multicast sessions accepts to receive information from. This source identification significantly reduces the routing complexity. SSM electronics and miniaturization supported the development of a new generation of multi-functional, low-cost sensor nodes.These new sensor nodes, with control components and communication functionality, are at the base of the development of Wireless Sensor Networks (WSNs).Wireless Sensor Networks are composed by a set of several nodes which can cooperate in order to perform certain measurements and tasks, and can re-organize themselves in an ad-hoc way. Typically, sensors collect ambient measurements, process them and transmit them to a sink node. The applicability of WSNs is becoming very high, and although some approaches have already been proposed, it is crucial to evaluate if: - multicast can be useful for the next generation Internet, which will integrate WSNs,- the current multicast protocols are well prepared for WSN environments.

III. DMRPS-Distributed Multicast Routing Protocol

A. Overview

DMRPS is a Two-Phase schema. It is initialized by one sensor node triggered by the pre-specified event, in the following termed Source Node,flooding an invitation packet to each sink. And then, it is acknowledged by sinks interested in the invitation, sending back a registration packet backward through those intermediate nodes to the source node. To correctly route the acknowledge and succedent data packets, intermediate nodes will record the upstream nodes id and downstream nodes id during this two-phase packet transmission, like the concept,Gradient, used in Directed Diffusion , while the main difference is, in DMRPS the direction of upstream and downstream is opposite to those of Directed Diffusion, in which the routing initialization is made by the only one sink.

B. Key Concept: Multicast Session Initialization

Each sensor node is triggered by a pre-specified event shall send report data packets to each sink.Unlike traditional routing schemas in WSN, the source node plays an important role to manage the multicast session in DMRPS. The protocol proceeds independently for each source sending to a multicast group (session). And the remaining description applies to a single session.First of all, a source node will periodically broadcast an Invitation intended to each sink when the node has data packets to send. This periodical invitation packet helps to explore the multicast tree and update the information of receivers to this source node during the request and response phase. When an intermediate node, a node hearing the invitation but not interested in it,receives the non-duplicate packet, will create an upstream node id entry in its upstream list and then re-transmit the packet. The upstream node id will be used to route back the registration (routing backward history). After hop by hop transmission, interested sinks receive the invitation at the first time, and a upstream node entry with the session id, just marked by the source node id, and the upstream node id, from which the sink is conveyed the invitation, is created, then a Registration packet is sent backward the shortest/fastest path as well as the invitation is retransmitted on and on until its TTL equals to ZERO. Once an intermediate node receives a registration, it firstly checks whether the node id carried in the registration matches its own, if does, it realizes that it is on the right path from source to the sink, and creates a downstream entry in its downstream list to record the id of the registration-transferring predecessor
(routing forward history), then subsequently broadcasts the registration,modified with a new upstream node id list according to its own upstream entry, to its own upstream nodes; otherwise the registration packet is just dropped. The construction of upstream and downstream list is the important foundation to creating distributed multicast routing. Finally,
the source will get all the registrations from interested sinks. Using these registrations, source node acts as the admin control for the event data report itself sending. Upon receiving the registration, if the sink passes the admission requirements, the source adds it to its Session Member list, (downstream list indeed) and the sink formally becomes the auditor otherwise the source will positively exclude the corresponding up-/down-stream entry for it in the up-/down-stream list of intermediate nodes along the path to the unqualified sink. Thus, the two-phase invitation and registration process constructs the multicast routing tree from the source to sinks.

IV. Introduction to Omnet++

OMNeT++ is an object-oriented modular discrete event network simulation framework. It has a generic architecture, so it can be (and has been) used in various problem domains:

- modeling of wired and wireless communication networks
- protocol modeling
- modeling of queueing networks

- modeling of multiprocessors and other distributed hardware systems
- validating of hardware architectures
- evaluating performance aspects of complex software systems in general, it can be used for the modeling and simulation of any system where the discrete event approach is suitable, and which can be conveniently mapped into entities communicating by exchanging messages.

