International Journal of Engineering Research-Online
A Peer Reviewed International Journal
Articles available online http://www.ijoer.in

Vol.2., Issue.6, 2014

REVIEW ARTICLE

ISSN: 2321-7758

HONEY ANT BASED CO-OPERATIVE DATA ACCUMULATION IN WIRELESS SENSOR NETWORK USING RANDOMIZED DISPERSIVE ROUTES

NAGARAJ.B.BARADELI
Tontadarya Collage of Engg, CS Dept, Gadag

Article Received: 19/10/2014 | Article Revised on: 10/11/2014 | Article Accepted on:13/11/2014

ABSTRACT

The proposed system introduces a novel cooperative storage mechanism that is based on the behaviour of Honey Ants. The proposed technique includes migratory data transporting agents called Ant Agents. In the proposed model, nodes with spare storage resource, volunteer to become 'repletes' which are then gorged with information by Ant Agents.The model overcomes the scale, heterogeneity and connectivity issues by placing the intelligence in its migratory execution units. The devices in the network cooperate by providing a common minimal system support for the receipt and execution of Ant Agents.

Keywords-Ant agent, traceability, security by RDP, revocation, wireless sensor network (WSN).

INTRODUCTION

Wireless sensor network is a collection of nodes organized into a cooperative network. Each node consists of processing capability (one or more microcontrollers, CPUs or DSP chips), may contain multiple types of memory (program, data and flash memories), have a RF transceiver (usually with a single Omni- directional antenna), have a power source (e.g., batteries and solar cells), and accommodate various sensors and actuators. The nodes communicate wirelessly and often self-organize after being deployed in an ad hoc fashion. Systems of 1000s or even 10,000 nodes are possible. Such systems can revolutionize the way we live and work.

Currently, wireless sensor networks are beginning to be deployed at an accelerated pace. It is not unreasonable to expect that in 10-15 years that the world will be covered with wireless sensor networks with access to them via the Internet. This can be considered as the Internet becoming a physical network. This new technology is exciting with unlimited potential for numerous application areas including environmental, medical, military, transportation, entertainment, crisis management, homeland defence, and smart spaces.

Wireless Sensor Networks have found wide acceptance in a range of user applications, due to their ability to monitor ambient conditions without human supervision. These networks typically consist of a large number of densely deployed, low-cost, low-power, multi-functional sensor nodes that interact with each other using wireless links.

The shrinking size and increasing density of wireless devices have profound implications for the future of wireless communications. Today's laptops and wireless phones may soon be outnumbered by ubiquitous computing devices such as smart dust , micro sensors and micro robots . In fact, a large number of low-cost, low-power, multi-functional sensors are already being deployed in many real life scenarios where monitoring ambient conditions is an integral part of the overall application.

Wireless Sensor Networks offer a practical, cost effective and capable solution to the problem of environment sensing, especially in scenarios where wired communication is not suitable or just not possible. Therefore, future sensor networks are expected to have a huge number of sensor devices that are dynamically connected to each other via wireless links. Because of their size limitations however, a large number of these sensor nodes are likely to have severe restrictions on their processing power, storage space, available memory as well as battery capacity. Unfortunately such rigorous resource limitations in wireless sensors preclude the full utilization of complex user applications in real life scenarios. Despite their individual shortcomings, cooperation among networked sensor nodes is known to result in large and powerful ad-hoc computing systems. Towards this end, we need to consider among other characteristics, the storage capacity of individual nodes as it is usually very limited, compared to the amount of data that a node generates.

Limited storage capability, coupled with temporal availability of sink nodes and varying importance of collected data mean that, the generated data needs to be stored locally, in the network, for later retrieval. To achieve this, we introduce a novel cooperative storage mechanism that is based on the behaviour of Honey Ants in the real world. Few nodes in the network, volunteer to become 'repletes' which are then satiated with data originating from other sensors in the network. The 'repletes' themselves do not produce any data and only act as extended storage for other sensors in the network. The proposed technique also includes migratory data transporting agents called Ant Agents, which are similar to Mobile Agents [in some aspects. These Ant Agents, originate at the sensing device and migrate the network, foraging for 'repletes' that have spare storage capacity. Once found, the agent deposits the data on the host 'replete' and returns back to its originating node to fetch the next piece of information.

