

UBIQUITOUS SENSOR NETWORKS FOR PERVASIVE COMPUTING

by

USN Team¹

Technical Report

RTMM, Kyung Hee University

Nov. 2005

Approved by _____
Project Supervisor

Date _____ 20th Nov.

¹ Shu Lei, Wu Xiaoling, Xu Hui, Wang Jin, Yang Jie, Riaz Ahmed Shaikh, Jinsung Cho, Sungyoung Lee, and Young Koo Lee.

RTMM, KYUNG HEE

UNIVERSITY

ABSTRACT

**UBIQUITOUS SENSOR
NETWORKS FOR PERVASIVE
COMPUTING**

by USN Team

Project Supervisor: Professor Sungyoung Lee
Department of Computer Engineering

Wireless sensor networks which ubiquitously deployed in our daily life are able to provide diverse sensed information for users. These ubiquitous sensor networks are currently playing the key role to realize the new coming pervasive computing paradigm as the underlying infrastructure.

The nature constraints of ubiquitous sensor networks, such as limited energy resource, low computation and transmission capacity strongly motivate our current research. In order to build up an energy-efficient ubiquitous sensor networks, we explored different energy-efficient methods from the beginning development stage to the final working stage. Several successful research results are presented in this technical report.

Furthermore, we consider that several sensor networks which are physically locating in different places sometimes should be integrated into one virtual sensor networks to provide more meaningful and comprehensive services for users.

In this technical report, we present a novel approach to integrate ubiquitous sensor networks with all-IP based wired/.wireless networks which is named V – IP Bridge. A comprehensive discussion and comparison is also provided to claim that our V – IP Bridge can cover all the advantages of existing related research works.

Through our V – IP Bridge, we can easily integrate different sensor networks which are locating in different places into one virtual sensor networks.

TABLE OF CONTENTS

1.1 Introduction.....	6
1.2 Particle Swarm Optimization	8
1.3 The Proposed Algorithm	9
1.3.1 Optimization of Coverage	9
1.3.2 Optimization of Energy Consumption.....	11
1.4 Performance Evaluation.....	12
1.5 Conclusion and Future Work.....	15
2.1 Introduction.....	17
2.2 System Model	18
2.2.1 Network Model	18
2.2.2 Propagation Model.....	18
2.2.3 Energy Expenditure.....	20
2.3 Proposed Algorithm.....	21
2.3.1 Localized Energy Aware Broadcast Algorithm	21
2.3.2 Broadcast Tree Calculation.....	23
2.3.2 Examples Constructed by the Various Algorithms	25
2.4 Performance Evaluation.....	25
2.5 Conclusions.....	29
3.1 Introduction.....	31
3.2 Several Original Clustering Algorithms	31
3.2.1 Highest-Degree Algorithm.....	31
3.2.2 Lowest-ID Algorithm.....	32
3.2.3 Node-Weight Algorithm.....	32
3.2.4 Weighted Clustering Algorithm.....	33
3.3 An Improved Weighted Clustering Algorithm	33
3.3.1 Principles.....	34
3.3.2 Steps	35
3.3.3 Example	35
3.4 A Novel Genetic Annealing based Clustering Algorithm (GACA).....	38
3.4.1 Principles.....	39
3.4.2 Steps	39
3.4.3 Example	40
3.5 Simulation Result and Analysis	43
3.5.1 Analysis of Average Cluster Number.....	43

3.5.2 Analysis of Topology Stability	45
3.5.3 Analysis of Cluster head Load-balancing.....	46
3.5.4 Analysis of Network Lifetime.....	48
3.6 Conclusion and Future Work.....	48
4.1 Introduction.....	50
4.2 Related Work.....	50
4.3 ETRI-QM	51
4.4 ETRI-PF.....	53
4.4.1 Problem Formulation	54
4.4.2 Steps of ETRI-PF	56
4.5 Simulation Result	59
4.6 Conclusion and Future Work.....	62
5.1 Introduction.....	64
5.2 Related Work.....	65
5.3 Communication Paradigms of Sensor Networks	67
5.4 Major Design Principles	68
5.5 V – IP Bridge.....	69
5.5.1 Key Idea.....	69
5.5.2 System Model of Virtual – IP Bridge	69
5.5.3 Workflow of Both Translation Components.....	72
5.6 An Example: G – IP Approach.....	74
5.6.1 Active Data Discovery and Registration.....	74
5.6.2 Data Information & Grid ID & IP address Mapping.....	75
5.6.3 Packets Translation in Grid ID – IP Address Mapping Layer	76
5.7 Discussion.....	78
5.7.1 Comparison with Related Researches	78
5.7.2 Integration of Different Sensor Networks.....	79
5.7.3 Drawbacks of Virtual – IP Bridge.....	80
5.8 Conclusion and Future Work.....	81
6.1 Introduction.....	83
6.2 Related work.....	84
6.3 Visual sensor networks: architecture.....	86
6.3.1 Image acquisition.....	87
6.3.2 Sensor Collaboration	87
6.3.3 Inter-scene tracking.....	88
6.4 Object tracking by extracting individual contour	88

6.5 Individual contour extraction: experimental Results	90
6.6 Conclusions and future work	93
7.1 Introduction.....	95
7.2 Introduction.....	96
7.3 System Model	97
7.3.1 Physical Layer Model.....	97
7.3.2 MAC Layer Protocol Model.....	98
7.4 Impact of Practical Models on Packet Reception and Energy Consumption	99
7.4.1 Reception Probability Model.....	99
7.4.2 Expected Energy Consumption	100
7.5 Impact of Practical Models on Power Aware Broadcast Protocols.....	100
7.5.1 EER Case.....	101
7.5.2 HHE Case.....	102
7.6 Conclusions.....	104

ENERGY-EFFICIENT DEPLOYMENT OF MOBILE SENSOR
NETWORKS BY PSO

1.1 Introduction

Mobile sensor networks consist of sensor nodes that are deployed in a large area, collecting important information from the sensor field. Communication between the nodes is wireless. Since the nodes have very limited energy resources, the energy consuming operations such as data collection, transmission and reception must be kept at a minimum.

In most cases, a large number of wireless sensor devices can be deployed in hostile areas without human involved, e.g. by air-dropping from an aircraft for remote monitoring and surveillance purposes. Once the sensors are deployed on the ground, their data are transmitted back to the base station to provide the necessary situational information.

The deployment of mobile sensor nodes in the region of interest (ROI) where interesting events might happen and the corresponding detection mechanism is required is one of the key issues in this area. Before a sensor can provide useful data to the system, it must be deployed in a location that is contextually appropriate. Optimum placement of sensors results in the maximum possible utilization of the available sensors. The proper choice for sensor locations based on application requirements is difficult. The deployment of a static network is often either human monitored or random. Though many scenarios adopt random deployment for practical reasons such as deployment cost and time, random deployment may not provide a uniform sensor distribution over the ROI, which is considered to be a desirable distribution in mobile sensor networks. Uneven node topology may lead to a short system lifetime.

The limited energy storage and memory of the deployed sensors prevent them from relaying data directly to the base station. It is therefore necessary to form a cluster based topology, and the cluster heads (CHs) provide the transmission relay to base station such as a satellite. And the aircraft carrying the sensors has a limited payload, so it is impossible to randomly drop thousands of sensors over the ROI, hoping the communication connectivity would arise by chance; thus, the mission must be performed with a fixed maximum number of sensors. In addition, the airdrop deployment may introduce uncertainty in the final sensor

positions. These limitations motivate the establishment of a planning system that optimizes the sensor reorganization process after initial random airdrop deployment, which results in the maximum possible utilization of the available sensors.

There are lots of research work [1], [2], [3], [4], [12] related to the sensor nodes placement in network topology design. Most of them focused on optimizing the location of the sensors in order to maximize their collective coverage. However only a single objective was considered in most of the research papers, other considerations such as energy consumption minimization are also of vital practical importance in the choice of the network deployment. Self-deployment methods using mobile nodes [4], [9] have been proposed to enhance network coverage and to extend the system lifetime via configuration of uniformly distributed node topologies from random node distributions. In [4], the authors present the virtual force algorithm (VFA) as a new approach for sensor deployment to improve the sensor field coverage after an initial random placement of sensor nodes. The cluster head executes the VFA algorithm to find new locations for sensors to enhance the overall coverage. They also considered unavoidable uncertainty existing in the precomputed sensor node locations. This uncertainty-aware deployment algorithm provides high coverage with a minimum number of sensor nodes. However they assumed that global information regarding other nodes is available. In [1], the authors examined the optimization of wireless sensor network layouts using a multi-objective genetic algorithm (GA) in which two competing objectives are considered, total sensor coverage and the lifetime of the network. However the computation of this method is not inexpensive.

In this chapter, we attempt to solve the coverage problem while considering energy efficiency using particle swarm optimization (PSO) algorithm, which can lead to computational faster convergence than genetic algorithm used to solve the deployment optimization problem in [1]. During the coverage optimization process, sensor nodes move to form a uniformly distributed topology according to the execution of algorithm at the base station. To the best of our knowledge, this is the first time to solve deployment optimization problem by PSO algorithm.

In the next section, the PSO algorithm is introduced and compared with GA. Modeling of sensor network and the deployment algorithm is presented in section 3, followed by simulation results in section 4. Some concluding remarks and future work are provided in section 5.

1.2 Particle Swarm Optimization

PSO, originally proposed by Eberhart and Kennedy [5] in 1995, and inspired by social behavior of bird flocking, has come to be widely used as a problem solving method in engineering and computer science.

The individuals, called, particles, are flown through the multidimensional search space with each particle representing a possible solution to the multidimensional problem. All of particles have fitness values, which are evaluated by the fitness function to be optimized, and have velocities, which direct the flying of the particles. PSO is initialized with a group of random solutions and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" factors. The first one, called *pbest*, is the best fitness it has achieved so far and it is also stored in memory. Another "best" value obtained so far by any particle in the population, is a global best and called *gbest*. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called *lbest*. After each iteration, the *pbest* and *gbest* (or *lbest*) are updated if a more dominating solution is found by the particle and population, respectively.

The PSO formulae define each particle in the D-dimensional space as $X_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD})$ where i represents the particle number, and d is the dimension. The memory of the previous best position is represented as $P_i = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{iD})$, and a velocity along each dimension as $V_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD})$. The updating equation [6] is as follows,

$$v_{id} = \omega \times v_{id} + c_1 \times \text{rand}() \times (p_{id} - x_{id}) + c_2 \times \text{rand}() \times (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

where ω is the inertia weight, and c_1 and c_2 are acceleration coefficients.

The role of the inertia weight ω is considered to be crucial for the PSO's convergence. The inertia weight is employed to control the impact of the previous history of velocities on the current velocity of each particle. Thus, the parameter ω regulates the trade-off between global and local exploration ability of the swarm. A large inertia weight facilitates global exploration, while a small one tends to facilitate local exploration, i.e. fine-tuning the current search area. A suitable value for the inertia weight ω balances the global and local exploration ability and, consequently, reduces the number of iterations required to locate the

optimum solution. Generally, it is better to initially set the inertia to a large value, in order to make better global exploration of the search space, and gradually decrease it to get more refined solutions. Thus, a time-decreasing inertia weight value is used [7].

