ISSN 1757-4439 (Print)
ISSN 1757-4447 (CD-ROM)

Communications of SIWN

Volume 6 ▪ April 2009

Editor-in-Chief
Huaglory Tianfield

The Systemics and Informatics World Network

siwn.org.uk

Communications of SIWN
April 2009 Volume 6 pp. 166-170

Energy Consumption Comparison between Autonomous and Central Wireless Sensor Network

Amir M Jafari, Adam Sklorz and Walter Lang

Institute for Microsensors, -actuators and -systems (IMSAS), University of Bremen
Email: {Jeffrey; asklorz; wlang}@imsas.uni-bremen.de
www.imsas.uni-bremen.de

Abstract: In order to implement wireless sensor network in automation process application, two configurations are considered, Autonomous and Central. In this paper these two structures are compared from the energy consumption viewpoint. Central and autonomous networks are simulated by Prowler simulator. In order to simulate the networks, SCAR routing algorithm is implemented in the second section. Prowler is used as simulator which is described in section 3. It is explained how it is modified and which energy model is implemented. In section after 50 random topologies are examined and their average is extracted to make the results independent from the topology. The networks' and nodes' energy consumptions are separately monitored to compare their behaviors. The result shows the ratio of central network energy consumption to the autonomous one. It is depicted the nodes' energy consumption in central network is distributed with higher standard deviation. Based on the result it will be discussed how the autonomous structure is more beneficial.

Keywords: wireless sensor network, autonomous network, central network, energy consumption, energy consumption distribution, prowler, sequential coordinate routing algorithm, automation process.

1. Introduction

Implementing wireless sensor network (WSN) in autonomous logistic system [11] has led to the new research area of comparing the autonomous and central network structure for WSN. The autonomous network and the related concepts can be generalized to discrete event control system in automation process applications. As example of such applications intelligent container [11] and Heating, Ventilating and Air Conditioning (HVAC) system can be named. In order to compare the functionality of the two networks' structures, the networks energy consumption and the nodes energy consumption distribution are considered in this paper.

In WSN, which is used for sensor/actuator wireless network in this paper, all nodes are equipped with a wireless transceiver, a tiny event driven microcontroller and batteries as power supply. "Tmote sky" from "Moteiv" [2] is taken for experiment and simulation as sample of such nodes. The radio range of the radio transceivers is limited, therefore mesh topology is applied for establishing the network [3]. Applying mesh topology for the networks requires a routing algorithm for forwarding the message one by one to a destination.

In central network architecture, the sensors send their information to the center of the network. When collecting the information is completed by the center, it performs the predefined control tasks and makes decisions. Then it sends instructions to the actuator(s) in the network. In a central network with mesh topology, nodes communicate either from the center to the node or from the node to the center. In this case, a routing algorithm called sink-oriented is applied. Sink-oriented algorithms support node-center or center-node path finding.

In an autonomous network each node has the capability to make decisions and perform discrete event control tasks like turning on-off the actuators. The nodes perform their predefined control task. If they need the data from other nodes for making a decision, they can send a request and ask for data. After making the decision, sensor can send instructions (decision) to other actuators (sensors). A routing algorithm called target-oriented is needed to send a message directly to another node without intervene of a center [5].

In order to assess the networks and nodes energy consumption behavior, they have to be simulated. There were two things to be prepared for simulation. The first one was the base topology and routing algorithm, the second one was the simulator. For producing the base topology, random generator and graph theory concepts were used. 50 random topologies with maximum 200 nodes were simulated in order to have independency of the simulation result from the topology. In each topology 200 random pairs were chosen as sensor-actuator pairs. For messages routing over the mesh topology, Sequential Coordinate Routing Algorithm (SCAR) [5] was applied. This algorithm has dual functionality; it can work as the sink-oriented algorithm as well as the target-oriented algorithm, therefore it could cover both networks. It means that in final simulation result, the difference between autonomous network and central network is independent from the routing algorithm properties, because both of the networks use the same routing algorithm. Generating of random topologies and implementing SCAR are detailed in the next section.

