Autonomous Messenger Based Routing in Disjoint Clusters of Mobile Sensor Networks

Nisar Hundewale, Qiong Cheng, Xiaolin Hu, Anu Bourgeois, Alex Zelikovsky

Abstract—**Most existing routing protocols in mobile ad hoc networks mainly focus on addressing the problems of dense and sparse yet connected networks. However, the crisis-driven and geography-driven applications typically challenge current routing protocols in disjoint mobile sensor networks (DMSN) where network partitions can occur and last for a significant period. In this paper, we apply agent-based simulation to studying the problems of efficient route discovery in the disjoint networks and present a novel autonomous messenger-based route discovery and routing protocol for disjoint clusters-based topology in DMSN. We have designed the framework for the route discovery and routing protocol. We design and implement two route discovery protocols, one based on straight line moving of messengers (SLMM) and the other based on flexible sharing policy of messengers (FSPM). Furthermore, we design the agent-based modeling and simulation in the application and implemented the prototype. The simulation of the framework prototype demonstrates that the route discovery and routing protocol based on FSPM increases network lifetime as compared to SLMM and also that FSPM exhibits higher data delivery than SLMM.**

Index Terms—**Greedy Algorithm, Dynamical Source Routing Protocol (DSR), Ad Hoc On Demand Distance Vector Routing (AODV), Connected Dominated Set (CDS), Disjoint Mobile Sensor Networks (DMSN), Straight Line Moving of Messengers (SLMM) and Flexible Sharing Policy of Messengers (FSPM).**

I. Introduction

With the advance of technologies in wireless communications and electronics, research and development of low cost, low-power, and multifunctional sensor nodes and their applications have thrived.

A mobile sensor network is composed of a large number of sensor nodes that include a mobile vehicle in each. It possesses the self-organizing and cooperative ability to detect, collect, process, predict and estimate some events of interest, especially in inaccessible terrains or disaster relief operations [1].

For the traditional Ad Hoc Mobile Wireless Network, currently, existing routing protocols may be categorized as table-driven routing protocols and source-initiated on-demand routing protocols [2]. Most of the existing work focuses on connected networks where an end-to-end path exists between any two nodes in the network [4].

Table-driven routing protocols attempt to maintain consistent, up-to-date routing information between each two nodes pair in the connected network. These protocols require each node to maintain one or more tables to store routing information [2]. And with the changes of network topology, these nodes respond to change the routing tables' information by propagating updates throughout the network in order to maintain a network view. Examples of existing tables driven routing protocols are destination-sequenced distance-vector routing, cluster head gateway switch routing, and the wireless routing protocol [2].

Source-initiated on-demand routing creates routes only when desired by the source node [2]. When a node requires a route to a destination, it initiates a route discovery protocol within the connected network and all possible route permutations have been examined once the process is done [2]. So the initialization of the discovery process updates the network view with the change of the connected network topology. This means that when sensor nodes move to a location that can be reached by another node, no routing path exists even if they detect the event of interest. So in this case, the detected event and data will be missed.

Different from traditional routing protocols, a message ferrying approach is developed in [4]. This method employs a set of special mobile nodes called ferries to provide communication service for regular nodes in the deployment area according to a specific route so that they are responsible for data delivery in sparse networks. In [4], the authors assume ferries have a long range radio so that it covers the whole deployment area and move faster than regular nodes. Consequently, in [4] authors ignore the route discovery in the sparse or disjoint networks. Channel sharing is another issue in such protocols since ferries need to co-ordinate globally among themselves in order to communicate with the base station without interference. As a result of channel sharing constraints the message delivery efficiency of the protocol would be affected drastically. Also, the protocol depends on the hardware capability and energy levels of the ferries. In case, ferries loose their energy then there will not be any communication between the network and the base station since non-ferry sensors can not perform the tasks of ferries due to their hardware limitation.

There are works demonstrating formation of energy efficient clusters in order to increase scalability and energy efficiency of communications in sensor networks [11]. However, such protocols emphasize on artificially forming clusters and choosing cluster heads. The performance of such protocols depends upon the formation of the clusters that is specific to the

All the authors are with Dept. of Computer Science, Georgia State University, Atlanta, GA 30303 USA. Emails: nisar@computer.org, qcheng1@student.gsu.edu, {xhu, anu, alexz}@cs.gsu.edu respectively.

topology or distribution of the sensors. Such protocols can not deliver messages if the clusters are disjoint or the cluster heads can not find another cluster head within its communication range.

In this paper, we address routing challenges in the naturally formed disjoint clusters of sensor networks. In disjoint clusters where no global knowledge of the other dynamically formed clusters is available, each disjoint cluster requires to find a route to the base station in order to deliver the sensory data and cluster knowledge to the base station. We present a distributed route discovery protocol that dynamically chooses messengers to travel close to the base station to communicate. The proposed protocol does not depend upon centralized control or prior global knowledge. It does not depend upon ferries or sensors with special hardware capabilities. The protocol can establish communication in disjoint network of clusters that has not been addressed previously. We pay attention to the route discovery and bring forth the autonomous messenger based route discovery and routing approach for route discovery in disjoint networks, one based on straight line moving of messengers and the other based on flexible sharing policy of messengers.