PSO shares many similarities with GA. Both algorithms start with a group of a randomly generated population, have fitness values to evaluate the population, update the population and search for the optimum with random techniques. However, PSO does not have genetic operators like crossover and mutation. Particles update themselves with the internal velocity. They also have memory, which is important to the algorithm [8].

Compared with GA, PSO is easy to implement, has few parameters to adjust, and requires only primitive mathematical operators, computationally inexpensive in terms of both memory requirements and speed while comprehensible. It usually results in faster convergence rates than GA. This feature suggests that PSO is a potential algorithm to optimize deployment in a sensor network.

1.3 The Proposed Algorithm

First of all, we present the model of mobile sensor network. We assume that each node knows its position in the problem space, all sensor members in a cluster are homogeneous and cluster heads are more powerful than sensor members. Communication coverage of each node is assumed to have a circular shape without any irregularity. The design variables are 2D coordinates of the sensor nodes, $\{(x_1, y_1), (x_2, y_2), \dots\}$. Sensor nodes are assumed to have certain mobility. Many research efforts into the sensor deployment problem in mobile sensor networks [4, 9] make this sensor mobility assumption reasonable.

1.3.1 Optimization of Coverage

We consider coverage as the first optimization objective. It is one of the measurement criteria of QOS of a sensor network.

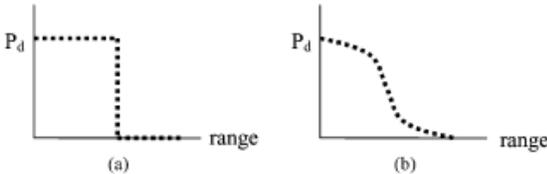


Figure 1.1: Sensor Coverage Models (a) Binary Sensor (b) Stochastic Sensor Models

The coverage of each sensor can be defined either by a binary sensor model or a probabilistic sensor model as shown in Fig. 1. In the binary sensor model, the

detection probability of the event of interest is 1 within the sensing range, otherwise, the probability is 0. Although the binary sensor model is simpler, it is not realistic as it assumes that sensor readings have no associated uncertainty. In reality, sensor detections are imprecise, hence the coverage needs to be expressed in probabilistic terms. In many cases, cheap sensors such as omnidirectional acoustic sensors or ultrasonic sensor are used. Some practical examples [4] include AWAIRS at UCLA/RSC Smart Dust at UC Berkeley, the USC-ISI network, the DARPA SensIT systems/networks, the ARL Advanced Sensor Program systems/networks, and the DARPA Emergent Surveillance Plexus (ESP). For omnidirectional acoustic sensors or ultrasonic sensors, a longer distance between the sensor and the target generally implies a greater loss in the signal strength or a lower signal-to-noise ratio. This suggests that we can build an abstract sensor model to express the uncertainty in sensor responses. In other words, a sensor node that is closer to a target is expected to have a higher detection probability about the target existence than the sensor node that is further away from the target.

In this chapter, the probabilistic sensor model given in Eq (3) is used, which is motivated in part by [11].

$$c_{ij}(x, y) = \begin{cases} 0, & \text{if } r + r_e \leq d_{ij}(x, y); \\ e^{-\lambda a^\beta}, & \text{if } r - r_e < d_{ij}(x, y) < r + r_e; \\ 1, & \text{if } r + r_e \geq d_{ij}(x, y). \end{cases} \quad (3)$$

The sensor field is represented by an $m \times n$ grid. An individual sensor node s on the sensor field is located at grid point (x, y) . Each sensor node has a detection range of r . For any grid point P at (i, j) , we denote the Euclidean distance between s at (x, y) and P at (i, j) as $d_{ij}(x, y)$, i.e., $d_{ij}(x, y) = \sqrt{(x-i)^2 + (y-j)^2}$. Eq (3) expresses the coverage $c_{ij}(x, y)$ of a grid point at (i, j) by sensor s at (x, y) , in which r_e ($r_e < r$) is a measure of the uncertainty in sensor detection, $a = d_{ij}(x, y) - (r - r_e)$, and λ and β are parameters that measure detection probability when a target is at distance greater than r_e but within a distance from the sensor. This model reflects the behavior of range sensing devices such as infrared and ultrasound sensors. The probabilistic sensor detection model is shown in Figure 1(b). The distances are measured in units of grid points. Figure 1(b) also illustrates the translation of a distance response from a sensor to the confidence level as a probability value about this sensor response. The coverage for the entire grid sensor field is calculated as the fraction of grid points that exceeds the threshold c_{thr} .

1.3.2 Optimization of Energy Consumption

After optimization of coverage, all the deployed sensor nodes move to their own positions. Now our goal is to minimize energy usage in a cluster based sensor network topology by finding the optimal cluster head positions. For this purpose, we assume a power consumption model [10] for the radio hardware energy dissipation where the transmitter dissipates energy to run the radio electronics and the power amplifier, and the receiver dissipates energy to run the radio electronics. This is one of the most widely used models in sensor network simulation analysis. For our approach, both the free space ($distance^2$ power loss) and the multi-path fading ($distance^4$ power loss) channel models were used. Assume that the sensor nodes inside a cluster have short distance dis to cluster head but each cluster head has long distance Dis to the base station. Thus for each sensor node inside a cluster, to transmit an l -bit message a distance dis to cluster head, the radio expends

$$E_{TS}(l, dis) = lE_{elec} + l\epsilon_{fs}dis^2 \quad (4)$$

For cluster head, however, to transmit an l -bit message a distance Dis to base station, the radio expends

$$E_{TH}(l, Dis) = lE_{elec} + l\epsilon_{mp}Dis^4 \quad (5)$$

In both cases, to receive the message, the radio expends:

$$E_R(l) = lE_{elec} \quad (6)$$

The electronics energy, E_{elec} , depends on factors such as the digital coding, modulation, filtering, and spreading of the signal, here we set as $E_{elec} = 50nJ/bit$, whereas the amplifier constant, is taken as $\epsilon_{fs} = 10pJ/bit/m^2$, $\epsilon_{mp} = 0.0013pJ/bit/m^2$.

So the energy loss of a sensor member in a cluster is

$$E_s(l, dis) = l(100 + 0.01dis^2) \quad (7)$$

The energy loss of a CH is

$$E_{CH}(l, Dis) = l(100 + 1.3 \times 10^{-6} \times Dis^4) \quad (8)$$

Since the energy consumption for computation is much less than that for communication, we neglect computation energy consumption here.

Assume m clusters with n_j sensor members in the j^{th} cluster C_j . The total energy loss E_{total} is the summation of the energy used by all sensor members and all the m cluster heads:

$$E_{total} = l \sum_{j=1}^m \sum_{i=1}^{n_j} \left(100 + 0.01 dis_{ij}^2 + \frac{100}{n_j} + \frac{1.3 \times 10^{-6} Dis_j^4}{n_j} \right) \quad (9)$$

Because only 2 terms are related to distance, we just set the fitness function as:

$$f = \sum_{j=1}^m \sum_{i=1}^{n_j} \left(0.01 dis_{ij}^2 + \frac{1.3 \times 10^{-6} Dis_j^4}{n_j} \right) \quad (10)$$

1.4 Performance Evaluation

The PSO starts with a “swarm” of sensors randomly generated. As shown in Fig. 3 is a randomly deployed sensor network with coverage value 0.31 calculated using approximate method mentioned in section 3.1. A linear decreasing inertia weight value from 0.95 to 0.4 is used, decided according to [6]. Acceleration coefficients c_1 and c_2 both are set to 2 as proposed in [6]. For optimizing coverage, we have used 20 particles, which are denoted by all sensor nodes coordinates, for our experiment in a 50×50 square sensor network, and the maximum number of generations we are running is 500. The maximum velocity of the particle is set to be 50. The other parameters of sensor models are set to be $r=5$, $r_e=3$, $\lambda=0.5$, $\beta=0.5$, $c_{th}=0.7$. The coverage is calculated as a fitness value in each generation.

After optimizing the coverage, all sensors move to their final locations in setup phase. Now the coordinates of potential cluster heads are set as particles in the sensor network. The communication range of each sensor node is 15 units with a

fixed remote base station at (25, 80). We start with a minimum number of clusters acceptable in the problem space to be 4. The node, which will become a cluster head, will not have any restriction on the transmission range. The nodes are organized into clusters by the base station. Each particle will have a fitness value, which will be evaluated by the fitness function (10) in each generation. Our purpose is to find the optimal location of cluster heads. Once the position of the cluster head is identified, if there is no node in that position then a potential cluster head nearest to the cluster head location will become a cluster head.

We also optimized the placement of cluster head in the 2-D space using GA. We used a simple GA algorithm with single-point crossover and selection based on a roulette-wheel process. The coordinates of the cluster head are the chromosomes in the population. For our experiment we are using 10 chromosomes in the population. The maximum number of generations allowed is 500. In each evolution we update the number of nodes included in the clusters. The criterion to find the best solution is that the total fitness value should be minimal.

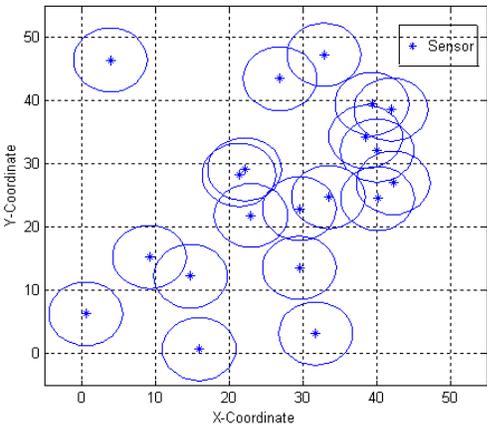


Figure 1.3 Randomly deployed sensor network with $r=5$ (Coverage value=0.31)

Fig. 4 shows the improvement of coverage during the execution of the PSO algorithm. Note that the upper bound for the coverage for the probabilistic sensor detection model (roughly 0.38) is lower than the upper bound for the case of binary sensor detection model (roughly 0.628). This due to the fact that the coverage for the binary sensor detection model is the fraction of the sensor field covered by the circles. For the probabilistic sensor detection model, even though there are a large number of grid points that are covered, the overall number of grid points with coverage probability greater than the required level is fewer.

Fig. 5 shows the convergence rate of PSO and GA. We ran the algorithm for both approaches several times and in every run PSO converges faster than GA, which was used in [1] for coverage and lifetime optimization. The main reason for the fast convergence of PSO is due to the velocity factor of the particle.

