An Autonomous Configuration Management Architecture for Wireless Sensor Networks

Muhammad Shoaib Siddiqui and Choong Seon Hong
Department of Computer Engineering,
Kyung Hee University
shoaib@networking.khu.ac.kr, cshong@khu.ac.kr

Abstract

This paper discusses the configuration management for the autonomic sensor network management, which is enabled by using a middleware. The detailed mechanism is discussed and scenarios for configuration management are also provided. Convergence time for the reconfiguration of the sensor nodes is also calculated using a mathematical model and simulations.

1. Introduction

We have focused our research on providing a comprehensive framework for Wireless Sensor Network (WSN) [1] management which should be autonomous, objective-aware and robust. In this regard, we have devised a management mechanism for providing autonomy and self-management in an application-aware WSN. Our mechanism is based on management agents, which resides in sensor nodes and utilizes the concept of virtualization to work in collaboration towards the management of the whole network. The data flow is based on information potential and is in the direction of lower potential from the higher information potential. This paper provides the conceptual framework for autonomic sensor network management, which is based on performance monitoring agents (manager agents) and configuration agents. The whole multi-agent based system is conceived by enabling virtualization at sensor nodes using a middleware. The mechanism is discussed in detail with the support of configuration management system.

Section 2 discusses some related work, while in section 3; we provide the architecture of the sensor node. Section 4 discusses the configuration management. Section 5 provides a mathematical model and simulations for the convergence time of the network reconfiguration and section 6 concludes our paper.

2. Related Work

WSNs are more unpredictable than traditional networks. With the deficiency of having scarce resources, it is tough to manage them. Management problems of WSN are tried to be solved by many researchers, in which policy based systems [2] have more promising solutions.

Management of WSN is not a new issue but recently, the research focus has been shifted to this problem. Some policy based [2] and P2P based systems [3] have mentioned solutions to the problem. Some other proposal for resource management are [4], [5], [6], [7], and [8]. All these proposals have introduced novel ideas for efficient management for WSN which are either light-weight or energy-aware. We propose a devisable management system for solving this problem, which is a preliminary effort in developing a comprehensive management framework for WSN in the future Internet.

3. Sensor Node

A sensor network consists of many sensor nodes, which are required to sense the environment and deliver the values to the sink node. The sink node uses these values to monitor the environment. All these nodes use same configuration although different sensor nodes have different sensors and application. To increase the efficiency of the network, the sensor network needs to be application aware. Hence, we provide configuration mechanism for the sensor nodes, according to the application that are running on the sensor nodes. The sink provides the basic configurations to each node according to the nodes' application.

The sensor node model is shown in Fig. 1. The configuration manager at the middleware is responsible for changing the configurations of the node and application dynamically. It uses the Configuration agents

† This research was supported by Future-based Technology Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (20100020728).

* CS Hong is the corresponding author.

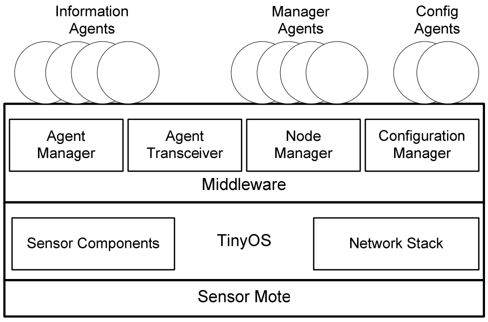


Figure 1. Sensor node system model.

for sending and receiving configuration from the management server at the gateway node. The node manager is responsible for the performance management of the sensor node and the network. Agent transceiver is responsible for assisting the mobile agents in migrating to or from the sensor node. Agent manager is responsible for creating, migrating, duplicating, comparing potentials and consuming the mobile agents.

3.1. Configuration Agent

Configuration agents are used for configuration of the sensor network and its nodes. These agents are totally different from the information agents. These agents are similar to data packets and contain configuration information for the sensor nodes. Configuration agents are used to access the configuration and application information of the sensor nodes.

4. Configuration Management

In a sensor network, each node is required to be configured according to the application it is running or reconfigured to optimize its performance. For this, each Sensor Node has a Configuration manager/agent, which is responsible for providing the application and node configuration to and from the Configuration Manager (sink node).

The primary goals for implementing configuration management are; (1) Enable the efficient Configuration for the Sensor nodes which helps in devising application-aware protocols for the sensor network. (2) Enhance and optimize the performance of the sensor nodes both as an individual and as a collaborating node towards a global goal or performance boost.