The naturally formed disjoint clusters in mobile sensor networks possess the nature of highly dynamical connection and disconnection which is not predicted and predetermined. Agent-based simulation adapts to the nature. Every autonomous agent corresponds to a self-configurable, self-adaptable and self-maintainable mobile sensor. We designed the agent-based simulator. It utilized the Discrete Event System Specification (DEVS) [12] and DEVSJAVA which was developed by University of Arizona[6].

The remainder of the paper is organized as follows: Section 2 presents the framework of autonomous messenger based routing. Section 3 introduces cluster based organization in network partition. Section 4 describes the messenger based route discovery protocols in disjoint networks and the two protocols we promote are discussed. Section 5 describes autonomous agents simulation based on DEVS & DEVSJAVA. Experiments and its result are presented in Section 6. Finally, we conclude the paper in Section 7.

II. A OVERVIEW OF AUTONOMOUS MESSENGER BASED ROUTING

A. Framework

Autonomous messenger based routing is a self-maintained approach for disjoint networks to reconfigure itself and adapt to changing connectivity and system stimulation. It includes three components, cluster based management component, routing path discovery component and routing component. For the route discovery component, we present two protocols. One is based on straight line moving of messengers and aims at making the messenger carrying data move in a straight line towards the base station and return back after sharing data with the base station. The second one is the route discovery protocol based on flexible sharing policy of messengers. The component framework graph is shown in Figure 1.

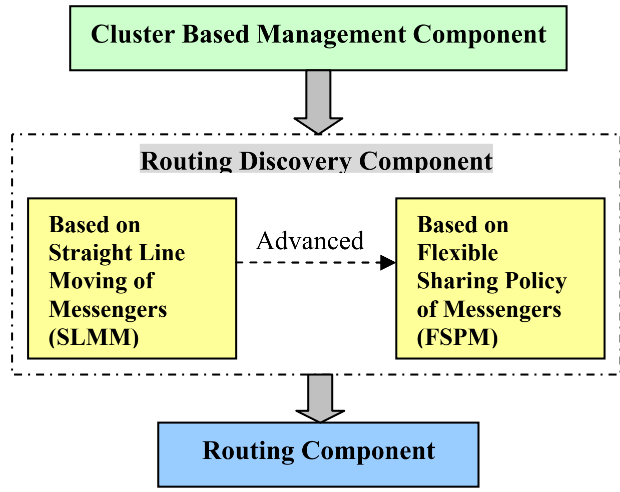


Figure 1. The framework

B. Cluster-Managed Topology Control in Every Network Partition

With the random advent of the interest events that take place at discrete points in space and time within a given area, mobile sensors form clusters around them. Every cluster may have an irregular periphery shape. Inside every cluster, we employ Dynamic Source Routing (DSR) as a routing protocol. The roles in a cluster include a cluster header, regular member and messenger. For every role, there exists a corresponding finite state machine. For example, a cluster header has ten states: power-on, power-off, stand-by, awaken, active, processing, detecting, selecting messenger, sending messenger and messenger returning. According to their different roles, there exists different action sets in which every action exhausts the corresponding battery energy amount.

The cluster management component consists of three packages: cluster members management package, messenger management package, and communication package.

C. Route discovery Component

Route discovery component aims at efficiently localizing the information of all clusters and sharing data between them. Here we assume that all mobile sensor nodes know the location of base stations. For simplification, in this paper, we assume that there is only one base station.

After studying parts of existing routing protocols, we present two route discovery protocols. The first one is based on straight line moving of messengers. After cluster header assigns the messaging task to the messenger, it carries data and moves out of the cluster towards the base station. Once it arrives at the base station, they share cluster data with each other. Here we ignore collision problems that could occur between sensor nodes and between sensor nodes and obstacles.

The second one is based on flexible sharing policy of messengers. Different from the first protocol, here a messenger can share data with clusters and other messengers when it moves towards the base station and passes by them. The design details will be discussed in Section 4.

III. Cluster based Organization in network partition

A. Cluster-Forming Process in DMSN

Every network partition will be a connected dominated set (CDS). For the CDS, cluster-forming process is indispensable and crucial because it is helpful to the localization service, query service and routing service. Furthermore, it is useful to the task management and mobility management, and even reduces the energy-consumption as the whole of DMSN.