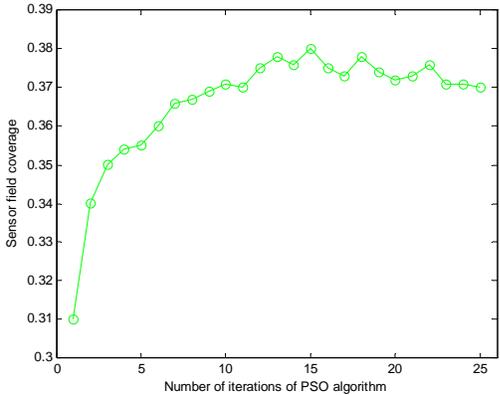


Figure 1.4 Optimal coverage achieved using PSO algorithm (probabilistic sensor detection model)

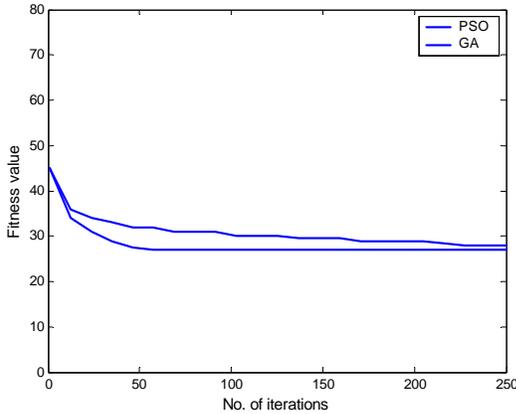


Figure 1.5 Comparison of convergence rate between PSO and GA based on Eq. (10)

Fig. 6 shows the final cluster topology in the sensor network space after coverage and energy consumption optimization when the number of clusters in the sensor space is 4. We can see from the figure that nodes are uniformly distributed among the clusters compared with the random deployment as shown in Fig 3. The four red stars denote cluster heads, the blue diamonds are sensor members, and the dashed circles are communication range of sensor nodes. The energy saved is the

difference between the initial fitness value and the final minimized fitness value. In this experiment, it is approximately 16.

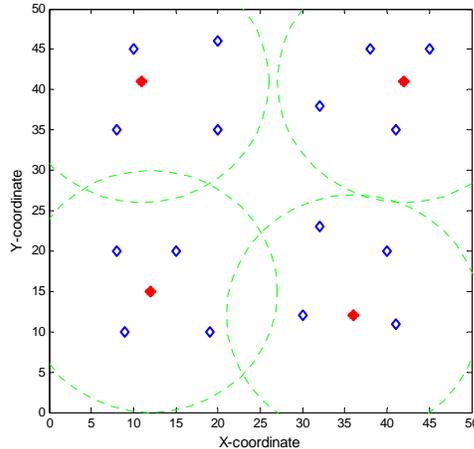


Figure 1.6 Energy efficient cluster formation using PSO

1.5 Conclusion and Future Work

The application of PSO algorithm to optimize the coverage in mobile sensor network deployment and energy consumption in cluster-based topology is discussed. We have used coverage as the first optimization objective to place the sensors uniformly based on a realistic probabilistic sensor model, and energy consumption as the second objective to find the optimal cluster head positions. The simulation results show that PSO algorithm has faster convergence rate than GA based layout optimization method while demonstrating good performance.

In the future work, we will take sensor movement energy consumption into account. Moreover, other objectives, such as time and distance for sensor moving will be further studied.

REFERENCES

- [1] Damien B. Jourdan, Olivier L. de Weck: Layout optimization for a wireless sensor network using a multi-objective genetic algorithm. IEEE (2004) 2466-2470
- [2] K. Chakrabarty, S. S. Iyengar, H. Qi and E. Cho: Grid coverage for surveillance and target location in distributed sensor networks. IEEE transactions on computers, vol.51, (2002) 1448-1453
- [3] A. Howard, M.J. Mataric and G. S. Sukhatme: Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area

- coverage problem. Proc. Int. Conf. on distributed Autonomous Robotic Systems, (2002) 299-308
- [4] Y. Zou and K. Chakrabarty: Sensor deployment and target localization based on virtual forces. Proc. IEEE Infocom Conference, vol. 2., (2003) 1293-1303
- [5] Kennedy and R. C. Eberhart: Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, (1995) 1942-1948
- [6] Yuhui Shi, Russell C. Eberhart: Empirical study of Particle Swarm Optimization. IEEE (1999) 1948-1950
- [7] K.E. Parsopoulos, M.N. Vrahatis. Particle Swarm Optimization Method in Multiobjective Problems. SAC, Madrid, Spain, ACM (2002)
- [8] <http://www.swarmintelligence.org/tutorials.php>
- [9] Nojeong Heo and Pramod K. Varshney: Energy-Efficient Deployment of Intelligent Mobile Sensor Networks. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems And Humans, Vol. 35, No. 1, January (2005)
- [10] Wendi B. Heinzelman, Anantha P. Chandrakasan, and Hari Balakrishnan: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless Communications, Vol. 1, No. 4, October (2002)
- [11] A. Elfes: Sonar-based real-world mapping and navigation. IEEE Journal of Robotics and Automation, Vol. RA-3, No. 3 (1987) 249–265
- [12] Archana Sekhar, B. S. Manoj and C. Siva Ram Murthy: Dynamic Coverage Maintenance Algorithms for Sensor Networks with Limited Mobility. Proc. PerCom (2005) 51-60

LOCALIZED ENERGY AWARE BROADCAST PROTOCOL FOR
WIRELESS NETWORKS WITH ANTENNAS

2.1 Introduction

In wireless networks which have limited resources such as sensor network, communication ranges are limited, thus many nodes must participate to the broadcast in order to have the whole network covered. The most important design criterion is energy and computation conservation, as nodes have limited resources. All the protocols that have been proposed for broadcast can be classified into two kinds of solutions: centralized and localized. Centralized solutions mean that each node should keep global network information and global topology. There exist several centralized energy-aware broadcast algorithms for the construction of broadcast trees with omni-directional antennas in the literature. In addition, the well-known energy-aware algorithm of Broadcast Incremental Power (BIP) [1] is “node-based” algorithm and exploits the “wireless broadcast advantage” property associated with omni-directional antennas, namely the capability for a node to reach several neighbors by using a transmission power level sufficient to reach the most distant one. Applying the incremental power philosophy to network with directional antennas, the Directional Broadcast Incremental Power (DBIP) algorithm [2] has very good performance in energy saving. The problem of centralized approach is that mobility of nodes or frequent changes in the node activity status (from “active” to “passive” and vice-versa) may cause global changes in topology which must be propagated throughout the network for any centralized solution. This may results in extreme and unacceptable communication overhead for networks. Hence, because of the limited resources of nodes, it is ideal that each node can decide on its own behavior based only on the information from nodes within a constant hop distance. Such distributed algorithms and protocols are called localized [3-7].

In this chapter, we propose and implement a localized energy-efficient broadcast protocol which is based on the “Incremental Power” philosophy for wireless networks with Directional Antenna, Localized Directional Broadcast Incremental Power Protocol (LDBIP). Our localized protocol only uses localized and distributed location information and computing to construct broadcast tree. The use of directional antennas can reduce the beam width angle to diffuse the radio transmission to one direction and thus provides energy savings and interference reduction. In our algorithm, source node sets up spanning tree with only position

information of its neighbors within certain hops. Directional antennas are used for transmitting broadcast packet, and the transmission power is adjusted for each transmission to the minimal necessary for reaching the particular neighbor. Relay node that receives broadcast packet will consider relay instructions included in received packet to compute its own localized spanning tree and do the same as source node. We compare the performance of our protocol (LDBIP) to those of BIP, DBIP and LBIP [8]. Experimental results show that in static wireless networks, this new protocol has better performance compared to BIP and LBIP, and similar performance to DBIP, and that in mobile wireless networks, LDBIP has better performance even compared to DBIP.

The remainder of the chapter is organized as follows: in Section 2.2, we introduce our system model including the impact of the use of directional antennas on energy consumption; Section 2.3 presents our localized energy-aware algorithm for broadcast tree construction, which exploits the properties of directional antennas; in Section 2.4, we compare the performance of our protocol (LDBIP) to those of BIP, DBIP and LBIP; in Section 2.5, we present our conclusions and future work on this research.

2.2 System Model

2.2.1 Network Model

We assume each node has a low-power Global Position System (GPS [9]) receiver, which provides the position information of the node itself. In every position based broadcast protocol, nodes need position information about neighborhood nodes. The method we used is as following: initially each node emits its position message containing its id, and when a node u receives this kind of special message from a node v , it adds v to its neighborhood table; in mobile network except initialization each node sets timer to check its position, and if mobility happens it will emit his position message again to let other nodes update neighborhood table.

2.2.2 Propagation Model

We use two kinds of propagation model, free space model [10] and two-ray ground reflection model [11]. The free space model considers ideal propagation condition that there is only one clear line-of-sight path between the transmitter and receiver, while the two-ray ground model takes reality into consideration and considers both the direct path and a ground reflection path.

The following equation to calculate the received signal power in free space at distance d from the transmitter

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}, \quad (1)$$

where P_t is the transmitted signal power. G_t and G_r are the antenna gains of the transmitter and the receiver respectively. L ($L \geq 1$) is the system loss, and λ is the wavelength.

The following equation to calculate the received signal power in Two-ray ground model at distance d

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L}, \quad (2)$$

where h_t and h_r are the heights of transmit and receive antennas respectively. However, the two-ray model does not give a good result for a short distance due to the oscillation caused by the constructive and destructive combination of the two rays, whereas, the free space model is still used when d is small. Therefore, a cross-over distance d_c is calculated. When $d < d_c$, Eqn. (1) is used. When $d > d_c$, Eqn.(2) is used. At the cross-over distance, Eqns. (1) and (2) give the same result.

So d_c can be calculated as

$$(4\pi h_t h_r) / \lambda. \quad (3)$$

When considering omni-directional antennas and uniform propagation conditions, it is common to select G_t and G_r as 1.

The use of directional antennas can permit energy savings and reduce interference by concentrating transmission energy where it is needed. We learn from [12] that because the amount of RF energy remains the same, but is distributed over less area, the apparent signal strength is higher. This apparent increase in signal strength is the antenna gain. We use an idealized model in which we assume that all of the transmitted energy is concentrated uniformly in a beam

of width θ , as shown in Fig. 1, then the gain of area covered by the beam can be calculated as

$$2 \sqrt{1 - \cos \frac{\theta}{2}}, \quad (4)$$

while the gain of the other areas is zero. As a consequence of the “wireless broadcast advantage” property of omni-directional systems [13], all nodes whose distance from Node i does not exceed r_{ij} will be able to receive the transmission with no further energy expenditure at Node i .

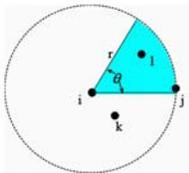


Figure 2.1 Use of Directional Antenna

While using directional antenna, the advantage property will be diminished, since only the nodes located within the transmitting node’s antenna beam can receive the signal. In Fig. 2.1, only j, l can receive the signal, while k cannot receive the signal.

We assume that the beam width θ is fixed beam width and one node can simultaneously support more than one directional antenna. Furthermore, we assume that each antenna beam can be pointed in any desired direction to provide connectivity to a subset of nodes that are within communication range. In addition, we use directional receiving antennas, which have a beneficial impact to avoid background noise and other user interferences.