1.1 Architecture Of Wireless Sensor Network

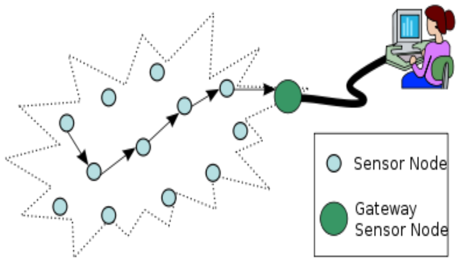


Fig 1.1: Sensor Network

Each such sensor network node has typically several parts:

- A radio transreciever with internal or external antenna
- A microcontroller for interfacing with sensors
- Energy source like a battery.
- Memory

Controller

The controller performs tasks, processes data and controls the functionality of other components in the sensor node. The most common controller is a microcontroller, other alternatives that can be used as a controller are generl purpose desktop microprocessor, digital signal processors, FPGAs and ASICs. A general purpose microprocessor generally has a higher power consumption than a microcontroller, therefore it is often not considered a suitable choice for a sensor node.

Transceiver

The functionality Of both transmitter and receiver are combined into a single device known as transceivers. Sensor nodes often make use of ISM band which gives free radio, spectrum allocation and global availability. The possible choices of wireless transmission media are Radio frequency (RF), Optical communication (Laser) and Infrared. Lasers require less energy, but need line-of-sight for communication and are sensitive to atmospheric conditions. Infrared, like lasers, needs no antenna but it is limited in its broadcasting capacity. Radio frequency based communication is the most relevant that fits most of the WSN applications. WSNs tend to use license-free communication frequencies: 173, 433, 868, and 915 MHz; and 2.4 GHz.

External memory

From an energy perspective, the most relevant kinds of memory are the on-chip memory of a microcontroller and Flash memory—off-chip RAM is rarely, if ever, used. Flash memories are used due to their cost and storage capacity. Memory requirements are very much application dependent. Two categories of memory based on the purpose of storage are: user memory used for storing application related or personal data, and program memory used for programming the device. Program memory also contains identification data of the device if present.

Power source

The sensor node consumes power for sensing, communicating and data processing. More energy is required for data communication than any other process. Power is stored either in batteries or capacitors. Batteries, both rechargeable and non-rechargeable, are the main source of power supply for sensor nodes.

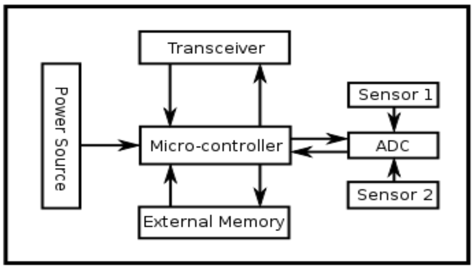


Fig 1.2 : A sensor node

1.2 Problem Definition

In the Existing system, a large number of sensor nodes are likely to have severe restrictions on their processing power, storage space, available memory as well as battery capacity. The storage capacity of individual nodes is very limited, compared to the amount of data that a node generates. Unfortunately such rigorous resource limitations in wireless sensors preclude the full utilization of complex user applications in real life scenarios. Limited storage capability, coupled with temporal availability of sink nodes and varying importance of collected data mean that, the generated data needs to be stored locally, in the network, for later retrieval.

Second issue that needs to be considered is the security of data transmitted over the network. Of the various possible security threats that may be experienced by a wireless sensor network (WSN), the compromised-node (CN) attack and the denial-of-service (DOS) attack are of more concern in this project. Severe CN and DOS attacks can disrupt normal data delivery between sensor nodes and the sink, or even partition the topology. More over once the routing algorithm becomes open to the adversary (this can be done, e.g., through a memory interrogation of the compromised nodes), the adversary can by itself compute the set of routes for any given source and destination.

1.3 PROPOSED SOLUTION

The proposed system introduces a novel cooperative storage mechanism that is based on the behaviour of Honey Ants. The proposed technique includes migratory data transporting agents called Ant Agents. In the proposed model, nodes with spare storage resource, volunteer to become 'repletes' which are then gorged with information by Ant Agents. The model overcomes the scale, heterogeneity and connectivity issues by placing the intelligence in its migratory execution units. The devices in the network cooperate by providing a common minimal system support for the receipt and execution of Ant Agents.