The system is consisted of two sub-systems, as shown in Fig. 2. One is the server at the sensor network gateway, which is the configuration manager at the sink node. The second sub-system is the configuration manager which resides and manages the configuration agents at the sensor nodes. Configuration Manager is responsible for

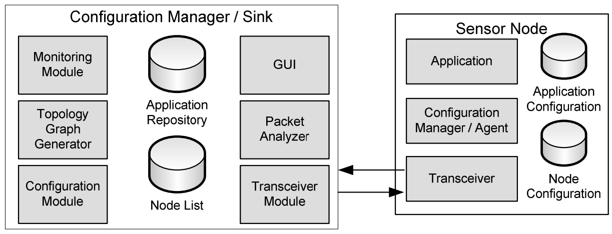


Figure 2. Information Flow and Architecture of the system

providing the sensor node application information to the sink node and applying the configuration parameter at the sensor node received from the Configuration Manager at the sink node.

The sink node should be able to know which node is running which application. Configuration manager at the sink maintains a topological graph of the sensor network.

There are two main data objects in the system. The first one is used to send the application information of the sensor node to the configuration manager. The other one contains the configuration information for the sensor node. The first data object is ApplicationInfo while the second data object is the Configuration. ApplicationInfo contains the node ID, the intermediate nodes and current configuration parameters. Configuration contains the sub-network ID for the node, its new configuration parameters and application parameters. These data elements are held within the configuration agents.

The information flow model of the system and the modular design of the system are shown in Fig. 2. Each Sensor Node has Configuration Agents, which are responsible for providing the application and node configuration to the Configuration Manager (sink node). It is also receives the configuration from the sink and apply them on the sensor node.

Configuration Manager at the sink node is responsible for creating the application aware configuration for the sensor nodes, and their applications. It also interacts with the user for changing the configuration within the sensor nodes. Configuration manager is also responsible for assessing the configuration of the nodes such that basic requirements of the network are not affected by the change in configuration. After assessing the changes as correct, it notifies the sensor nodes about the new values.

Configuration manager at the sink node has 3 important modules (shown in Fig. 2). Monitoring Module is responsible for monitoring and visualization of the sensor network. While the Topology Graph Generator is responsible for creating the logical topology of the network. Configuration Module is the main module which creates the configurations for each sensor node according to the application it is running. Node list is the directory of all the nodes with information about their application and the current configuration parameters. Application & Configuration repository contains the list of each application that is being run at the sensor nodes, along with its configurations.

The Configuration Manager makes reconfiguration decisions for the sensor node based on policies defined in the system. These policies are defined by a network manager or based on intuition & intelligence. The decision may be automated or monitored by a network manager. The automation of decision making is another study and out of the scope of this paper.

4.1. Mechanism

After the sensor node is started, it broadcast a 'hello' message for the neighbor discovery. After that it goes to a receiving state for a given time period, in which it receives the reply messages or 'hello' messages from the other nodes. When a sensor node receives a 'hello' packet it responds to it by a reply (or ACK) packet. According to the received packets it creates a neighbor list. The sensor node creates application information (Info) packet in which it includes its node configuration, application information and configuration, and neighbor list. This packet is then sent to the Configuration Manager at the sink. Intermediate nodes forward these packets to the CM. After collection these packets from the nodes, the CM creates a logical network topological graph of the sensor network. This graph is colored according to the different application being run on the sensor nodes. CM also maintains the information about that node such as, the application it is running, node configuration and the configuration data for the running applications.

According to the application information and network configuration, application aware application can be designed or the configuration of the running application can be changed. CM devise new configuration for the sensor nodes and their running applications and creates a response packet for the sensor nodes and unicast them to the sensor nodes.

When a sensor node receives its packet, the configuration agent at the sensor node is responsible to apply these configurations to the sensor node. After receiving the response packet the sensor nodes replies with an ACK packet. The graph and the configuration data at the CM is then updated according to the new information.

5. Convergence Time

When the configurations of the sensor nodes are changed, the time taken by the sensor network to implement these new changes is called the convergence time. We have mathematically analyzed the system for the convergence time, when the configuration of each node needs to be changed. We also performed simulation using NS2 and provide a comparison between the mathematical and simulation model.

5.1. Mathematical model:

Let us represent a sensor network *SN*, with *N* Number of nodes as a tree. If in this tree, each parent node has *C* number of child nodes then the depth of the tree *L* can be given by

$$N = C^L - 1 \quad or \quad \sum_{i=1}^{L} C^{i-1} = N \tag{1}$$

Here, we can say that if *C* is the average number of child nodes and root node is at *L=1* then *N* provides the approximate number of nodes in the *SN*.

Now, suppose the round trip time (RTT) between two adjacent nodes is *d* then the RTT between the root node and a node with depth *l* would be *(l-1)d*. For each node in the network to receive a packet from the root node and sends an acknowledgement, we have to sum up the delay faced up by each node. This total delay would be the theoretical convergence time of the sensor network *SN*.