For network simulation described in this paper a probabilistic wireless network simulator (Prowler) [1] was used. In comparison to other simulators like TOSSIM [4] [9], Prowler has the advantage that different tasks could be assigned to different nodes while in TOSSIM application for

sai: cosiwn.2009.04.036

all nodes must be the same. The random topologies with SCAR algorithm are inserted in Prowler. For measuring the energy consumption, energy model of the nodes wireless transceiver was implemented in Prowler. In section 3 overview of Prowler is given and implementation of random topology, SCAR algorithm and node's energy model are explained. Results of the simulation and an optimization factor are discussed in section 4. Differences among the two networks functionalities are also discussed in this section.

2. Random Topology Generation and SCAR Implementation

In order to simulate autonomous and central network, initially a base topology should be defined. Topology of the network could have impact on the result of the simulation; therefore result could be manipulated by predefining the topology. To avoid this problem, topology is defined by random scattering of points. Two hundred (200) points are scattered with their Cartesian coordinate by MATLAB random generator. Then with the help of the channel attenuation function of the simulator (Prowler) radio model, received signal strength in other nodes is calculated. When a node signal reception is greater than the minimum required reception level, the two points are connected. In this way a graph is constructed with the nodes and their connection. This graph is not necessarily a connected graph [8]. It could have some isolated components, which represents the isolated nodes. They are removed by the usage of the Deep First Search (DFS) [10] algorithm. The remained graph is the component with the maximum number of the connected nodes. This graph is called network graph. Deleting the smaller isolated components does not have an impact on the simulation result, because each of them could be considered as a different form of random generation output.

After constructing the base topology, the SCAR algorithm is set up. In SCAR, the message is routed based on the sequential coordinates associated with each node [5]. When a node receives a message, it looks at the destination's coordinates inserted in the message header. Subsequently based on it's coordinates and the destination's coordinates, the node finds the next hop's coordinates. After inserting the coordinates in the message header, the node broadcasts the message. When the next node receives the message, it looks at the next hop's coordinates. If they are not the same as the receiver's coordinates, receiver node ignores the message; otherwise it follows the above procedure to find the next hop's coordinates again. In case that the destination's coordinates are the same as the receiver node's coordinates, message is arrived at the destination and transmission task is over.

In order to assign the sequential coordinates to the nodes, the Minimum Length Tree (MLT) of the network graph is extracted. Before this step the edge weight of all the connected pairs should be defined. The edge weight is defined as the difference between transmitted and received signal powers. In reality, the edge weight is computed by difference between the signal transmission power and Received Signal Strength Indicator (RSSI) [7] register value. By defining the edge weight in this form, they represent the channel attenuation between two nodes. MLT is a tree of this graph so that the total summation of its edges' weight is minimal. Based on the claim in [5] the total network energy consumption over the infinite time transmission would be less than any other trees of this network. MLT is extracted by "Kruskal" algorithm [8]. After extraction of MLT, the sequential coordinates are assigned to the nodes.

In the central network, the place of the center has a significant impact on the network's energy consumption. Therefore based on the graph theory concept, center is located on the median of the network [6]. A node that has the minimum summation of distances from other nodes in a network is called Median. In median of the network, the nodes consume minimum energy to transmit the message to the center. On the other hand, location of the center is defined by the topology properties. Consequently any result in the energy consumption is driven from the topology properties and not intervened by the other factors like human choice. Median is found by "Dijkstra's" algorithm [6]. Since the result of the simulation is dependent on the generated random topology, this process is repeated for 50 times in order to build 50 random topologies. The final product is a set of 50 topologies with random distribution and not-fixed node numbers. Each node has a specific and unique coordinate in each topology. The simulation is run for all these 50 topologies with autonomous and central architecture, then the average of all the 50 results are computed as the final result.

3. Prowler Modification with Energy Model

3.1 Prowler Introduction [1]

Prowler is a MATLAB-based probabilistic wireless network simulator. It is developed based on the Berkeley MICA platform which works with TinyOS. Prowler is event driven like TinyOS and applications are called by firing an event. It can incorporate any number of nodes and applications. One of the positive factors of Prowler is that the nodes can be distinguished by their IDs. Also different applications or statuses can be defined for each of them, whereas in some common simulators like TOSSIM it is not possible [4] [9]. Prowler supports a Graphical User Interface (GUI) with command line and visualization.