2.2.3 Energy Expenditure

In addition to RF propagation, energy is also expended for transmission (encoding, modulation, etc.) and reception (demodulation, decoding, etc.). We define

- p^T = transmission processing power and
- p^R = reception processing power.

The total power expenditure of a node, when transmitting to a maximum range r over a sector of width θ , is

$$p = p^{RF}(r, \theta) + p^T + p^R \quad (5)$$

Where $p^{RF}(r, \theta)$ is RF propagation energy expenditure and the term p^R is not needed for the source node. A leaf node, since it does not transmit but only receives, has a total power expenditure of p^R .

2.3 Proposed Algorithm

2.3.1 Localized Energy Aware Broadcast Algorithm

The goal of the localized algorithm is to allow a localized and distributed computation of broadcast tree. We assume every node knows its local neighbors position information.

The principle is as follows: the source node S (the one that initiates the broadcast) computes the broadcast tree with its local neighborhood position information and sends the broadcast packet to each of its one hop neighbor, while includes N (integer, $N > 0$) hops computed relay information and the N th hop relay nodes id in broadcast packet. For each of other nodes, for example, node U who receives the packet for the first time, three cases can happen:

- The packet contains both relay instructions for U and U 's *id*. U will use these relay instructions to construct its own local broadcast tree. Then, instead of starting from an empty tree as S did, it extends the broadcasting tree based on what source S has calculated for it. By this way, the joint neighborhood nodes of S and U will use the same spanning tree.
- The packet contains only relay instructions for U . U will just follow these relay instructions to relay the packet.
- There are no relay instructions for U . In this case, node U does nothing.

After the procedure mentioned above, node U will rebroadcast the packet again to its own one hop neighbor and include N hops computed relay information for its own relay nodes and the N th hop relay nodes id, just like what source node

has done. The reason why we use N to refer relay nodes hop number is that the range within which each node manage positional information on other nodes can be changed according to requirement, and the optimal changes according to the application demands and the node's hardware performance.

In this principle, there may be some nodes which will receive this packet more than one time, then at this time, node can simple drop the packet and doesn't rebroadcast again. In order to reduce overlap, we use the neighbor nodes elimination scheme. Source node will include its local N hops neighbor nodes in packet, because these nodes certainly will receive the packet soon. Once the node which is in charge of recalculating local spanning tree receives the packet, except recording the relay information it should also record the nodes which will be covered soon. If the covered node is not used in relay information and also is a neighbor node of this node, then this node will delete it from its neighbor list and after deletion calculate its own broadcast tree. Fig.2.2 is the pseudo-code of the proposed algorithm.

```

0. Randomly select source node S
1. For source node S:
2. { /*****source node's locale calculation*****/
3.   Computes its local broadcast tree;
4.   Set up broadcast packet P;
5.   Include N hops relay instructions in packet P;
6.   Include N hops neighbors' ID in packet P;
7.   Include Nth hop relay instructions in packet P;
8.   Send packet P to each of its one hop neighbor using directional antenna;
9. }
10. For any node U (except S):
11. if (node U receives packet P){
12.   if ( the first time){
13.     Inspect packet P;
14.     if (there is relay instruction for U){
15.       if (U's id exists in Nth hop relay nodes' id){
16.         Search and record all relay instructions for U;
17.         /*****Neighbor Nodes Elimination Scheme*****/
18.         Check included covered nodes' ID;
19.         While ( (ID != U's address)&&( ID ∉ relay instruction info) )
20.           if (ID ⊂ U's local neighbors list)
21.             delete this node record from U's local neighbors list;
22.         /*****U's local calculation*****/
23.         Refer recorded relay instructions;
24.         Use U's modified local neighbors list;

```

```

25.         Computes U's local broadcast tree;
26.         Act as source node;
27.         }else if (U's id does not exist in Nth hop relay nodes' id)
28.             Only relay received packet as recorded relay instructions;
29.         }else if (there is no relay instruction for U)
30.             Do nothing;
31.         } else
32.             Simply drop packet P;
33.     }

```

Figure 2.2 Pseudo-code of the Proposed Algorithm

2.3.2 Broadcast Tree Calculation

As for how to set up broadcast tree, we have considered two basic approaches with directional antennas:

- Construct the tree by using an algorithm designed for omni-directional antennas; then reduce each antenna beam to our fixed beam width.
- Incorporate directional antenna properties into the tree-construction process.

The first approach can be based on any tree-construction algorithm. The “beam-reduction” phase is performed after the tree is constructed. The second approach which takes directional antenna into consideration at each step of the tree construction process can be used only with algorithms that construct trees by adding one node at a time. In this section, we describe the later approach applied in our algorithm *LDBIP* in detail.

The incremental power philosophy, originally developed for use with omni-directional antennas, can be applied to tree construction in networks with directional antennas as well. At each step of the tree-construction process, a single node is added, whereas variables involved in computing cost (and incremental cost) are not only transmitter power but beam width θ as well. In our simple system model, we use fixed beam width θ_f , that means for adding a new node, we can only have two choices: set up a new directional antenna to reach a new node; raise the length range of beam to check whether there is new node covered or not. A pseudo code of the broadcast tree calculation algorithm can be written as Fig. 2.3.

Input: given an undirected weighted graph $G(N,A)$, where N : set of nodes, A : set of edges

Initialization: set $T:=\{S\}$ where S is the source node. Set $P(i):=0$ for all $1 \leq i \leq |N|$ where $P(i)$ is the transmission power of node i .

Procedure:

while $|T| \neq |N|$

do find an edge $(i, j) \in T \times (N-T)$ with fixed beam width θ_f such that ΔP_{ij} is minimum; if an edge $(i,k) \in T \times T$ raising the length range of beam can cover a node $j \in (N-T)$, then incremental power $\Delta P_{ij} = d_{ij}^{\alpha} \frac{\theta_f}{2\pi} - P(i)$; otherwise, $\Delta P_{ij} = d_{ij}^{\alpha} \frac{\theta_f}{2\pi}$.

add node j to T , i.e., $T := T \cup \{j\}$.

set $P(i) := P(i) + \Delta P_{ij}$.

Figure 2.3 Pseudo code of broadcast tree calculation algorithm

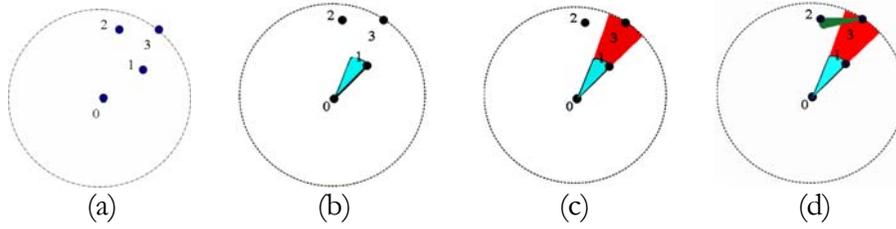


Figure 2.4 Nodes Addition in LDBIP

Fig.2.4(a) shows a simple example in which the source node has 4 local neighbor nodes 0, 1, 2, and 3. Node 1 is the closest to 0, so it is added first; in Fig.2.4(b), an antenna with beam width of θ_f is centered between 0 and Node 1. Then we must decide which node to add next (Node 2 or Node 3), and which node (that is already in the tree) should be its parent. In this example, the beam from 0 to Node 1 can be extended to include both Node 1 and Node 3, without setting up a new beam. Compared to other choices that setting up a new beam from Node 0 to Node 2, or from Node 1 to Node 2, this method has minimum incremental power. Therefore, Node 3 is added next by increasing the communication range of Node 0 and Node 1. In Fig.2.4(c), finally, Node 2 must be added to the tree. Three possibilities are respectively to set up a new beam from Node 0, Node 1 or Node 3. Here we assume that Node 3 has minimum distance. Then in Fig.2.4 (d) we set up a new beam from Node 3 to Node 2.

2.3.2 Examples Constructed by the Various Algorithms

Fig.2.5 shows the broadcast tree produced by BIP, DBIP, LBIP and LDBIP for a 12-node network, where the source node is shown larger than the other nodes. These broadcast trees are generated in our simulation work, which use the system model mentioned in Section 2.2.

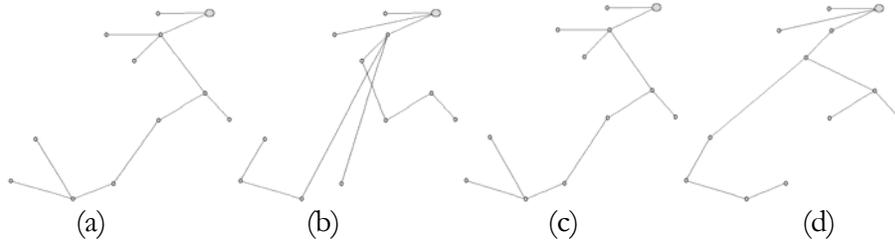


Figure 2.5 Broadcast Tree. (a) BIP (b) DBIP ($\theta_f = 30$) (c) LBIP (d) LDBIP ($\theta_f = 30$)

Because DBIP and LDBIP use directional antenna, therefore in our simulation system, according to different θ_f , we can get different broadcast tree; of course, the according energy consumption will also be different. Furthermore, because our algorithm LDBIP is distributed, which means every node only calculates its two hops neighborhood broadcast tree, the Fig.2.5(c) in fact is the combination of all nodes' broadcast tree. Based on our algorithm, the joint parts of nodes' broadcast tree will not have too much difference because nodes refer relay information from other nodes and apply the neighbor nodes elimination scheme.

2.4 Performance Evaluation

In this section, we present our performance evaluation for our localized algorithm *LDBIP*, and also compare it with two centralized algorithm *BIP* and *DBIP* which are very effective centralized protocols in energy consumption and with another localized algorithm *LBIP*. Especially for *LBIP* and *LDBIP*, we choose the hop number N as 2. We use ns2 as our simulation tool and assume AT&T's Wave LAN PCMCIA card as wireless node model which parameters are listed in table 2.1. As for system model, we apply the network, propagation, and energy model mentioned in Section 2.2.

Table 2.1 Parameters for Wireless Node Model

	AT&T's Wave LAN PCMCIA card
frequency	914MHZ
maximum transmission range	40m
maximum transmit power	8.5872e-4 W
receiving power	0.395 watts

transmitting power	0.660 watts
omni-antenna gain of receiver/transmitter	1db
fixed beam width of directional antennas	30
directional antenna receiver/transmitter gain	58.6955db
MAC protocol	802.11
propagation model	free space / two ray ground

The wireless network is always composed of 100 nodes randomly placed in a square area which size is changed to obtain different network density D defined as the average number of neighbors per each node. The formula can be written as:

$$D = N * \frac{\pi r^2}{A^2}, \quad (6)$$

where A represents the edge length of deployment square area, and r is the maximum transmission range. From Eqn. (6), we can get calculate A by

$$A = r \sqrt{\frac{N \pi}{D}}. \quad (7)$$

For each measure, 50 broadcasts are launched and for each broadcast, a new network is generated.