As given by (1), we know that the number of nodes, which are at depth *l*, are C^l*-1* and the delay faced by each of these nodes is *(l-1).d*. Hence for each node, the delay depends upon the depth of the node, as shown in (2). Hence, the total delay D_{total} can be given by (3)

$$D_l = C^{l-1}.(l-1)d \tag{2}$$

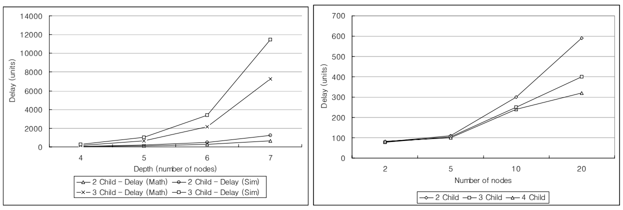
$$D_{total} = \sum_{i=1}^{L} C^{i-1}.(i-1)d \tag{3}$$

Here, D_l is the delay observed by nodes at depth *l*, while D_{total} is the convergence time for the sensor network *SN*. The relation between *L* and *N* is given by (1).

5.2. Simulation

We performed the simulation in NS-2. The network model was consisted of sensor nodes placed randomly within an area of 1000 x 1000 m^2. Each node has a propagation range of 150m with channel capacity 2Mbps. The size of the data payload is 512. Each run of simulation is executed of 900 seconds of simulation time. The medium access control protocol used is IEEE 802.11 DCF. The traffic used is constant bit rate (CBR).

The root node 'Node 0'sends a UDP packet to each node



(a) (b)

Figure 3. Convergence time for sensor network reconfiguration. Mathematical and simulated.

to simulate the scenario that each node is provided with a reconfiguration settings and calculate the time it takes to receive back the acknowledgement. If the acknowledgement is not received within certain time, a retransmission is done. The delay is calculated in units such that the round trip time between two adjacent nodes is *d=1* unit. This is done as to compare the simulation results with the mathematical findings.

In the first part of our simulation we increased the number of nodes in the network and analyzed the effect on convergence time for the reconfiguration of each node. We also consider the number of children nodes that each node has, as it also effects the convergence time. Fig. 3(a) shows the results obtained from the mathematical model and the simulation. On the x-axis, the number of nodes is shown as the average depth of the tree. With average number of children for each node is 2, the corresponding number of nodes to depth 4, 5, 6 and 7 are 34, 98, 258, and 642 respectively. Similarly for *C=3*, for *L* = 4, 5, 6 and 7, the value of N are 102, 426, 1641 and 6015, respectively.

On the y-axis we have calculated the delay in d units (where d is the RTT between 2 adjacent nodes). Form the graph, we can observe that as the depth of the tree (of sensor network) increases, the number of nodes increases exponentially and so does the delay. But the delay calculated by mathematical model is more than that of the simulated result when the number of nodes is increased. This is due to the fact that in the mathematical model, only one transmission can take place at a time, while in simulation simultaneous communications are possible.

In the second part, we simulate a scenario in which only few randomly selected nodes require reconfiguration and calculated their average convergence time, as shown in Fig. 3(b). On the x-axis is the number of nodes that need to be reconfigured and y-axis shows the delay in d units. The total number of nodes is limited to 100 nodes. We perform the simulation in three configurations; with each node having 2, 3 and 4 child nodes. Similar to the first part, the number of child nodes of each node effects the convergence time of the network. This is due to the fact that it reduces the depth of the sensor network tree and increases the number of simultaneous communications. Retransmission (packet drop due to interference) increases the delay, also.

6. Conclusion

We have devised an agent based mechanism for the management of the WSN. Our mechanism provides performance monitoring and configuration management for the sensor nodes and network. Using dynamic reconfiguration of the sensor nodes would increase the performance of the sensor nodes and the overall sensor network, but the convergence of network should be swift.

References

[1] I. F. Akyildiz et al, “Wireless sensor networks: a survey,” Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 38 n.4, 15 March 2002, pp.393-422,

[2] S. Calo , M. Sloman, “Guest Editorial: Policy-Based Management of Networks and Services,” Journal of Network and Systems Management, vol. 11 n.3, Sep. 2003, pp..249-252

[3] A. Binzenhofer et al, “A P2P-based framework for distributed network management”, Proc. Wireless Systems and Network Architectures in Next Generation Internet, 2006, pp. 198–210. Springer.

[4] Sean Walton and Eric Eide, “Resource management aspects for sensor network software. In PLOS 2007, Stevenson, WA, October 2007.

[5] Jason Waterman et al “Peloton: Coordinated Resource Management for Sensor Networks” in HotOS-XII, 2009

[6] Benkoczi, et al, "Data relaying with optimal resource management in wireless sensor networks", Proceedings of the IEEE International Workshop on Performance and Management of Wireless and Mobile Networks, Sydney, Australia, November 2005, pp. 617 - 618.

[7] Xiao-Hui Lin et al, "Energy Efficient Resource Management Techniques in Wireless Sensor Networks," in Handbook of Wireless Ad Hoc and Sensor Network, Springer, 2008.

[8] J. Gao et al. “Distributed resource management and matching in sensor networks”, In IPSN’09, pages 97–108, April 2009.