In the cluster-based topology control of network partition, sensors' roles are categorized as cluster head, general sensor, messenger, and sink node by their responsibility in DMSN. Among them, a sink node has the powerful ability of communication by internet or satellite. Sometimes, a sink node is the representation of base station. Communication for the others depends on the mobile status of sensor nodes. Concretely, when sensor nodes, which are roaming in the supervised area, sense or actuate an event of interest (target), they may check if surveillance of the target exists by other sensor nodes. If they detect that the target is unmonitored, then they may enter into the transition range of the target, form a cluster and be a cluster head. Another sensor node may join in the cluster and be a cluster member when the cluster needs more members.

B. Cluster Management Mechanism

Cluster management includes the member management, data management and even action management such as dynamical routing management. Because most of our work focuses on the route discovery in disjoint mobile sensor networks, we just employ keeping every node proper state transition so that the entire network partition state is maintained.

IV. Messenger based route discovery and routing protocol in disjoint Mobile Sensor Networks (DMSN)

A. Background

In DMSN, there exist relatively stable network partitions between which any communication package in the one cannot reach the other. In every network partition, it is a connected dominated set which is organized in the clustering-based hierarchical control infrastructure [7]. Our research focuses on how the disconnected clusters discover each other in an efficient way. On the consequence, we bring forth two messenger based discovery protocols.

In developing the discovery protocols for the DMSN, we make the following assumptions:

1. An infrastructure has been built to localize the sensor nodes and targets by using dead reckoning localization, or land markers or beacons with feature of the error detection and error proofread [8]. So we assume that all sensors have the global position information;
2. We ignore the computational burdens and communication burdens in the different single nodes;
3. Every event of interest can be detected in time. It is possible to estimate the velocity and acceleration of the targets;
4. Energy decay assumption: Signal emitted from a source decays as the same power law as a function of source-sensor distance;
5. The range of the transmission is the disc-unit. With the radius of the disc-unit growing, the signal of transmission decays. We ignore the existence of obstacles;
6. The mobile sensor can avoid collision with other sensors and obstacles;
7. The mobile sensor has unique identity;
8. The mobile sensor moving in a constant speed;

We make three other assumptions mainly for simplicity purpose. First, we assume there is only one base station in the system. Secondly, we do not consider noise and inaccuracy in the sensing and moving of the sensor nodes. Thirdly, we ignore the accident event when the messenger disappears. We note that the discovery protocols developed in this paper should also work when these three assumptions are removed.

B. Messenger Assignment Policy

Cluster head appoints the general sensor with the highest energy in its cluster to be the messenger every time. For improving the efficiency and reducing the energy consumption in moving of messenger, condition 1 below and an alternative one of condition 2 and 3 should be met before the cluster head appoints a messenger and the messenger can execute the task.

1. When cluster head has full cached pipeline used to store the collected monitor data and cluster information;
2. When the waiting time is more than a min specified wait time before sending a messenger;
3. When the waiting time is more than a max specified wait time before resending a messenger;

The first condition guarantees that enough information has been collected before appointing a messenger. The second condition guarantees the proper time space for sending a messenger. The specified minimum wait time should meet the requirement that the number of messengers do not reduce the necessary number of regular member nodes and affect the detection quality of the whole cluster. The third condition guarantees the cluster will continue to send out the next messenger for sharing cluster information when the previous messenger disappears for a long time.

C. Discovery Protocol Based on Straight Line Moving of Messenger

1) Protocol Description

Once a sensor node in a cluster receives the request of sending out messages from the cluster head by multiple hops communication of the other neighbor nodes, it will move out of the cluster towards base station. When it arrives in base station, it will exchange the carrying data with base station. It means base station will give the other clusters information to the messenger. And then the messenger will carry the data and return to its original cluster. The process is shown in Figure 2. After iterating the process in different clusters, eventually all the cluster information will be shared in linear time.

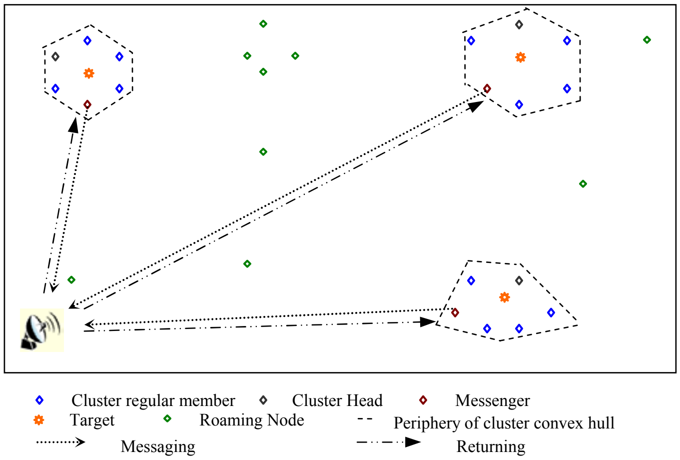


Figure 2. Planar diagram of messenger moving towards base station

2) *Communication Technique*

a) Communication Process

According to the protocol description in the previous sub section and the defined packet structure, the communication process is shown in Figure 3. Here the prerequisite process is that regular cluster members detect the target, collect data and report the collected data to cluster head. And then the steps of communication are as follows: [The step number is consistent with the label of Figure 3]