RAR (Reach Ability Ratio) is the percentage of nodes in the network that received the message. Ideally, each broadcast can guarantee 100% RAR value. While in sparse network since the maximum transmission range of nodes is not big enough to guarantee the network connectivity, RAR may be less than 100%. To compare the different protocols, we observe the total power consumption over the network when a broadcast has occurred. We compute a ratio named EER, that represents the energy consumption of the considered protocol compared to the energy that would have been spent by a Blind Flooding (each node retransmits once with maximum transmission range). The value of EER is so defined by:

$$EER = \frac{E_{protocol}}{E_{flooding}} \times 100. \quad (8)$$

We also observe SRB (Saved Rebroadcast) which is the percentage of nodes in the network that received the message but did not relay it. A Blind Flooding has a SRB of 0%, since each node has to retransmit once the message.

Our simulation work is based on two steps: first we test the performance of our protocol in static wireless ad hoc network, and then we take mobile network into consideration. To compare the performance with those of other protocols, we observe the total power consumption over the network. In mobile simulation environment, the energy consumption includes not only the energy consumption for broadcasting message, but also that for propagation for mobility.

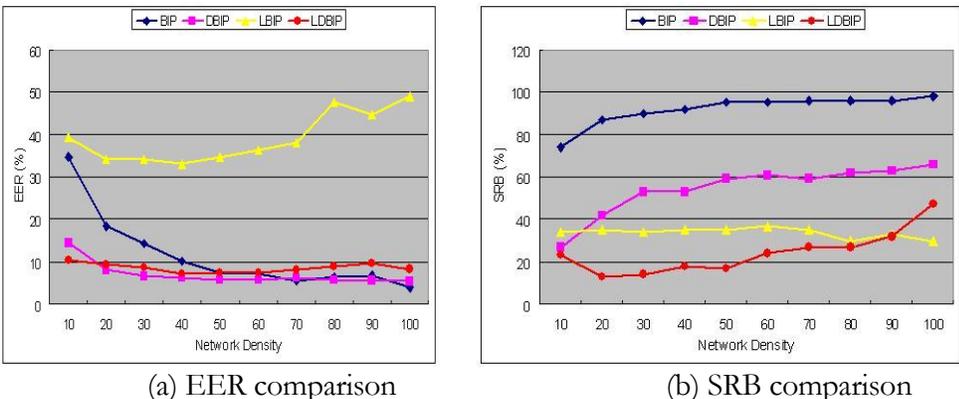


Figure.2.6. Performance Comparison in Static Wireless Network

Fig.2.6 shows EER and SRB comparison for *BIP*, *DBIP*, *LBIP* and *LDBIP* protocols in static wireless networks with different network density. As for the RAR value, since we choose the network density which can guarantee the network connectivity, so all the RAR results are 100%. From Fig.2.6 (a) we can find that all the four protocols have much better energy conservation than flooding. Because of employing directional antenna, *DBIP* and *LDBIP* have much less energy consumption compared to *BIP* which uses omni-directional antenna in low network density and similar saving energy performance in high network density. Also benefiting from directional antenna, compared to another localized algorithm *LBIP*, our proposal *LDBIP* has much better performance in energy conservation. In addition, the energy conservation performance of *DBIP* and *LDBIP* is stable despite of network density. Compared to centralized algorithm *DBIP*, our localized algorithm *LDBIP* has a little more energy consumption. That is because our algorithm employs the topology of only local neighbors whereas *DBIP* utilizes the total network topology to calculate energy efficient broadcast tree. From Fig.2.6 (b) we can observe localized protocols have less SRB compared to centralized protocols, since localized protocols only calculate local

broadcasting tree which cause unnecessary relay instructions compared to centralized protocols. In addition, using omni-directional antenna can save more retransmission, since “wireless broadcast advantage” will be decreased by employing directional antenna.

Now we take mobility into consideration. In our simulation we use mobile scenarios to simulate the nodes’ mobility in mobile networks. These mobile scenarios are randomly generated by special tool of ns2, “setdest [14]”. As we mentioned in section 2.2 positioning, in mobile network except initialization each node should set timer to check whether this node has moved or not. If mobility occurs, node will use its maximum transmission radius to emit its new location information to let other nodes update their neighborhood table. In centralized solution, this information must be propagated throughout the network, In order to compare between different protocols, we use the same mobile scenario in certain network density.

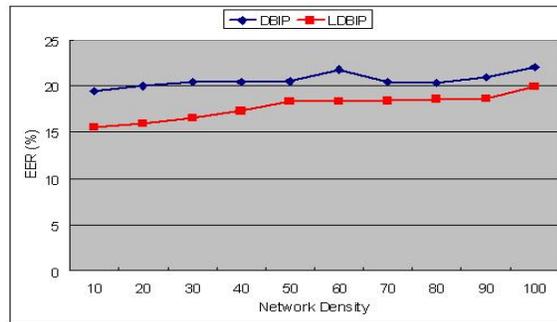


Figure 2.7 EER Comparison in Mobile Network

Fig.2.7 shows EER comparison for *DBIP* and *LDBIP* protocols in mobile networks with different network density. Compared to centralized algorithm *DBIP* in mobile network, our localized algorithm *LDBIP* has better energy saving performance. That is because in centralized solution, e.g. *DBIP*, mobility of nodes need to be broadcasted throughout the network, while in our localized algorithm *LDBIP*, mobility will be only propagated to that nodes’ neighborhood. Therefore *LDBIP* can get better performance. From this, we can infer that as mobility increases in mobile scenarios, *LDBIP* can get much better performance in energy conservation. In addition, as for SRB comparison in mobile network, there is little difference with that in static network.

In summary, our localized protocol *LDBIP* can only use localized location information and distributed computation to complete broadcasting task. Our simulation work verifies that in mobile networks, our localized energy-aware protocol has very good performance in energy conservation.

2.5 Conclusions

In this chapter, we proposed the new localized energy-aware broadcast protocol for wireless networks with directional antennas which have limited energy and computation resources. Our algorithm is based on the localized information and distributed computation method, which means, rather than source node collects all location information of network to calculate broadcast tree, every node collects some part of the whole network's nodes location information and participates calculating broadcast tree. At the cost of a few more information stored in the broadcast packets, our localized algorithm offers better energy saving result than well-known centralized algorithm DBIP in mobile environment. Especially, if mobility of nodes increases in network, our distributed algorithm can get lesser energy consumption and better performance than centralized solution.

In future work, we plan to take realistic facts into consideration for energy consumption and network lifetime.

REFERENCES

- [1]. J. E. Wieselthier, G. D. Nguyen, A. Ephremides: On the construction of energy-efficient broadcast and multicast trees in wireless networks. Proc. IEEE INFOCOM (2000) 585-594
- [2]. J.E. Wieselthier, G.D. Nguyen, A. Ephremides: Energy-Limited Wireless Networking with Directional Antennas: The Case of Session-Based Multicasting. Proc. IEEE INFOCOM (2002) 190-199
- [3]. P. Bose, P. Morin, I. Stojmenovic, J. Urrutia: Routing with guarantee delivery in ad hoc networks. ACM/Kluwer Wireless Networks (2001) 609-616
- [4]. T. Chu, I. Nikolaidis: Energy efficient broadcast in mobile ad hoc networks. In Proc. Ad-Hoc Networks and Wireless (ADHOC-NOW), Toronto, Canada (2002) 177-190
- [5]. W. Peng, X. Lu: On the reduction of broadcast redundancy in mobile ad hoc networks. In Proc. Annual Workshop on Mobile and Ad Hoc Networking and Computing (MobiHoc'2000), Boston, Massachusetts, USA (2000) 129-130
- [6]. A. Qayyum, L. Viennot, A. Laouiti: Multipoint relaying for flooding broadcast messages in mobile wireless networks. In Proc. 35th Annual Hawaii International Conference on System Sciences (HICSS-35), Hawaii, USA (2002)
- [7]. J. Wu, H. Li: A dominating-set-based routing scheme in ad hoc wireless networks. In Proc. 3rd Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm (DIALM'99), Seattle, USA (1999) 7-14

- [8]. F.Ingelrest, D.Simplot-Ryl: Localized Broadcast Incremental Power Protocol for Wireless Ad Hoc Networks. 10th IEEE Symposium on Computers and Communications (ISCC 2005), Cartagena, Spain (2005)
- [9]. E.D. Kaplan: Understanding GPS: Principles and Applications. Artech House (1996)
- [10]. H.T. Friis: A note on a simple transmission formula. Proc. of the IRE, Vol. 41, May (1946) 254-256
- [11]. H.T. Friis: Introduction to radio and radio antennas. IEEE Spectrum, April (1971) 55-61
- [12]. Joseph J. Carr: Directional or Omni-directional Antenna. Joe Carr's Receiving Antenna Handbook, Hightext (1993)
- [13]. J.E. Wieselthier, G.D. Nguyen, A. Ephremides: Algorithms for Energy-Efficient Multicasting in Static Ad Hoc Wireless Networks. Mobile Networks and Applications (MONET), vol. 6, no. 3 (2001) 251-263
- [14]. Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns/>.

CLUSTER HEAD LEVEL OPTIMIZATION

3.1 Introduction

Up to now, there are many routing protocols based on various strategies in Mobile Ad hoc NETWORK (MANET), and they can be classified into several kinds as follows: (1) proactive and reactive; (2) flat and hierarchical; (3) GPS assisted and non-GPS assisted. These kinds of protocols can be used solely or together. Here we mainly discuss the hierarchical routing protocols, which are based on the clustering algorithm [1, 2]. Since they have both the C/S structure of centrality and scalability nature of distribution, it is very likely that they will become the first choice of routing selection in the future.

Several original clustering algorithms have been proposed in MANET. These include: (1) Highest-Degree Algorithm; (2) Lowest-ID Algorithm; (3) Node-weight Algorithm; (4) Weighted Clustering Algorithm; (5) Others, like RCC (Random Competition based Clustering), LCC (Least Cluster Change), LEACH etc.

In the following section, we will first introduce several original clustering algorithms in section 3.2. Based on which, an improved weighted clustering algorithm is proposed in section 3.3. In section 3.4, another novel Genetic Annealing based Clustering Algorithm (GACA) is also presented. Some simulation results and comparison is given in section 3.5. Finally, the conclusion is drawn in section 3.6.

3.2 Several Original Clustering Algorithms

Similar to the cellular network, the MANET can be divided into several clusters. Each cluster is composed of one cluster head and many normal nodes, and all the cluster heads form an entire dominating set. The cluster head is in charge of collecting information (signaling, message, etc.) and allocating resources within its cluster. And the normal nodes communicate with each other through their cluster head as is shown in Fig. 3.1, no matter they are in the same cluster or not.

3.2.1 Highest-Degree Algorithm

The Highest-Degree Algorithm, which is also known as connectivity-based clustering algorithm, was originally proposed by Gerla and Parekh [3, 4] in which

the degree of a node is computed based on its distance from others. A node x is considered to be a neighbor of another node y if x lies within the transmission range of y . The node with maximum number of neighbors (i.e., maximum degree) is chosen as a cluster head.