When a sensor node wants to send a packet to the sink, it first breaks the packet into M shares according to a $(T;M)$-threshold secret sharing algorithm, e.g., the Shamir's algorithm. Each share is then transmitted to some randomly picked neighbour. That neighbour will continue to relay the share it has received to other randomly picked neighbours, and so on. In each information share, there is a TTL field, whose initial value is set by the source node to control the total number of randomized relays. After each relay, the TTL field is reduced by 1. When the TTL count reaches 0, the final node receiving this share stops the random propagation phase and begins to route this share towards the sink using normal single-path routing.

To achieve this we include an algorithm called Non-repetitive Random Propagation NRRP, this adds a "node-in-route" (NIR) field to the header of each share. Initially, this field is empty. Starting from the source node, whenever a node propagates the share to the next hop, the id of the up-stream node is appended to the share's NIR field. Nodes included in NIR are excluded from the random pick of the next hop of propagation. This non repetitive propagation guarantees that the share will be relayed to a different node in each step of random propagation, leading to better propagation efficiency.

1.4 Objectives of Study

The end objectives of the study are:

- Understand the features and functionalities of sensor networks
- Develop a Co- operative Adhoc Network that leads to better storage efficiency.
- Understand the features of Honey Ant based Co- operative data accumulation.
- Design efficient Ant Agent software.
- Design robust Ant Routing Algorithm.
- Induce security in the system by deploying Randomized Dispersive Propagation of messages over the WSN.

2 DEVELOPMENT METHODOLOGY

The waterfall model is a model which was developed for software development; that is to create software. It is called as such because the model develops systematically from one phase to other in a downward fashion, like a waterfall.

As said earlier the waterfall model has been structured on multiple phases especially to help out the software construction companies to develop an organized system of construction. By following this method, the project will be divided into many stages thus easing out the whole process. For example you start with Phase I and according to this model, one only progresses to the next Phase once the previous one has been completed. This way one moves progressively to the final stage and once that point is reached, you cannot turn back; similar to the water in a waterfall.

Definition Study / Analysis: During this phase research is being conducted which includes brainstorming about the software, what it is going to be and what purpose is it going to fulfill.

Technical Design / Detail Design: After the basic design gets approved, then a more elaborated technical design can be planned. Here the functions of each of the part are decided and the engineering units are placed for example modules, programs etc.

Construction / Implementation: In this phase the source code of the programs is written.

Testing: At this phase, the whole design and its construction is put under a test to check its functionality. If there are any errors then they will surface at this point of the process.

Integration: in the phase of Integration, the company puts it in use after the system has been successfully tested.

Management and Maintenance: Maintenance and management is needed to ensure that the system will continue to perform as desired

.

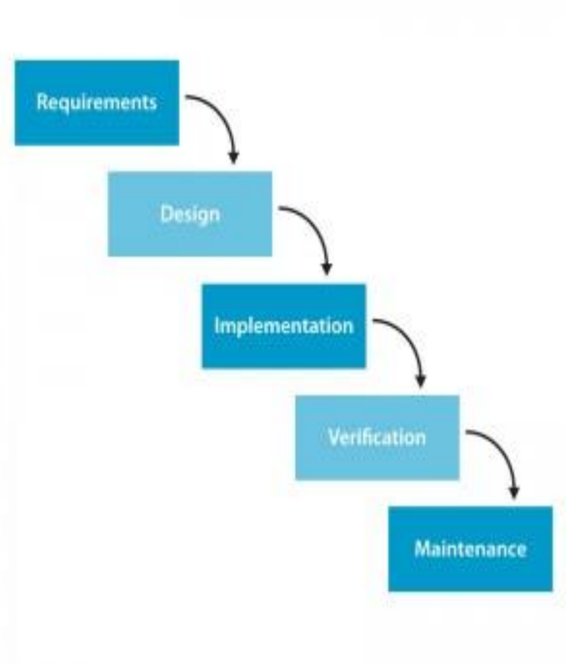


Fig:2.1 waterfall model

3 SYSTEM DESIGN

Design is the first step in the development phase for any engineered product or system. The process of applying various techniques and principles for the purpose of defining a device is to permit its physical realization.