1. Cluster Head collects data from the regular members and stores them to the pipeline memory. These data will be transported as Regular Sensor Data Packet. When the cached pipeline is full, it will trigger the event for cluster head to select a maximum energy sensor node as the messenger. When the wait time interval calculated from the previous messaging event is more than the specified minimum wait time, or the wait time is more than the specified maximum leaving time for the previous messenger, it will enter step 2;
2. Cluster head generates Cluster Set Information Packet, and route the packet in its own connected dominating set to the chosen messenger by the on-demand routing solutions for mobile environments like Dynamic Source Routing (DSR) [9]. Messenger receives the Cluster Set Information Packet and retrieves the command type and trigger time. When the command type is 0, the messenger will enter step 3 after the trigger time;
3. Messenger carrying the cluster set information packet moves out of the cluster and goes towards the base station. It does not stop moving until reaching the base station. The moving messenger saves the track information. The cluster set information packet will include the cluster's own information and other cluster information that was due to the previous messengers' contribution;
4. At the base station, the messenger will unload the carrying packet. The base station retrieves the cluster set information, compares them with the local cluster set, and adds or updates the latest cluster information. It next constructs a packet for the updated local cluster set and sends the packet to the messenger;
5. The messenger carries the new packet and returns towards its cluster along the same track path. It continues step 5 until meeting its cluster;
6. When the messenger meets a regular member of its cluster, it will unload the packet from the base station and route the packet to the cluster head by the same routing solution in step 2.

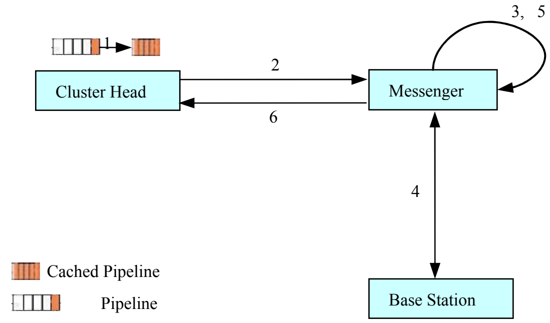


Figure 3. Communication protocol based on Straight Line Moving of Messenger

The labels attached to arrows represent the steps for communication

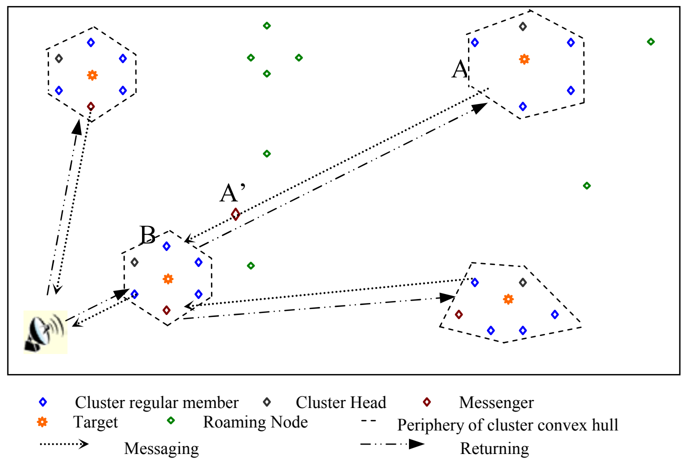


Figure 4. Planar diagram of messenger moving towards base station

The communication process literately happens so that cluster information will be shared in based station and in all the network partitions.

D. Discovery Protocol Based on Flexible Sharing Rule of Messenger

1) Protocol Description

The protocol using the flexible sharing rule has the same packet structure as the above protocol based on straight line moving of messenger. The main difference is that a messenger shares cluster information with intermediate clusters it passes by or with messengers from other clusters. For example in Figure 4, messenger A' carrying the information of cluster A moves towards the base station. When it encounters cluster B, it

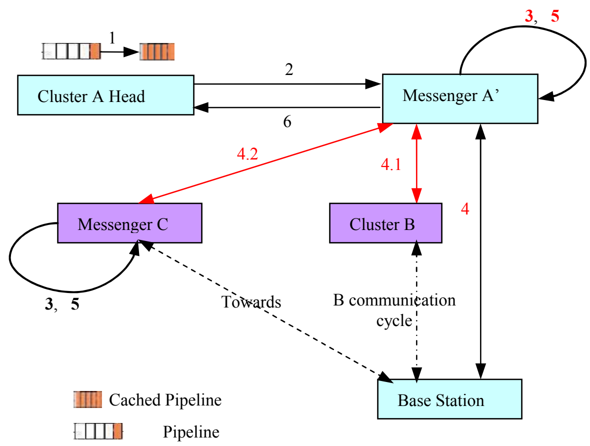


Figure 5. Communication protocol based on Straight Line Moving of Messenger

The labels attached to arrows represent the steps for communication

stops its movement to the base station and instead it unloads the packet at cluster B. After B sends A' the packet of the update local cluster set information, A' will return directly. Cluster B will then send a messenger carrying the mixed cluster set information to the base station in its next time step.