Modules can be connected with each other via gates (other systems would call them ports), and combined to form compound modules. The depth of module nesting is not limited. Modules communicate through message passing, where messages may carry arbitrary data structures. Modules can may messages along predefined paths via gates and connections, or directly to their destination; the latter is useful for wireless simulations, for example. Modules may have parameters, which can be used to customize module behaviour, and/or to parameterize the model's topology. Modules at the lowest level of the module hierarchy are called simple modules, and they encapsulate behaviour. Simple modules are programmed in C++, and make use of the simulation library.OMNeT++ simulations can be run under various user interfaces. Graphical, animating user interfaces are highly useful for demonstration and debugging purposes, and command-line user interfaces are best for batch execution.OMNeT++ also supports parallel distributed simulation. OMNeT++ can use several mechanisms for communication between partitions of a parallel distributed simulation, for example MPI or named pipes. The parallel simulation algorithm can easily be extended or new ones plugged in. Models do not need any source so that data will be travelled through that be used for classroom presentation of parallel simulation algorithms, because simulations can be run in parallel even under the GUI which provides detailed feedback on what is going on. The simulator as well as user interfaces and tools are highly portable. They are tested on the most common operating systems (Linux, Mac OS/X, Windows)

V. Problem Statement

The problem with the wireless sensor nodes is that they consume more energy while communication. If we go with broadcasting of data then it is not feasible to send data in that way. As network will go down because of more energy consumed by the wireless sensors. Another way is to go with unicasting but this will create large no of paths while sending data from source to sink. So multicasting is the solution to this problem where there is one source and there are multiple receivers. Efficient Routing means less number of nodes to be used even if we have large no of receivers.This is very important nodes are limited in their resources.The main aim of this dissertation is to reduce the nodes while establishing the path from destination to the displayed while executing simulation. We took 100 sensor nodes while forming a Sensor network.We made an array of Sensor node while establishing a path from destination to. special instrumentation to be run in parallel -- it is just a matter of configuration. OMNeT++ can even the source by using intelligent reservation policy.Along with this paths are established intelligently. According to the various research papers multicasting shows better results than unicasting and broadcasting. According to our investigation our protocol used for routing is showing better results.

VI. Simulation of the Proposed Model by Omnet++

We have taken 100 sensor nodes and a communication range is defined for each sensor node.Communication range is set similar for all the sensor nodes. A sensor node can communicate with its neighbouring nodes present in the communication range.In Omnet++ 4.0 we used @ display parameter in order to define a radius of 50 units for the communication range. A circle is used around the sensor node with 50 units as its radius and those nodes which comes under this range are selected by the sensor node. In the same way this procedure is similar for the other sensor nodes in the sensor network.Sensor Network contains collection of sensor nodes. In Omnet++ 4.0 we made an NED file named as Sensor.ned and we used @ display property to specify the name of the image that will be displayed while executing simulation. We took 100 sensor nodes while forming a Sensor network.We made an array of Sensor node whose dimension is set to 100 in the SenNetwork.ned file. We set the battery capacity of each sensor node in the beginning as 100.Battery capacity is taken as an attribute of sensor node .We have specified a parameter with the help of @display where the first sensor node and other sensor nodes will be displayed during execution of simulation. Each sensor node starts from the location (100,100) and then second node is displayed with the deltax added to x parameter.Our value of deltax is 50 that means if first node is displayed at 100,100 then second node is displayed at 150,100 and third will be displayed at 200,100.For the first row y value is same but while going in the next row the value of y will be incremented by deltay . The value of deltay taken in our simulation as 50. So the x co-ordinate of the nodes in the second row will be changed in the same way as in the first row and the only change in the second row will be in the value of y which will be changed to 150. We made a message file named as Packet.msg so as to send messages from one sensor node to another sensor node. In Packet.msg file we kept various fields which are used nnnwhile communication between the sensor nodes. When we make a.msg file in Omnet++ 4.0 then Omnet++automatically make .h and .cc files for the .msg file.We can modify .h and .cc files that are automatically made by the simulator.But in our simulation we have not made any modification to .h and .cc files. But when .cc and .h files are made then we are free to use the functions related to the fields we specified in .msg file. We used a field name in .msg file which is of type string. So we can set some string for our message by using functions like setName("message") and to retrieve the name of the message we can use the function like getName( ). In the similar we have used getSessionid( ) ,setSessionid( ), getSenderid( ) and setSenderid( ) functions while sending messages.There is one configuration file omnetpp.ini file in which we have set our