The designer's goal is to produce a model or representation of an entity that will later be built. The process by which the model is developed combines intuition and judgment based on experience in building similar entities, a set of criteria that enable quantity to be judged and a process of iteration that ultimately leads to final design representation.

3.1 Design Principles

Basic principles have to be followed in the designing of a project. Design principles enable the software engineer to navigate the design process. The following

- The design should be traceable to the analysis model.
- The design should not reinvent the wheel or the cycle.
- The design should exhibit uniformly and integration.
- The design should be structured to degrade gently, even aberrant data events or operating conditions are encountered.
- The design should be assessed for quality as it is being created.
- The design should be reviewed to minimize conceptual errors.

3.2 Design Considerations

There are many aspects to consider in the design of a piece of software the importance of each should reflect the goals the software is trying to achieve. Some of these aspects are:

- Compatibility: The software is able to operate with other products that are designed for interoperability with another product.
- Extensibility: New capabilities can be added to the software without major changes to the underlying architecture.
- Fault tolerance: The software is resistant and able to recover from component failure.
- Maintainability: The software can be restored to a specified condition within a specified period of time.
- Modularity: The resulting software comprises well defined, independent components that lead to better maintainability.
- Reusability: The modular components designed should capture the essence of the functionality expected out of them and no more or less.
- Reliability: The software is able to perform a required function under stated conditions for a specified period of time.
- Robustness: The software is able to operate under stress or tolerate unpredictable or invalid input.
- Security: The software is able to withstand hostile acts and influences.
- Usability: The software user interface must be intuitive to its target user/audience. Default values for the parameters must be chosen so that they are a good choice for the majority of the users

3.3 System Overview

Wireless Sensor Networks have found wide acceptance in a range of user applications, due to their ability to monitor ambient conditions without human supervision. These networks typically consist of a large number of densely deployed, low-cost, low-power, multi-functional sensor nodes that interact with each other using wireless links.

Unfortunately such rigorous resource limitations in wireless sensors preclude the full utilization of complex user applications in real life scenarios. Despite their individual shortcomings, cooperation among networked sensor nodes is known to result in large and powerful ad-hoc computing systems. Towards this end, we need to consider among other characteristics, the storage capacity of individual nodes as it is usually very limited, compared to the amount of data that a node generates.

Limited storage capability, coupled with temporal availability of sink nodes and varying importance of collected data mean that, the generated data needs to be stored locally, in the network, for later retrieval. To achieve this, we introduce a novel cooperative storage mechanism.

The proposed technique also includes migratory data transporting agents called Ant Agents, which are similar to Mobile Agents. These Ant Agents, originate at the sensing device and migrate the network, foraging for 'repletes' that have spare storage capacity. Once found, the agent deposits the data on the host 'replete' and returns back to its originating node to fetch the next piece of information.

We also use Dispersed Random Propagation of the ant agents thereby enhancing the security of the network.

Architectural Design

Considering the resource constraints of the nodes involved, the system architecture of ACS aims to offload as much intelligence as possible on to the Ant Agents while keeping its device support requirements to a bare minimum. Figure 5 shows the system architecture support needed for the proposed ACS model. As depicted, the Security Manager first verifies the credentials of all incoming Ant Agents. AAs with valid security credentials are then queued up for execution at the Virtual Machine.

The Virtual Machine acts as a hardware abstraction layer for loading, scheduling and executing tasks generated by incoming AAs. It interfaces with the Resource Manager, which estimates the amount of storage space left in Tag Space. The Tag Space represents the name-based memory region that stores data objects persistent across AA executions.

When an AA is executed at the Virtual Machine, it first queries the Resource Manager to check if the hosting device is can be its 'replete '. If yes, the sensor data is now deposited in the Tag Space of the hosting device. If not, the AA continues its foraging phase by executing its ant routing algorithm. Post execution, the AAs are injected back into the network to allow them to migrate to their next destination.