2) Communication Mechanism

Figure 5 shows the communication mechanism for the route discovery and routing protocol of flexible sharing rule of messenger. Compared with SLMM, the FSPM protocol has the same operation in step 1, 2, 4, 5 and 6.

3. The same as the previous protocol, messenger carries the cluster set information and move towards base station. The difference is that every the other *r/v* (assuming *r* is the transmission radius and v is the speed) time messenger floods a probing message, and waits for replying. When it gets response from the other cluster, it will enter step 4.1. When it gets response from the other messenger, it will enter step 4.2;

4.1 Messenger shares its carrying packet with the met cluster in the same way as sharing information with base station. If finishing the process, it will directly enter step 5. However, the met cluster sends out cluster information in its communication cycle. The cluster information may mix some information from the messenger;

4.2 Messenger shares its carrying packet with the met cluster in the same way as sharing information with base station. If finishing the process, it will directly enter step 5. However, the met messenger will continue its step 3 until meeting the other cluster or messenger or arriving at base station. Therefore, the moving process is also the one of greedily sharing carrying packet of a moving messenger.

V. Autonomous Agents Simulation Based on DEVS & DEVSJAVA

Due to the dynamic and non-predetermined connection and disconnection among sensor nodes in DMSN, we promote the Autonomous Agents Simulation for the routing protocol and develop a simulator prototype based on DEVS & DEVSJAVA.

A. DEVS and DEVSJAVA

The DEVS (Discrete Event System Specification) formalism is derived from generic dynamic systems theory [6]. The two kinds of DEVS models, atomic model and coupled model, may be flexibly combined to form a component-based hierarchy structure that can be applied to both continuous and discrete phenomena and systems [6]. In them, the atomic model is an inseparable basic component, and a coupled model consists of multiple subcomponents (atomic model component and/or coupled model component) and may be formed in a hierarchical way [6]. Just like a unit system, every atomic model of the formalism has a time base, inputs, states, outputs, and functions for determining next states and outputs given current states and inputs [6]. In the next sub section, we build a DEVS model for routing path discovery process in DMSN. After that, we present the essentials of DEVS formalism.

DEVSJAVA [6] is a DEVS-based simulation environment that is pure JAVA, and component-based. It provides a platform for engineers and researcher to build DEVS models for different application and to simulate these models by as-fast-as or real-time mode.

Corresponding to the only two DEVS models (atomic model and coupled model); there are two basic types of DEVS simulation components in DEVSJAVA, atomic simulator and coordinator. They can be extended to different simulators such as real time decentralized simulator, distributed as-fast-as simulator and heap-based simulator.

B. An Autonomous Agents Simulation for the disjoint Mobile Sensor Networks (DMSN) Application

1) Basic Architecture of DMSN Agents

Figure 6 provides a pictorial view of the basic architecture of DMSN agents and how every type of agent works and is configured.

Every autonomous agent corresponds to a self-configurable, self-adaptable and self-maintainable mobile sensor. They autonomously adjust or acquire their behavior through checking the status of itself and the whole network and dynamically interacting with the environment by communicating with requests and responses and actuating events of interest.

In the architecture, there is one and only one DMSN coordinator agent who manages the registration/deregistration of cluster coordinator agents and is responsible for coupling management in DMSN and dynamic environment simulation management including the message input or output of the simulation environment. It aims at coordinating the cluster coordinator agents and the interaction between them. Every cluster is coordinated by a cluster coordinator agent who is responsible for member management, role management, action management and topology management. Namely, the cluster

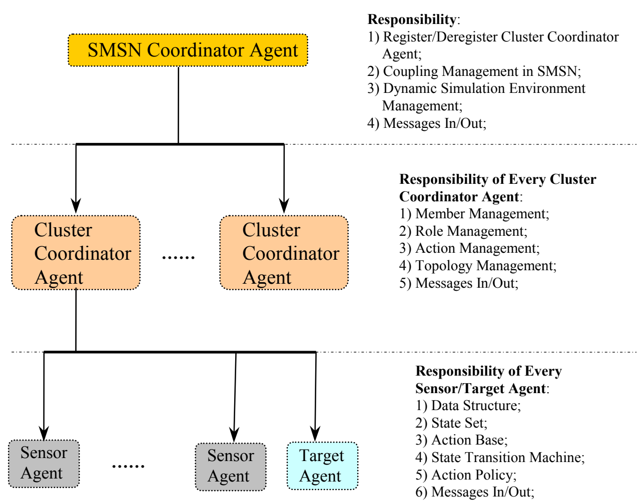


Figure 6. Basic architecture of DMSN agents

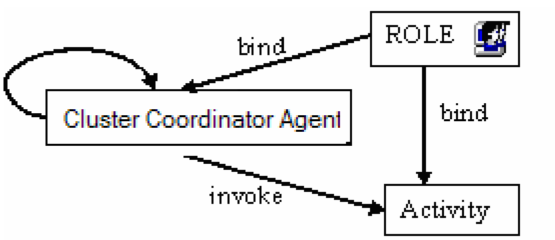


Figure 7. Relation among role, cluster agent and activity component.

coordinator agent aims at managing and coordinating the sensor agents, target agents and the interaction between them.