Each event is defined by a name and all the events are inserted in an event handler queue. After performing an event task, event handler orders the events queue with respect to events' timing and runs the events with higher priorities. Timing in Prowler is based on the bit timing. It assumes the bit rate as 40 kb/s then 40 thousand time units is counted as one second. Simulation will run for specific period of time or limited number of events. After one message reaches the destination or a message was forwarded by all the nodes in flooding, the first sender node is initialized for sending the message one more time by running the "Clock_Tick" event.

Prowler uses a simple carrier sense multiple access protocol for MAC-layer. It waits for random time interval, and then it checks the transceiver channel. If the channel is occupied, it awaits for a random back-off time interval and then checks the channel again. If it is not busy, it transmits the message. If channel is busy, it repeats the process four times. Consequently if the channel is busy at all four repeats, it drops the message.

Prowler has the capability to detect data collision. It can use two models. The simpler model used in this paper, checks the status of the receiver nodes. If they are in "transmit" or "receive" mode, data collision happens. Prowler uses a channel model for computation of received signal strength. It can apply the environment noises by random probabilistic value on the received signal quality. By using this technique it is able to model the environmental noises and nondeterministic nature of the communication channel. Prowler provides a "sysstat.m" file for analyzing the result of the simulation and extracting the nodes and the system statistics.

3.2 Prowler Modification

In order to use the Prowler for the simulation of autonomous and central networks, a few changes have to be done. These changes refer to the parameter changes and adding the application and the simulation presentation. Prowler supports a graphical user interface, which is not required in this application. On the other hand it takes a long time to run the simulation. Therefore the visualization and GUI were removed and the core of the simulator is used for the simulation.

Whenever the above simulation runs, a message is sent to a receiver. This change is led to remove the re-initialization of the simulation task during the simulation period. It means that the simulation in new condition will be over if the message reaches destination during the simulation time. After running each simulation, the "simstat.m" is called and system and nodes statuses are saved for later statistic analysis.

Since this research is based on the "Tmote Sky" platform, some parameters in simulation were changed. Prowler parameters are set for MICAZ with 40 Kbits/s bit rate, therefore bit time is set to 1/40000 s. In Tmote sky, the CC2420 radio chip is implemented which works with 250 Kbiots/s. Considering this fact, the bit time was changed to inverse the new bit rate (1/250Kbits/s). In order to have a better resolution, the generated Cartesian coordinates of each topology were multiplied by 10 and the "RECEPTION_LIMIT" was set to 0.5. This reception limit with radio model function (refer to Eq. (1)) is almost equal to 0.1 in a one square unit dimension plane. Prowler considers output transmission power indeterministic by incorporating a variance coefficient. Since in our energy model the programmable transmission power is taken into consideration, this variance parameter is fixed to zero in order to have deterministic behavior.

$$f(x) = P_{rec\_ideal}(x) = \frac{P_{out}}{1+x^2} \qquad (1)$$

When a packet does not arrive to destination for any reason, the packet is defined as lost. For recording the number of the lost packets, an event "Packet_Lost" is added to the events. In the "topology.m" file, new topology structure is generated. The routing algorithm is integrated in "channel_idle_check" event. If the channel is idle, the SCAR is called in order to find the next hop. Based on the radio propagation data, the required transmission power for next hop is computed. In reality, the transmission power can be computed by looking at the last received RSSI value from the receiver node. By clarifying the transmission power the overhearing nodes are identified. Because their reception energy is part of the network energy consumption. In addition their reception can cause the data collision with other nodes. In this simulation it is assumed that all nodes work in same channel. In order to simulate the energy consumption, the transceiver energy model was implemented in Prowler.

3.3 Energy Model

The physical layer energy model is divided into two parts: reception and transmission. The reception energy consumption is calculated by Eq. (2). The radio chip supply voltage (V_{DD}) and the reception current (I_{rec}) are extracted from CC2420 data sheet [7]. The time "t" is computed by multiplication of the TinyOS packet length which is equal to 40 bytes (320 bits) and the inverse of the radio chip bit rate 250 Kbits/s (1/250Kbits/s).

$$\begin{aligned} P_{rec} &= V_{DD} \times I_{rec} \times t = \\ &2.1 \times 0.0197 \times 40 \times 8 \times (250000)^{-1} = \\ &52.95\ \mu J/packet \end{aligned} \qquad (2)$$