network that is SenNetwork.For programming of the sensor nodes we have made a Sensor.h and Sensor.cc files. In Sensor.h file we used various variables. For eg. We have used a variable of type cModule in order to get a reference to the node present in the Sensor Network so as to send message to that node.SendDirect() is the one of the method we have used in .cc file so as to send the message from a sensor node to another sensor node. We have defined initialize(),handle Message() and finish() methods in .cc file. Our maximum coding is in handle Message( ) which automatically gets called whenever a message is arrived on a module. For example if we send a message from one sensor node to another sensor node then when message is arrived on the another sensor node then handle Message is called. In the initialize function we initialized various variables that we are going to use in the simulation especially in the handleMessage( ) function. In finish( ) method that is called for all the modules whenever the simulation is terminated,we have removed the entries present in the data structures of type cArray. Initialize() method is also called for all the modules. The variable which we declare in header file and initialized in initialize method of cc file are unique for every module. For example if we make an int x variable in the header file then that variable if initialized in initialize method then it will be initialized for all the modules present in the network. According to our simulation the x value for 100 nodes will be initialized to 0 if x is initialized to 0 in initialize( ) method.

We took five sensor nodes as sinks when we started investigation.One node was taken as source. We have node[23] as our source node. We initialized five sensors which were acted as sinks. We made 5 messages and we set the name for all the messages as Hello4.Along with this we set a different sessionid for all the handleMessage() function for node with index messages as well. We took 0 sessionid for first sink message,1 sessionid for second sink message and soon for other sink messages.Senderid for the messages is set as -1 for all the sinks and the receiver is another parameter that is set for all messages and the value for it is set as 4.We set 4 value for receiver because our session id starts from 0 and we have 5 sinks.We have used a statement If(this-> getIndex()==5) in the initialize method so as to make a sensor node with index value 5 as sink. In the same way other sinks are also selected. The meaning of above if statement is that whenever the initialization of node with index equal to 5 is going on,only then the message with name Hello4 will be made and is scheduled for node with index 5.This process will be similar for all the sinks that we have taken.When a message is scheduled in Omnet++ for a particular module,then message is stored in the FES(Future Event Set) data structure of Omnet++ and handleMessage () function gets called for the messages that are scheduled. Handlemessage() function is called for the module that has scheduled its message and execute according to the time,message is scheduled.For example if message for node with index 5 is scheduled before a node with index 6 then handlemessage() function will be called for 5 node before calling 6. In handleMessage() function,I have used various if blocks where we check the name of the message and according to that processing is carried out.If a message has its name as Hello4 then node will send the message to the neighbouring nodes present in the communication range of the sensor node.For developing the logic for sending the message from a node to its neighbouring nodes we used a getDisplayString( ) method and we stored its value in the object of type cDisplayString. Then we found the co-ordinates of the sensor node by using another method of cDisplayString class and that is getTagArg().This getTagArg( ) returns a value of type const char * and conversion is carried out from string to integer,so that we can use that value in order to get the location of node present in the sensor network.In this way we accessed the values that we set in the .NED file. We accessed these values from the Sensor.cc file.In the same way radius value is retrieved by us.After finding location of the node we made an arrangement so that we would send the message from one sensor node to its all the neighbours present in the communication range by taking help we retrieved from getDisplayString( ) function. Message with name “n” is used when message is transmitted to the neighbours of the sensor node. When message with name “n” is processed then it means that message will be processed individually at the neighbours of the sensor node who send the “n” messages.According to our simulation the message with name “f” is send to the node which has transmitted various “n” messages. After this step,while processing of “f” message, selection of path or reservation will take place. Path selection is done according to the id of the source node. An id of sensor node is selected which is nearest to the source node id. Along with this an entry is made in the sensor node whose id is selected.This is achieved by utilizing the feedback messages “f” send by the sensor nodes.According to my model, reservation means that when a sensor node finds some already present entry in some other sensor node, And if that entry belongs to some other receiver then sensor node will make an entry in other sensor node so that when the information is transmitted from the source node to other receiver then it will be received by the sensor node which has done its registration. Registrations are performed so that only least number of sessions related to other receivers will be active and less number of nodes to be used while sending data from the source.Reservation to the another node is made by sending a “reservation” message.