The coordinator agents provide register service to maintain the growth and shrink of the whole group (see Figure 6).

The cluster coordinator agent has role and action management (see Figure 7). These roles are assigned to different sensor agents. By the assignment of the role, sensor agents have the same interface but dynamically perform different functions. Furthermore, the role component provides a policy to restrain the whole status of DMSN and activities of sensor nodes.

2) *Modeling and Simulation of DMSN Agents*

Figure 8 provides a pictorial view of the framework of DMSN agents modeling and simulation. It presents the relationship between DEVS simulation, DEVS modeling and DMSN agents system. We now describe the detailed design.

Every sensor node in DMSN is a sensor agent and can be represented as an atomic model. Figure 9 shows the basic model of the i^{th} sensor. Due to the high dynamic connection and disconnection, we employ coordinator agent to keep the consistent and synchronized coupled record between multiple sensor nodes. The approach used by the coordinator agent is to add and remove coupling between models. Figure 5 shows the dynamic coupling in different sensor nodes at certain system running phase.

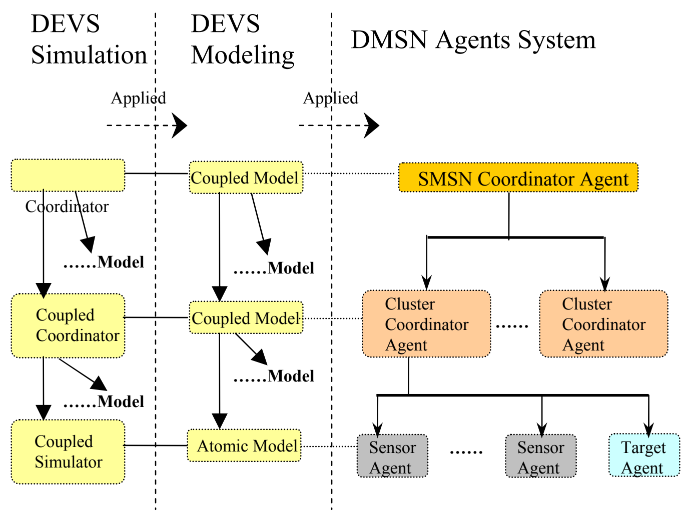


Figure 8. Framework for DMSN agents modeling & simulation

—— represents the reference relation.

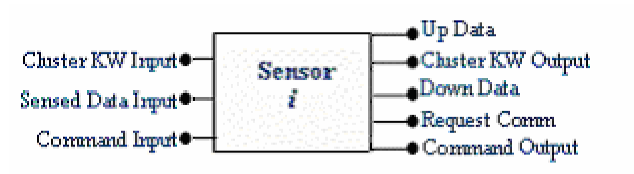


Figure 9. The atomic model for the i^{th} sensor

(1) Sensor atomic model

According to DEVS formalism [6], the sensor atomic model is defined as:

$SensorM=<X,S,Y,\delta_{int},\ \delta_{ext},\ \lambda,\ ta>$

where, X is the set of three input values {Cluster KW Input, Sensed Data Input, Command Input };

S is the set of thirteen statuses {power-on, power-off, stand-by, awaken, active, processing, detecting, selecting messenger, sending messenger, messenger returning, messaging, exchange data, time-step}.

Y is the set of five output values {Cluster KW Output, Request Command, Up Data, Down Data, Command Output}

δ_{int} : S→S is the internal transition function.

δ_{ext} : Q× X^b →S is the external transition function when there exist at least one input, where Q ∈ {(s, e) | s ∈ S, 0≤e≤ta(s)} is the total state set, X^b is the set of Xs, and e is the elapsed time since last transition.

λ : S→ Y^b is the output function.

ta: S→ $R^+_{0,\infty}$ is the time advance function.

Every sensor atomic model has eight ports for input and output (see Figure 9). According to sensor current status, the function of ports may be changed from for inputting to for outputting or from for outputting to for inputting. For example, when a regular sensor node is changed to a cluster head node (namely, change happens from regular sensor role to cluster head role), it can automatically change "Up Data", a output port of regular sensor role, to the input port for receiving data from regular sensor nodes. Additionally, roaming status is different from active status because the sensor in active status has been in a cluster and may be detecting and processing data, but one in roaming status just moves around searching for new targets or the available position in a cluster.