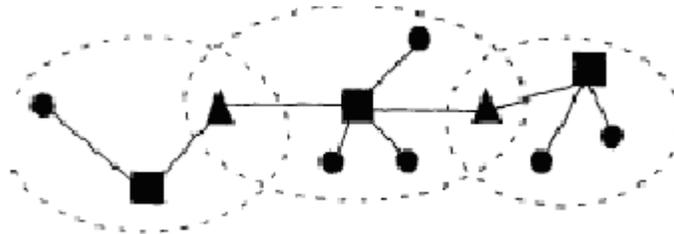


Figure 3.1: The hierarchical clustering architecture in MANET

Experiments demonstrate that the system has a low rate of cluster head change, but the throughput is low under the Highest-Degree Algorithm. As the number of nodes in a cluster increases, the throughput drops and hence a gradual degradation in the system performance is caused. All these drawbacks occur because this approach does not have any restriction on the upper bound of node degree in a cluster.

3.2.2 Lowest-ID Algorithm

This Lowest-ID Algorithm was originally proposed by Baker and Ephremides [5, 6]. It assigns a unique id to each node and chooses the node with the minimum id as a cluster head.

As for this algorithm, the system performance is better compared with the Highest-Degree Algorithm in terms of throughput. But it does not attempt to balance the load uniformly across all the nodes.

3.2.3 Node-Weight Algorithm

Basagni et al. [7, 8] proposed two algorithms, namely distributed clustering algorithm (DCA) and distributed mobility adaptive clustering algorithm (DMAC). In this two approaches, each node is assigned a weight based on its suitability of being a cluster head. A node is chosen to be a cluster head if its weight is higher than any of its neighbor's weight; otherwise, it joins a neighboring cluster head.

Results proved that the number of updates required is smaller than the Highest-Degree and Lowest-ID Algorithms. Since node weights were varied in each simulation cycle, computing the clusterheads becomes very expensive and there

are no optimizations on the system parameters such as throughput and power control.

3.2.4 Weighted Clustering Algorithm

The Weighted Clustering Algorithm (WCA) was originally proposed by M. Chatterjee et al [9, 10]. It takes four factors into consideration and makes the selection of cluster head and maintenance of cluster more reasonable. As is shown in equation (1), the four factors are node degree difference, distance summation to all its neighboring nodes, velocity and remaining battery power respectively. And their corresponding weights are w_1 to w_4 . Besides, it converts the clustering problem into an optimization problem and an objective function is formulated.

$$W_i = w_1\Delta_i + w_2D_i + w_3V_i + w_4E_i \quad (1)$$

However, only those nodes whose neighbor number is less than a fixed threshold value (a representation of capacity of a node) can be selected as a cluster head in WCA. It is not very desirable in the practical application. For example, many well-connected nodes whose neighbor number is larger than the fixed threshold might be a good candidate as well. Besides, its energy model is too simple. It treats the cluster head and the normal nodes equally and its remaining power is a linear function of time, which is also not very desirable. So, we will propose an improved clustering algorithm in the next section.

3.3 An Improved Weighted Clustering Algorithm

From the discussion mentioned above, we can see that most clustering algorithm, except for the WCA, only take one of the following factors into consideration, such as the node degree, ID, speed (velocity) or remaining battery power. When the problem in one aspect is solved, some other problems are introduced simultaneously. Inspired by the basic idea of WCA, we proposed an improved clustering algorithm. It considers the factors from four aspects, and makes the selection of cluster head and maintenance of cluster more reasonable. Besides, it converts the problem of cluster head selection into a problem of optimization as follows:

$$W_i = w_1N_i + w_2V_i + w_3D_i + w_4E_i$$

here, the w_1, w_2, w_3, w_4 are the weights of four aspects respectively, and $\sum_{i=1}^4 w_i = 1$.

N_i, V_i, D_i, E_i are node degree, velocity, distance summation to all its neighboring nodes and remaining battery power of node i respectively. They are all normalized, so $W_i \in (0,1)$. By solving the optimization problem of $\min (W_i)$, the cluster heads and their affiliated normal nodes are selected and a tradeoff is made from four aspects.

3.3.1 Principles

In order to determine the fitness degree of a node as a cluster head, we need to consider from the following aspects.

If the node degree is higher, then the node is more stable as a cluster head. Here we make a simple conversion $N_i = |d_i - M|$, where d_i is the practical degree of node i and M is the maximum degree. The smaller N_i is, the better node i will be as a cluster head. As for those nodes whose practical degree is larger than the maximum degree M , we also treat them as cluster head candidates. Once they are chosen as cluster heads, we will choose M nodes with less distance and velocity as their normal nodes. It is a distinctive difference between the WCA and our improved algorithm, and it can work very well under densely deployed ad hoc networks where the WCA becomes useless.

If the node velocity V_i is lower, then the node is more stable as a cluster head too.

If the distance summation of node i to all its neighbors D_i is smaller, it will consume less transmission power to communication with the normal nodes within its cluster.

If the remaining battery power E_i is higher, the longer it will be for node i to serve as a cluster head. Here we make another conversion and set a energy-consuming model. All the E_i s are set to zero initiatorily. If the node serves as a cluster head, we assume that it consumes 0.1 unit of energy and if normal node, 0.02 unit of energy. Once some E_i is above 1 (normalized), we believe that this node is out of energy and the network will become useless rapidly due to the avalanche effect. The energy-consuming relationship of 5:1 is commonly used among many papers. And it meets with the minimization problem very well.

3.3.2 Steps

Taking node i as an example, we compute its W_i according to the following steps and then judge whether it is a cluster head or a normal node.

Step 1: Compute its practical degree and then derive the $N_i = |d_i - M|$.

Step 2: Compute the distance summation D_i to its neighboring nodes.

Step 3: Set the velocity V_i according to the random way-point mobility model.

Step 4: Initiatorily set E_i to zero and increase their values according to the energy-consuming model. Our algorithm terminates once some E_i is above 1 (normalized).

Step 5: Compute W_i according to various w_i under different application.

Step 6: Taking the node with minimum W_i as the first cluster head and its neighboring nodes as its normal nodes within the same cluster. Then we go on with this process until all the nodes act as either cluster heads or normal nodes.

Step 7: All the nodes move randomly after one unit time (1s) and it goes back to step 1 again. And it terminates until a maximum number of time is reached or some node is out of energy.

It is worth noting that, as for those nodes whose degree is larger than the maximum degree M , we still chose them as cluster heads and select M neighboring nodes with less W_i s if they have a minimum value of W_i .

3.3.3 Example

To help the readers understand it better, we will give a simple example here.

In a network environment ($100 \times 100 \text{ m}^2$), there are N (10) nodes randomly deployed. Each node has a transmission range of 40m. A random way-point mobility model is adopted. The threshold value M is equal to 6 and the maximum velocity is 5m/s.

Then, we will do the clustering algorithm as following:

Step 1: As for N nodes, compute their practical degree d_i and converted normalized degree N_i , as are shown in table 3.1 and 3.2;

Table 3.1. The distribution of N (10) nodes

ID	1	2	3	4	5	6	7	8	9	10
X	37	68	25	76	47	32	31	45	78	2
Y	31	54	22	89	78	63	54	47	9	56

Table 3.2 The practical degree d_i and N_i

ID	1	2	3	4	5	6	7	8	9	10
d_i	5	6	3	2	5	6	5	6	0	2
N'_i	1	0	3	4	1	0	1	0	6	4
N_i	0.05	0	0.15	0.2	0.05	0	0.05	0	0.3	0.2

Step 2: Compute the normalized distance summation D_i according to table 3.1 and 3.3, and then we get table 3.4;

Table 3.3. Neighboring information

	1	2	3	4	5	6	7	8	9	10
1		√	√			√	√	√		
2	√			√	√	√	√	√		
3	√						√	√		
4		√			√					
5		√		√		√	√	√		
6	√	√			√		√	√		√
7	√	√	√		√	√		√		√
8	√	√	√		√	√	√			
9										

10						$\sqrt{\quad}$	$\sqrt{\quad}$			
----	--	--	--	--	--	----------------	----------------	--	--	--

Table 3.4 Normalized distance summation

ID	1	2	3	4	5	6	7	8	9	10
D_i	0.11	0.18	0.07	0.06	0.12	0.13	0.15	0.12	0	0.05

Step 3: Randomly setting the velocities of N nodes between $[0, V_{max}]$ as in table 3.5;

Table 3.5 Normalized velocity

ID	1	2	3	4	5	6	7	8	9	10
V_i	0.17	0.08	0.07	0.08	0.09	0.16	0.13	0.08	0.05	0.10

Step 4: Setting the remaining energy of all nodes as 0, and add 0.1 to the cluster head and 0.02 to the normal nodes every time unit.

Table 3.6 Normalized remaining energy

ID	1	2	3	4	5	6	7	8	9	10
E_i	0	0	0	0	0	0	0	0	0	0

Step 5: Let $w_1=0.5;w_2=0.2;w_3=0.1;w_4=0.2$, and compute each W_i as in table 3.7;

Table 3.7 Normalized objective value

ID	1	2	3	4	5	6	7	8	9	10
W_i	0.06	0.04	0.10	0.12	0.06	0.04	0.07	0.03	0.15	0.12

Step 6: From node 1, we choose the cluster heads and their neighbors according to W_i . Here, we get node 8 with the $\min(W_i)$ as the first cluster head and its neighbor 1、2、3、5、6、7 as the ordinary nodes in the cluster. Then, we choose other cluster head and its member from the left nodes. And go on. Finally,

we can derive all the cluster head set $\{8,4,9,10\}$ and their corresponding neighbors. And the first set of E_i is derived in table 3.8.

Table 3.8 Normalized remaining energy after initialization

ID	1	2	3	4	5	6	7	8	9	10
E_i	0.02	0.02	0.02	0.1	0.02	0.02	0.02	0.1	0.1	0.1

Step 7: After each time unit, every node moves according to the random way-point model. Re-compute the values of N_i , V_i , D_i and E_i . Once any E_i is larger than 1, our clustering algorithm terminates due to the effect of avalanche.

Finally, after 13 time units, the remaining energy of node 9 comes to 1.06 and we believe that the whole network becomes useless. And the distribution of all node's E_i is shown in table 3.9.

Table 3.9 The final remaining energy distribution

ID	1	2	3	4	5	6	7	8	9	10
E_i	0.34	0.5	0.34	0.9	0.42	0.5	0.34	0.58	1.06	0.5

From table 3.9, we can derive that each node plays a role as cluster head in turn. As for those nodes whose E_i s are 0.34, they serve 1 time as cluster head and 12 times as ordinary nodes ($1 \times 0.1 + 12 \times 0.02 = 0.34$). As for 0.5, 3 times as cluster head and 10 times as ordinary nodes ($3 \times 0.1 + 10 \times 0.02 = 0.5$). And so on.