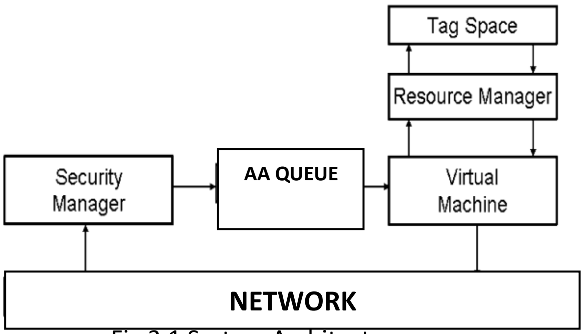


Fig 3.1 System Architecture

A.Security Manage

To prevent excessive use of its resources, a device needs to perform some form of admission control. In the proposed architecture, the Security Manager component performs this role. It is primarily responsible for receiving incoming AAs, verifying their security credentials and passing them onto the Virtual Machine for execution.

B. Virtual Machine:The Virtual Machine is a hardware abstraction layer for the execution of AAs across all heterogeneous hardware platforms present in the wireless sensor network. Examples include Java Virtual Machine, K Virtual Machine, etc.

C. Resource Manager: The Resource Manager is responsible for keeping track of the Tag Space usage. It is also responsible for making queries on the Tag Space, but hides away the implementation details from the Virtual Machine.

D. Tag Space: A Tag Space consists of a limited number of tags that are persistent across AA executions. Figure 6 from illustrates the structure of a tag. It consists of an identifier, a digital signature, lifetime information and data. The identifier represents the name of the tag. The access of AAs to tags is restricted based on the digital signature. The tag lifetime specifies the time at which the tag will be reclaimed by the device from the Tag Space. AAs can store data (like pheromone) at the Tag Space by creating their own tags.

ID	Digital Signature	Lifetime	Data

Fig 3.2 Tag Structure

ANT AGENTS

AAs like Mobile Agents, are migratory execution units consisting of code, data and an execution state. In ACS, the behaviour of AAs is modeled on the behaviour of honey ants in nature. Just as honey ants cooperate with each other to forage for food, AAs in wireless sensor networks cooperate with each other to forage for repletes. AAs are intelligent and are capable of routing themselves without needing any external support. When admitted for execution on the hosting device, the computation code within the AA is embodied into a task. During its execution this task may modify the data sections of AA, modify the local tags to which it has access or may migrate to another device.

A. Format

In addition to its identity and authentication information, an AA is comprised of a code section, a data section as well as a lightweight execution state. A digital signature together with the AA and sensor IDs identifies an AA. The digital signature is used by the host devices to protect the access to an AA tags. The code and data sections contain mobile code and sensor data that an AA carries from one device to another. The data section sets a bound on the expected storage needs of the AA at the host device. Figure 7 depicts the skeletal structure of the Ant Agent.

Digital Signature	Sensor ID	AA ID	State	Code	Data

Fig. 3.3 Format of an Ant Agent

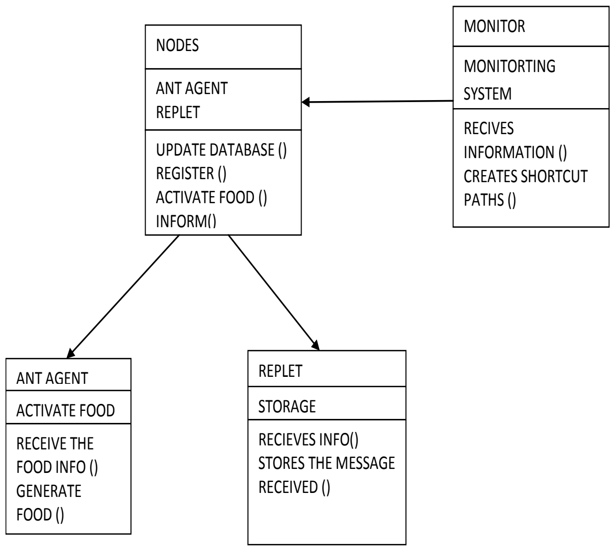
3.4 Modular design:

3.4.1 Class Diagram:

In software engineering, a class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. This project develops the system using 3 modules namely

- Monitor module
- Ant Agent Module
- Replete Module

The NODES class is a super class of the ant agent and the replete node, hence some functions here are inherited and some are overwritten. The diagram below shows the connectivity between the classes and their respective attributes and member functions



.

Fig: 3.3 class diagram of the system

3.4.2 Monitor Module Flow Diagram:

This diagrammatic representation can give a step-by-step solution to a given problem. Process operations are represented in these boxes, and arrows connecting them represent flow of control. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.