The state transition in sensor atomic model is as the following pseudo code (here we just list two state transitions, roaming and active):

```
PROCEDURE ROUTING (sensor)
1 IF (sensor is in roaming status)
2   THEN IF (targets have been sensed)
3     THEN IF (cluster exists around the targets)
4       THEN IF (cluster is full & cluster doesn't need more)
5         THEN Change direction
6         ELSE Enter in the cluster and possess a suitable
               position; Share the cluster Head content;
               Change status;
7       ELSE Become the head of the cluster
8   ELSE Continue to move with random angle
9 ELSE IF (sensor is in active status)
10    THEN IF (sensor is cluster head)
11      THEN Recording; receiving data; change status;
               processing, checking pipeline and
               waiting time; select messenger; send
               messenger;
12      ELSE Recording; sending to cluster head
```

VI. EXPERIMENTS & RESULTS

Experiments and evaluation are conducted to compare the energy consumption of the proposed two protocols: route discovery based on straight line moving of messengers and based on flexible sharing policy of messengers. These experiments are based on the agents modeling and simulation.

a) Simulation Environment

Ten disjoint targets were randomly located in the 800 m * 640 m simulation area in our preliminary experiments. One hundred sensors were initially roaming in it by the random way for detecting targets. Each sensor had a radio propagation range of 50m. All simulations were run with the logic simulation time of 100 milliseconds. Every cluster has the maximum members of eight sensors. The energy consumption units are initialized for every type of activities as follows:

SENSOR_FULL_ENERGY = 100000;
ENERGY_SENSOR_MOVING = 10;
ENERGY_SENSOR_TRANSMISSION = 1;
SENSOR_NO_ENERGY = 10;

The DEVS simulation platform was developed by University of Arizona [6]. Our model mentioned in the previous sections allows more sensors and targets and there is no constraint for transmission radium and cluster members.

We run the simulation on Windows XP system with Pentium ® 4 CPU 2.66 GHz processor. As shown in the GUI interface (see Figure 10), the base station is the icon in the left-bottom corner. Every cycle represents a formed cluster around a target. The straight line represents the moving path of messengers of different clusters. These experiments were run individually in the time steps of 600, 700, 800, 900, 1000, 1100, 1200, 1300, 1400 and 1500.

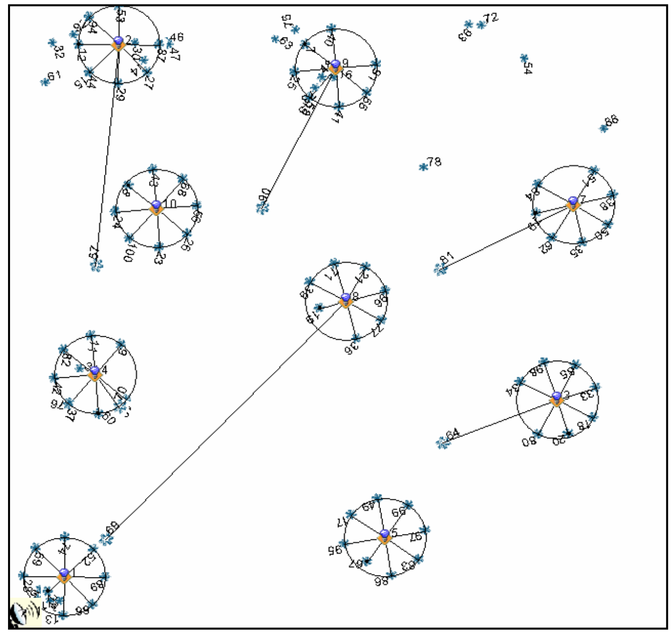


Figure 10. Simulation of route discovery and routing in DMSN

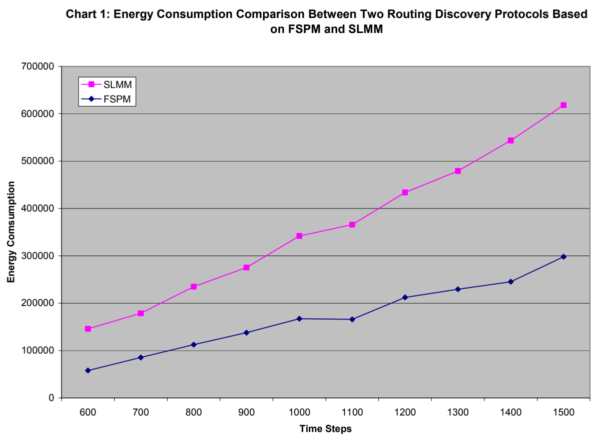
b) Simulation of Energy Consumption

The limited battery capacity greatly challenges the design and population of mobile sensor networks. It is a vital issue to save energy when determining the network route [10]. We denote energy consumption for the entire network as EC. Energy for a sensor is mainly consumed in moving and transmission. The moving activities include the roaming of sensor nodes and the messaging of messenger. The transmission energy consumption happens when regular sensor nodes in a cluster upload data or messages, messengers exchange data between each other, and cluster head appoints a sensor as the messenger.