3.4 A Novel Genetic Annealing based Clustering Algorithm (GACA)

As we know, the selection of cluster heads set, which is also called dominant set in Graphic Theory, is a NP-hard problem. Therefore, it is very difficult to find a global optimum. So, we can take a further step to use the computational intelligence methods, such as Genetic Algorithm (GA) or Simulated Annealing (SA), to optimize the objective function.

The principles of GA and SA are beyond the scope of our research in this chapter. So, we will directly come to the principles and steps of our proposed GACA in the following.

3.4.1 Principles

In most of the clustering models, the authors usually perform the clustering algorithm according to the node ID sequence, which is arranged from 1 to N. There's much randomness in this process, since different arrangement may cause different cluster head set. Inspired by the idea of GA, we make L sets of arrangement of [1, N] and operate in parallel to improve the speed.

Secondly, during the Roulette Wheel Selection process, we not only save the best solution and pass it to the next generation, but also make an improvement in the judgment equation.

To further reduce the randomness in the NP-hard problem, we make a crossover operation by exchange the locations of various pairs of nodes. In this case, the ID of each node is still unique, and this rule is not violated. More importantly, another new solution set is derived and it is the basis of SA.

By using SA, the speed of convergence can be greatly improved.

In the demonstration of one example, the authors can understand our clustering algorithm more easily.

3.4.2 Steps

The steps of our GACA are as follows. And it usually takes 5 to 10 iterations to convergence. So, we can say that it converges very fast.

Step 1: As for N nodes, randomly generate L integer arrangements in the range of [1, N].

Step 2: According to these random arrangements and the clustering principle of WCA, derive L sets of cluster heads and compute their corresponding $\sum w_{old}$.

Step 3: According to the Roulette Wheel Selection and Elitism in GA, select L sets of cluster heads which are better, and replace the original ones.

Step 4: As for each of the L sets of cluster heads, perform the crossover operator and derive the new L sets of cluster heads and their $\sum w_{new}$.

Step 5: According to the Metropolis “accept or reject” criteria in SA, decide whether to take the one from L sets of cluster heads in $\sum w_{iold}$ or in $\sum w_{inew}$. And the new L sets of cluster heads in the next generation are obtained.

Step 6: Repeat Step 3 to 5 until it converges or a certain number of iteration is reached. And in our simulation, it usually takes 5 to 10 iterations to converge. Then the global optimal or sub-optimal solution $\min(\sum w_{inew})$ ($i=1, 2, \dots, L$) is obtained and their corresponding set of cluster heads is known.

3.4.3 Example

In the same network environment, we will do our Genetic Annealing based Clustering Algorithm (GACA) as following:

Step 1: Randomly generate L(10) sets of [1,N] arrangements, shown in table 3.10;

Table 3.10 L(10) sets of [1,N] arrangements, here N=20

12	2	7	14	1	6	13	15	...	18	4	19	16	3	10	5	9	8
5	8	2	13	12	19	16	3	...	6	15	10	9	1	7	20	4	18
...
4	19	13	11	5	6	2	8	...	12	18	7	10	14	16	1	3	20
19	7	6	8	3	13	12	4	...	2	11	18	14	17	20	16	5	10

Step 2: Compute the corresponding W_i for each of the 20 nodes as follows, and then do the clustering algorithm according to our improved weighted clustering algorithm;

$$w=[0.0365,0.0494,0.0427,0.0358,0.0401,0.0250,0.0656,0.0776,0.0496,0.0308,0.0475, 0.0571,0.0658,0.0376,0.0648,0.0639,0.0750,0.0765,0.0514,0.0072]$$

Table 3.11 L(10) sets of Cluster heads

2	3	5	9	15	
5	4	9	17	6	

8	3	19	5	1	
15	10	3	1		
11	8	19	1	4	
4	19	11	9		
5	8	2	19	3	9
12	7	6	15		
14	12	8	9		
19	7	12			

By adding the W_i s of each cluster head, we can get the $\sum w_{iold}$ as follows:

$$\sum w_{iold} = [0.2467, 0.2255, 0.2483, 0.1748, 0.2489, 0.1844, 0.3108, 0.2125, 0.2219, 0.1741]$$

Step 3: According to the roulette wheel selection criteria $P_i = \frac{e^{-\sum w_i}}{\sum_{i=1}^L (e^{-\sum w_i})}$, we will

choose the best L (10) candidates from $\sum w_{iold}$, and save the best one without selection.

The final P_i and $\sum w_{iold}$ are :

$$P_i = [0.0747, 0.1669, 0.2403, 0.3935, 0.4665, 0.6057, 0.6450, 0.7501, 0.8457, 1.0000]$$

$$\sum w_{iold} = [0.2125, 0.2125, 0.1748, 0.2467, 0.3108, 0.1741, 0.1844, 0.3108, 0.1844, 0.1741]$$

Step 4: Perform one pair of crossover on the nodes in Table 3.10, and generate 10 set of $\sum w_{inew}$;

Table 3.12 Node arrangement after crossover

12	2	7	19	1	6	13	15	...	18	4	14	16	3	10	5	9	8
5	7	2	13	12	19	16	3	...	6	15	10	9	1	8	20	4	18

...
4	19	3	11	5	6	2	8	...	12	18	7	10	14	16	1	13	20
19	7	6	8	3	13	20	4	...	2	11	18	14	17	12	16	5	10

$$\sum w_{inew} = [0.2125, 0.1816, 0.1748, 0.2467, 0.3108, 0.1741, 0.1844, 0.3108, 0.1844, 0.1741]$$

Step 5: According to the Metropolis “accept or reject” criteria in SA, decide whether to take the one from L sets of cluster heads in $\sum w_{iold}$ or in $\sum w_{inew}$.

If $\sum w_{inew} \leq \sum w_{iold}$, then we accept $\sum w_{inew}$ directly and take it as one of the L members in the next generation. If $\sum w_{inew} > \sum w_{iold}$, we do not reject it directly,

but accept it with some probability. In other words, if $e^{-\frac{\sum w_{inew} - \sum w_{iold}}{\alpha T}}$ is larger than a randomly generated number in the range of (0,1), which shows that $\sum w_{inew}$ and $\sum w_{iold}$ may be very close to each other, we will still take it. Or else, we will reject the one in $\sum w_{inew}$ and take its counterpart in $\sum w_{iold}$. If we still take the one in $\sum w_{iold}$, we can conclude that the crossover operation in this generation has not yielded a better optimal solution candidate or the GACA converges. Besides, we make $T = \alpha T$ (α is a constant between 0 and 1 and we normally take 0.9) after each iteration, so that $\sum w_{inew}$ and $\sum w_{iold}$ must be closer if $\sum w_{inew}$ is to be accepted. In this way, our GACA will not be trapped in the local optima and the premature effect can be avoided. In other words, the diversity of searching space can be ensured and it is similar to the mutation operator in GA.

And finally, we get:

$$\sum w_i = [0.2125, 0.1816, 0.1748, 0.2467, 0.3108, 0.1741, 0.1844, 0.3108, 0.1844, 0.1741]$$

here, since each $\sum w_{inew}$ is less than $\sum w_{iold}$, so $\sum w_i$ is equal to $\sum w_{inew}$.

Step 6: Repeat Step 3 to 5 until it converges or a certain number of iteration is reached. And in our simulation, it usually takes 5 to 10 iterations to converge.

Then the global optimal or sub-optimal solution $\min (\sum_{i=1, 2 \dots L} w_{i_{new}})$ is obtained and their corresponding set of cluster heads is known.

In the end, the final cluster head set $\{19, 7, 12\}$ is achieved, and the corresponding $\min(\sum w_i)$ is equal to 0.1741.

From this result, we can see that the average cluster is minimum. And it ensures a lower response time or latency. The topology is the stablest and the network lifetime can be enlarged.

3.5 Simulation Result and Analysis

We set our simulation environment as follows. There are N nodes randomly placed within a range of 100 by 100 m², whose transmission range varies from 15m to 50m. A Random Waypoint mobility model is adopted here. And our GACA parameters are listed in table 3.13.

Table 3.13 GACA parameters

M	L	α	ε
1	10	0.9	0.01

3.5.1 Analysis of Average Cluster Number

As is shown in figure 3.2, we simulate N nodes whose transmission range varies from 15m to 50m.

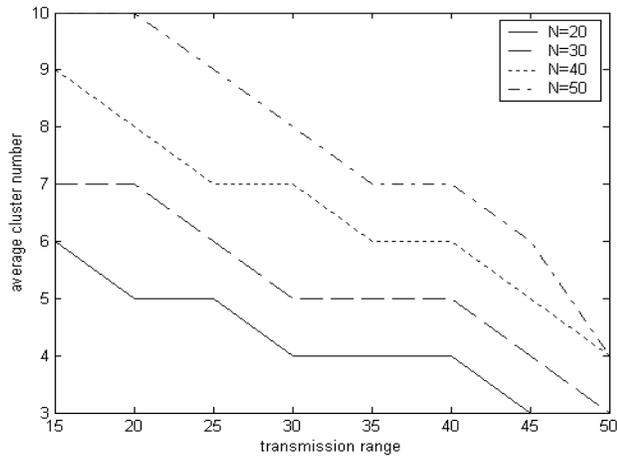


Figure 3.2: Average cluster number under various transmission range

We can conclude that:

- (1) The average number of cluster decreases as the transmission range increases.
- (2) As for a smaller transmission range, the average number of cluster differs greatly for various N. But when the transmission range is about 50m, one node can almost cover the entire network. So it only takes 3 to 5 clusters to cover all the N nodes.

Besides, we do the same research under various velocities.

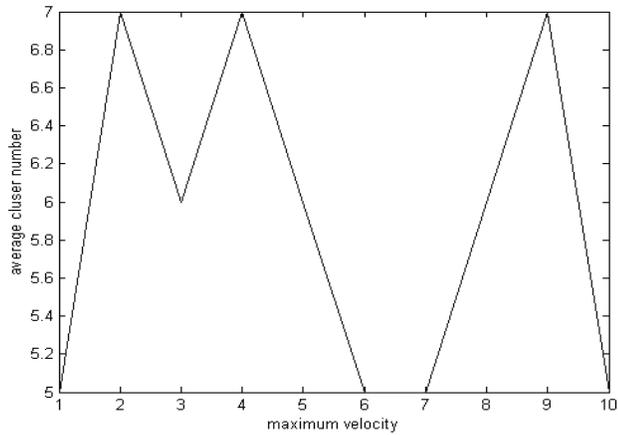


Figure 3.3: Average cluster number under various maximum velocities

Taking $N=R=30$ as an example, we can draw the conclusion from figure 3.3 that: the average number of cluster varies randomly between 5 and 7, and it is not related with the velocity.