VII. Results And Discussions

Multicasting is a technique which has its advantages of less communication overhead involvement. In our simulation of the proposed model we used the mixture of broadcasting , multicasting and unicasting techniques.As there is necessity of developing some protocols for routing in wireless sensor network which does not require maintaining a global routing substructure, link state or distance vector, etc for multicast tree structures. Distributed multicasting technology is the most energy-efficient way to cope with this problem. This technique has been used in our proposed model. According to cluster-based organization technique used in wireless sensor network where overhead is of maintaining cluster head is not used in our model. We used intelligent path selection and intelligent reservation in our proposed model which helps in setting path that is closest to the source node and in the same

way reservation is made with the sensor node that is closest to the source node.Distributed routing is implemented as whenever a path is made from one sensor node to the another then routing table of the node is updated in upstream(while going from sink to source node) with the sensor id and in the same way other sensor nodes operates.According to analysis the less no of hops are used when we took 5 sensors,10 sensors,15 sensors and 20 sensors respectively. We found that our proposed model is showing better results.According to DMRP (Distributed Multicasting Routing Protocol, Ominiscient and Diffusion,we find that performance of our proposed model better than the three methods mentioned above. Our proposed model is working for the case when we deploy the sensors in the sensor network in a particular way we have simulated.We have simulated by taking 10 sensor in a row and 100 sensor overall taken.Future work of this proposed model is to change the sensor node in the way we have arranged and try to discover some methods so that our policies of path selection and reservation can be applied in that scenario.

A. Analysis

We found that our proposed model is giving good results in accordance to the DMRP (Distributed Multicasting Routing Protocol,Ominiscient and Diffusion

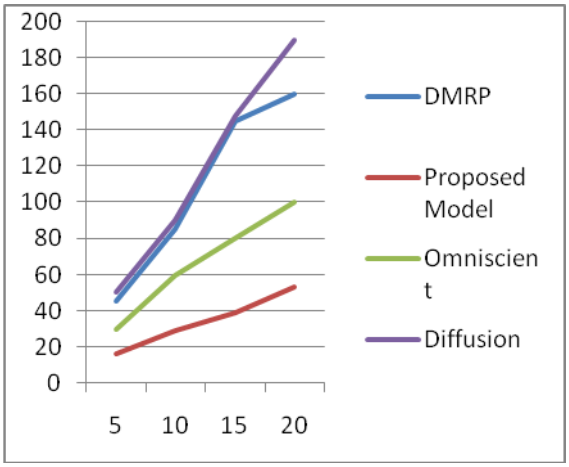

Fig 1: Total Length Vs Number of Sinks

This graph shows total length of constructed routes in y axis and number of sinks on the x-axis.

SUMMARY AND CONCLUSIONS

We have worked over Omnet++ 4.0 for Windows and investigated on the topic Efficient routing in wireless sensor network Using Multicasting. We reviewed research papers and we simulate our proposed model on Omnet++ 4.0 simulator. According to the Results and Discussions we found that new protocols are to be designed which uses Distributed routing.Some mechanisms need to be discovered which utilizes less no of sensor nodes while communication.Our investigation showed good results which will be used by the researcher community for further work so as to decrease the number of sensor nodes used in the communication when more number of sinks are involved.

REFERENCES

[1] https://en.wikipedia.org/wiki/Wireless_sensor_network
[2] Th. Arampatzis, J. Lygeros, and S. Manesis, “A Survey of .,lm Applications of Wireless Sensors and Wireless Sensor Networks”, Proceedings of the 13th Mediterrean Conference on Control and Automation, Limassol, Cyprus, June 27-29, 2005, pp 719-724.
[3] David Culler, Deborah Estrin, Mani B Srivastava “Overview of Sensor Networks”, In IEEE Computer ,vol:37 , no:8 , pp:41-49 , August 2004.
[4] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E.Cayirci. “A Survey on Sensor Networks”, IEEE Communications Magazine, August 2002. Page(s) 102-114.
[5] Qingfeng Huang, Chenyang Lu and GruiaCatalin Roman, “Spatiotemporal Multicast in Sensor Networks”, SenSys’03, November 5–7, 2003, Los Angeles, California, USA.
[6] Qingfeng Huang, Chenyang Lu, and Gruia-Catalin Roman, “Mobicast: Just-in-Time Multicast for Sensor Networks under Spatiotemporal Constraints”, International
[7] Workshop on Information Processing in Sensor Networks (IPSN'03), Lecture Notes in Computer Science 2634,Sringer Verlag, April 2003.
[8] A. Sheth, B. Shucker, R. Han, "VLM2: A Very Lightweight Mobile Multicast System for Wireless Sensor Networks", IEEE Wireless Communications and Networking Conference (WCNC) 2003, New Orleans, Louisiana, pp. 1936-1941.
[9] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks", in the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA,August 2000.