A linear formula of energy consumption (EC) was defined according to the activities (See formula 1). We denoted as W_m the weight of a moving activity and as W_t the weight of a transmission activity. R_o is set as the steps of roaming activity. M_g is set as the steps of messaging activity. D_{up} is set as the amount of updated packets. D_{ex} is set as the amount of exchanged packets between messengers. D_{am} is set as the times of appointing messenger.

$$EC = W_m(R_o + M_g) + W_t(D_{up} + D_{ex} + D_{am}) \quad (1)$$

We employ the average value of all EC of repetitive experiments as the EC evaluation data because targets were randomly localized and sensors were moving in the random direction. These random nature influences the happening of messaging and packet exchanging. Experiment results are shown in the following Chart 1.



The experiment results demonstrate that FSPM route discovery protocol is consistently more energy efficient than SLMM. As the simulation time increases we can see that FSPM becomes more and more energy efficient.

VII. CONCLUSION

In this paper, we apply agent-based simulation to studying an autonomous messenger-based route discovery and routing protocol for disjoint clusters-based topology in DMSN. We present the proposed protocol along with the simulation model based on DEVS and preliminary results. This paper mainly focuses on studying the delocalization, route discovery and routing protocol in disjoint mobile sensor networks. In doing so, we present two messenger based route discovery protocols, one based on straight line moving of messengers (SLMM), and a novel protocol based on flexible sharing policy of messengers (FSPM). According to the dynamic properties of the application, we design the agents based model and simulation and implemented a prototype simulator. With varied repetitive experiments, we draw a conclusion that the flexible sharing policy route discovery and routing protocol (FSPM) provides more energy savings than the straight line moving one (SLMM). In turn, our protocol not only increases network life time but also enables sensor data and cluster knowledge sharing for co-operative efforts of the autonomous agents to perform autonomous monitoring of the terrain and communicating with the base station for prolonged duration in an energy-efficient manner. Our future work will focus on improving the efficiency of the flexible sharing policy protocol and the uncertainty problem in disjoint network and finishing the routing component. We plan to make a better comparison of current and further improved FSPM protocol. This work is one of the few to consider mobile messengers for routing in disjoint clusters within sensor networks. Routing among disjoint clusters within a sensor network remains an open problem. This work can be further enhanced to consider challenges in the crisis-driven mobile sensor networks. As this is a new research area there is much room for further study and improvement.

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. *"A Survey On Sensor Networks"* IEEE Commun. Mag. (ASSC02), 40(8):102–114, 2002.

[2] Elizabeth M. Royer, Chai-Keong Toh, *"A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks"*, IEEE Personal Communications, Vol. 6, No. 2, pp. 46-55, April 1999.

[3] Chandrakasan, and Hari Balakrishnan, *"Energy-Efficient Communication Protocols for Wireless Microsensor Networks"*, Proc. Hawaiian Int'l Conf. on Systems Science, January 2000.

[4] Wenrui Zhao, Mostafa Ammar, Ellen Zegura, *"A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks"* Proceedings of ACM Mobihoc 2004.

[5] Cui, X., Hardin, T., Ragade, R. K., and Elmaghraby, A. S. *"A Swarm-based Fuzzy Logic Control Mobile Sensor Network for Hazardous Contaminants Localization"*, IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004).

[6] Bernard P. Zeigler: *"Introduction to DEVS Modeling and Simulation with JAVA: Developing Component-Based Simulation Models. In DEVS-Java reference Guide"*, www.acims.arizona.edu

[7] Suman Banerjee, Samir Khuller *"A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks"* IEEE Infocom 2001, Anchorage, Alaska, April 2001.

[8] Sameer Tilak, Vinay Kolar, Nael B. Abu-Ghazaleh and Kyoung-Don Kang, *"Dynamic Localization Protocols for Mobile Sensor Networks"* IEEE IWSEEASN 2005.

[9] David B. Johnson and David A. Maltz. *"Dynamic Source Routing in Ad Hoc Wireless Networks"*, In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5, pages 153-181, Kluwer Academic Publishers, 1996.

[10] H. Liu, J. Lie, Y. Pan, and Y.-Q. Zhang, *"An adaptive genetic fuzzy multi-path routing protocol for wireless ad-hoc networks"*, Proc. of SAWN2005, May, 2005.

[11] Ossama Younis, Sonia Fahmi and Paolo Santi *"Robust Communications for Sensor Networks in Hostile Environments"* Proceedings of the IEEE International Workshop on Quality of Service (IWQoS), pp. 10.19, June 2004.

[12] Zeigler B. P., H. Praehofer, and T. G. Kim *"Theory of modeling and simulation. 2nd ed. New York",* Academic Press 2000.