Table 1. Energy consumption for each bit transmission in 8 levels

Level	I_{trans} (mA)	W_{trans} /per bit (μJ/bit)
1	8.5	0.0714
2	9.9	0.083
3	11.2	0.094
4	12.5	0.105
5	13.9	0.117
6	15.2	0.127
7	16.5	0.138
8	17.4	0.146

The transmission energy is computed in the same way; the only difference is that because of the different transmission levels the current consumption in each level is different. Table 1 shows the transmission energy consumption for each bit. The values of I_{trans} are from CC2420 data sheet [7]. In Prowler whenever a message is transmitted or received, even if it is a collided message, the energy consumption is computed. The value is placed into energy consumption field of the corresponding event. In "simstat.m" by analyzing the events queue, the corresponding node's ID and energy consumption are extracted and inserted in the node's statistics. The node statistics of each topology are used for simulation result assessment. It is important to note that this simulation is just for data transmission and it does not include the microcontroller process energy consumption. The wireless nodes usually work in sleep mode. Whenever they receive an interrupt, they wake up. After performing the computation, they go back to the sleep mode. Transition from the sleep mode to the awake mode (or vice versa) cost energy. Since the transition energy in comparison to transmission or reception energy is very small, it is neglected.

4. Simulation

To compare the autonomous and central networks, a task is defined. In this task 200 random nodes are supposed to measure an environmental parameter like temperature. The decision corresponding to each parameter is made. Then 200 other random nodes receive 200 decisions based on the first group of the nodes' data. In reality, the first bunch of 200 nodes could be sensors (Group S) and the second bunch could be actuators (Group A) The data transmission is done one by one separately for 200 times so that a sender from the "Group S" communicates with a receiver node from the "Group A", then it is repeated again. The difference between autonomous and central network is where the decision is made. In the central network the nodes from the "Group S" send their data to a center. After center made the decision, it sends the result to the corresponding nodes in the "Group A". In autonomous network not only the "Group S" of the nodes measures the environmental parameters but also they make the decisions as well. Then they directly send their decisions to their corresponding node in the "Group A" of the nodes.

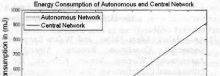

Fig 1. Networks' energy consumption for 200 random pair of nodes in 50 random topologies.

The horizontal axis of Fig. 1 and Fig. 2 show the transmission numbers between nodes from the "Group S" and nodes from the "Group A". The vertical axis of Fig. 1 shows network energy consumption in mill joule (mJ). In order to have independent result from a topology the above experiment is done for 50 random topologies. Each network energy consumption value on the Fig. 1 is average of the energy consumptions of 50 random topologies.

In Fig. 1 it is observed that for performing the above task in two networks, the central network consumes more energy than autonomous network. At the end of the task, the total energy consumption ratio of central to autonomous network is equal to 1.74 which means that the central network has consumed about 75 percent more than the autonomous one. A network is said to perform better when it consumes less energy for a task in comparison with other networks. In other words, less energy consumption for a task means better performance. Then, the performance is proportional to the inverse of energy consumption values. By considering this relation between energy consumption and performance, Fig. 2 shows the comparative performance of two networks. It can be seen that not only the performance of autonomous network is higher but also when the number of transmissions between the random pairs increases, the performance of the autonomous network drops faster than central network. These two results of autonomous network, less energy consumption and slower decrease of performance, are advantages of using an autonomous network. In order to have better resolution the horizontal axis is limited to the smaller numbers in Fig. 2.

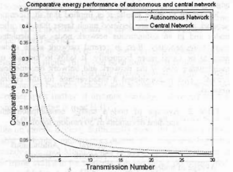

Fig 2. Comparative energy performance of autonomous and central network

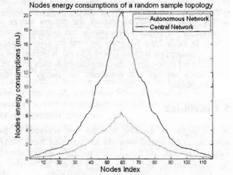

Fig 3. Nodes energy consumption of a sample random topology

In Fig. 3 the nodes' energy consumption of one random topology for the above mentioned task is depicted. This graph is produced after sorting the nodes' energy consumption values. The larger energy consumption values of the nodes are brought to the middle of the horizontal axis and the smaller values to the corners. In this graph it can be observed that the nodes' energy consumption in the central network is distributed more widely than the autonomous network.