The process begins from Start into the Monitor operations which monitors and keeps log of the entire system. The Monitor will receive information regarding the data available in the network from the Repletes, based on this 2 steps are taken

- Send all ant agents to collect the data from the source
- Carry the data from the source to the spare node/ replete

After this the process stops, we can Refresh the process to start from beginning.

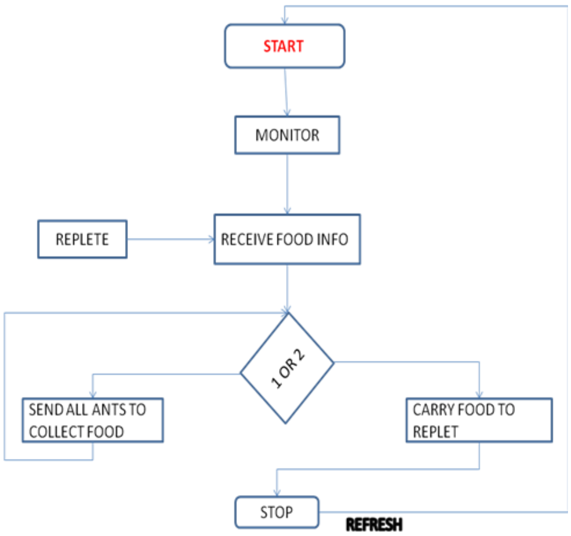


Fig: 3.5 flow diagram of monitor module

3.4.3 Ant Agent and Replete Module flow diagram:

The nodes are off 2 types namely the data generating normal nodes and the replete node. The replete nodes cannot introduce ant agents in the network. The Process begins from Start where the administrator registers as either ant agent on a node or as a replete.

- If it is registered as ant agent then these entities go in search of data generated so as to collect it and store in the replete node
- If the node is registerd as replete then it does not produce any data but stores the data that is brought in by the ant agents.

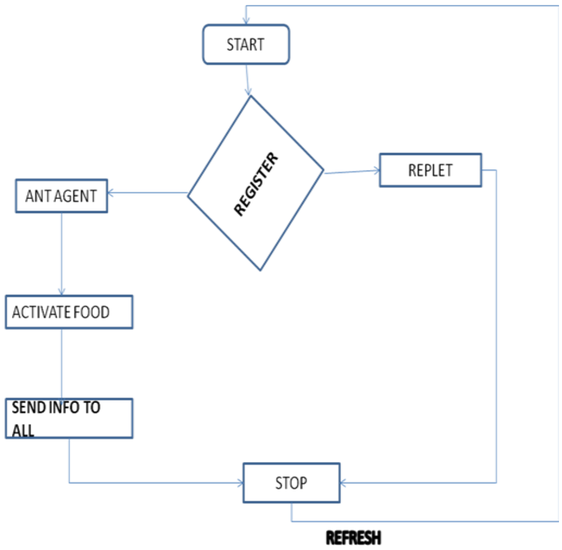


Fig 3.6: Flow Diagram of the ant agent and replete module

4 BIBLIOGRAPHY

Reference documents:

- Honey Ant based cooperative data accumulation in wireless sensor networks-Santosh Kulkarni and Prathima Agrawal 2011 ieee
- Secure Data Collection in Wireless Sensor Networks Using Randomized Dispersive Route - Tao Shu, Sisi Liu, and Marwan Krunz
- S. Kulkarni and P. Agrawal, "Ad-hoc cooperative computation in wireless networks using ant like agents:' in Proceedings of the 17th International Coriference on Advanced Computing and Communications (ADCOM), 2009.
- P. C. Lee, V. Misra, and D. Rubenstein. Distributed algorithms for secure multipath routing in attack-resistant networks. IEEE/ACM Transactions on Networking, 15(6):1490–1501, Dec. 2007
- "Scalable and efficient ant-based routing algorithm for ad-hoc networks:' IEICE Transactions on Communications, vol. E-89B, no. 4, pp. 1231-1238, January 2006.

Websites:

- http://www.IEEE Explorer.com
- http://www.google.com
- *www.java.net/blog/opinali/archive/android-java*
- http://www.wikipedia.com