In order to investigate this result in general case, the standard deviations averages of the node's energy consumption for 50 random topologies are listed in Table 2. Table 2 shows that the average of the nodes' energy consumption for 50 random topologies and their standard deviations average in the central network are greater than same parameter in the autonomous network. Briefly the conclusions are that the average of the nodes' energy consumption in the central network is greater than the autonomous one and also the network energy consumption is distributed over the nodes more evenly in the autonomous network.

Regarding to these results it is implied that in a central network some nodes are depleted much faster than the other nodes and in an autonomous network nodes are depleted more homogenously, then in central network some nodes need to be cared more frequently. It leads to increase of maintenance cost of the network and network become less robust. This is considered a disadvantage for the central network.

Table 2. Average of node's energy consumption and standard deviations of 50 random topologies

	Central Network	Autonomous Network
Aveage of nodes' energy consumption (mJ)	5.02	1.44
Average of standard deviations	5.17	1.4

In the autonomous network the process energy for computation of a control task is provided by the nodes' battery while in a central network the process energy is provided by the center which usually has unlimited power supply. In Table 2 the difference between the nodes average energy consumption is (5.02-1.44=) 3.58 mJ. It implies that if up to 3.58 mJ is devoted to the computation in each node, the autonomous network still consumes less energy than the central network.

5. Conclusion

In this paper, as the first step the two concepts of the autonomous and central networks are explained. Next, SCAR routing algorithm and Prowler as a simulator are introduced. By implementing the sensor energy model in Prowler, a simulation is run for autonomous and central networks. By comparison of the networks' energy consumption behavior and networks' energy consumption distribution over the nodes the following results are extracted. In the case of a node is being informed of another node's data or decision, generally a central network consumes more energy than an autonomous one. In comparison to an autonomous network, a central network's performance drops faster. An average node's energy consumption in an autonomous network is less compared to a central network. The network energy consumption distribution over nodes in an autonomous network is more homogenous than a central network. In central network some nodes are depleted much faster than others, therefore reduces the maintenance time period. It leads to have less robust network with more maintenance cost.

References

[1] Gyula Simon, Péter Völgyesi, Miklós Maróti, Ákos Lédeczi, Simulation-based optimization of communication protocols for large-scale wireless sensor networks, Submitted to 2003 IEEE Aerospace Conference, Big Sky, MT, March 8-15, 2003.

[2] Moteiv Corporation, tmote-sky-datasheet-102, www.moteiv.com 2006.

[3] Amir Jafari, Adam Sklorz, Design & Construction of an Ambience Monitoring System Based on a Wireless Sensor Network, University of Bremen, Bremen, Germany, 2007.

[4] Philip Levis, Nelson Lee, Matt Welsh, and David Culler TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications. In the Proceedings of First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003).

[5] Amir Jafari, Adam Sklorz, and Walter Lang, SCAR: Sequential Coordinate Routing Algorithm for Autonomous Wireless Sensor Network, the 2nd International Conference on New Technologies, Mobility and Security (NTMS 2008), Tangier, Morocco, 5-7 November 2008, pp 427-432.

[6] Gary Chartrand, Ortrud R. Ollermann, Applied and algorithmic Graph Theory McGraw-Hill, 1993. pp. 58–61.

[7] Chipcon AS, CC2420 Data Sheet, www.chipcon.com 2006.

[8] Dieter Jung nickel, Graphs, networks and Algorithms, Third Edition, Speringer-Verlag, 2008 pp. 1–28, 97-121.

[9] Thomas W. Carley, Sidh: A Wireless Sensor Network Simulator, Department of Electrical & Computer Engineering University of Maryland at College Park. ISR technical Reports, 2004.

[10]L.R. Foulds, Graph Theory Applications, Springer-Verlag, 1992. pp. 161–174.

[11]R. Jedermann, C.Behrens, R.Laur,W. Lang, Intelligent containers and sensor networks, Approaches to apply autonomous cooperation on systems with limited resources. In: Hülsmann, M.; Windt, K. (eds.): Understanding Autonomous Cooperation & Control in Logistics – The Impact on Management, Information and Communication and Material Flow. Springer, Berlin, 2007, pp. 365-392.

SIWN
The Systemics and Informatics World Network

http://siwn.org.uk/
Email: editorial@siwn.org.uk