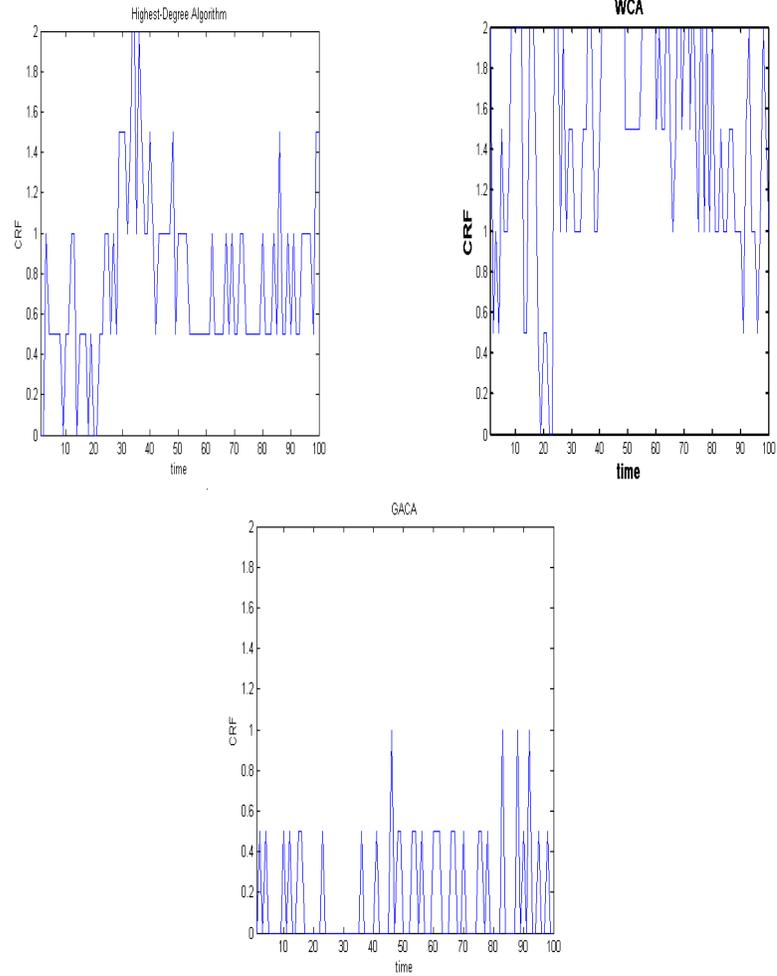
In fact, it matches with the practical situation too. For example, when one node with large velocity moves out of a cluster, it is highly possible that some other node gets into the same cluster. Or some of the nodes might move toward the same direction, which results in a relatively slow velocity and a stable cluster too.

3.5.2 Analysis of Topology Stability

As is mentioned before, the cluster head and their affiliated normal nodes may change their roles as they move. Here, we define a cluster reaffiliation factor (CRF) as follows:

$$\text{CRF} = \frac{1}{2} \sum_i |N_{i1} - N_{i2}| \quad (2)$$

here, i is the average number of cluster, and N_{i1}, N_{i2} are the degree of node i at different times. For example, we assume that cluster head 1 and 2 have 6 and 5 neighbors at first, i.e. $N_{11} = 6, N_{21} = 5$. As they move after one unit time, their neighbors (degrees) become 5 and 6, i.e. $N_{12} = 5, N_{22} = 6$. We can derive that CRF is equal to 1. So, we believe equivalently that one node in cluster 1 moves into cluster 2 and one reaffiliation is made.



(a) Highest-Degree Algorithm (b) WCA (c) GACA

Figure 3.4: CRF under various clustering algorithms

Under the maximum velocity of 10 m/s, we compared the CRF performance of Highest-Degree Algorithm, WCA and our GACA. From figure 3.4, we can see that GACA has the lowest CRF, which shows that it is the stablest clustering strategy among three of them. And WCA has the highest CRF value. The average CRF values of them are 1.56, 0.77 and 0.17 respectively.

Besides, we did some other experiments about CRF. We got the conclusion that the CRF increases as the velocity increases.

3.5.3 Analysis of Cluster head Load-balancing

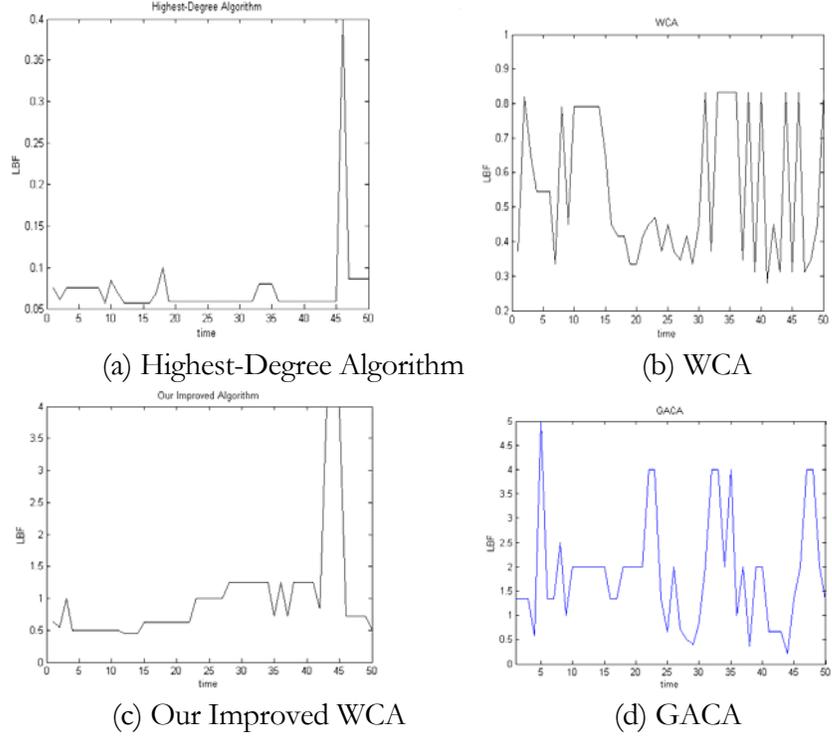


Figure 3.5: LBF under various clustering algorithms

We take the same definition of load-balancing factor (LBF) as is defined in [9]:

$$\text{LBF} = \frac{n_c}{\sum_i (x_i - \mu)^2}, \quad \mu = \frac{(N - n_c)}{n_c}$$

where, n_c is the average number of cluster, N is the number of all nodes, and x_i is the practical degree of node i . The larger LBF is, the better the load is balanced among the network. Taking $N=20$, $M=4$ as an example. The ideal case is that there are 4 clusters and each cluster head has a degree of 4, i.e. $n_c = x_i = 4$. Then, $\mu = \frac{(20-4)}{4} = 4$. So LBF is infinite, which shows that the load is perfectly balanced.

For simplicity, we do not consider the factor of network lifetime here (we will discuss it later in next section). So we set the simulation parameters as follows. $(X, Y) = [100, 100]$, $N=20$, $R=30$, $M=4$, maximum velocity $V_{\max} = 5$ and

$w_1 = 0.7, w_2 = 0.2, w_3 = 0.1, w_4 = 0$. It should be noted that we make N_i as our primary focus of attention ($w_1 = 0.7$), because it represents the matching degree of the practical case and ideal case directly. Figure 3.5 shows the LBF distribution under Highest-Degree Algorithm, WCA, our improved weighted clustering algorithm and GACA. From figure 3.5 we can see that: the Highest-Degree Algorithm has the worst performance, WCA is secondary to it, and our two improved clustering algorithms are better. Besides, the WCA will become useless under densely deployed ad hoc networks while our algorithm still works well. And their average values are 0.09, 0.38, 1.19 and 1.86 respectively.

3.5.4 Analysis of Network Lifetime

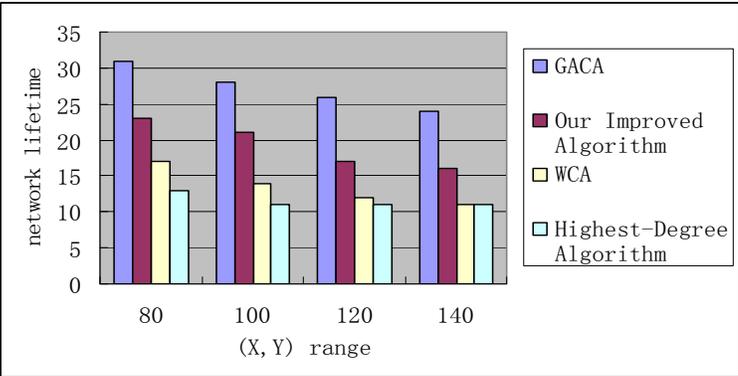


Figure 3.6: LBF under various clustering algorithms

The factor of node lifetime or network lifetime is a very important performance parameter in almost all the networks, such as cellular network, 4G network, ad hoc wireless network, sensor network, etc. A lot of work has been done in this aspect. For example, we can adaptively adjust the transmission power according to the source-destination distance or periodically turn off the radio and codec circuits to prolong node lifetime and the network lifetime.

Finally, we made a comparison between the aforementioned four clustering algorithms in the aspect of network lifetime, as is shown in figure 3.6. From which, we can see that GACA achieves the best performance, our improved weighted clustering algorithm is second to it, the WCA and the Highest-Degree algorithm are worst.

3.6 Conclusion and Future Work

In this chapter, we first introduced some related work about clustering algorithms. Based on which, we proposed an improved weighted clustering algorithm and a

novel Genetic Annealing based Clustering Algorithm (GACA). A lot of simulation results are provided and a comparison is made between our two clustering algorithms and the original ones, which shows that our algorithms have a better system performance on average.

There is still a lot of work to do. For example, the energy-consuming model is simple in this chapter. We can rebuild it by considering the practical traffic in the application layer. And we can save more energy by considering its status as active, idle or sleeping. Besides, we can prolong the network lifetime by adopting the cross-layer optimization methods.

REFERENCES

- [1] C.C. Chiang. Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel [C]. Proceedings of IEEE SICON'97, April 1997, pp.197-211.
- [2] Mingliang Jiang, Jinyang Li, Y.C. Tay. Cluster Based Routing Protocol [EB/OL]. August, 1999 IETF Draft.
- [3] A.K. Parekh. Selecting routers in ad-hoc wireless networks [C]. Proceedings of the SBT/IEEE International Telecommunications Symposium, August 1994.
- [4] M. Gerla and J.T.C. Tsai. Multicenter, mobile, multimedia radio network [J]. ACM/Baltzer Wireless Networks, 1(3),1995, pp. 255-265.
- [5] D.J. Baker and A. Ephremides. A distributed algorithm for organizing mobile radio telecommunication networks [C]. Proceedings of the 2nd International Conference on Distributed Computer Systems, April, 1981, pp. 476–483.
- [6] D.J. Baker and A. Ephremides. The architectural organization of a mobile radio network via a distributed algorithm [J]. IEEE Transactions on Communications COM-29 11(1981) pp. 1694–1701.
- [7] S. Basagni. Distributed clustering for ad hoc networks [C]. Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks, June 1999, pp. 310–315.
- [8] S. Basagni. Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks [C]. Proceedings of Vehicular Technology Conference, VTC, Vol. 2, 1999-Fall, pp. 889–893.
- [9] M. Chatterjee, S.K. Das and D. Turgut. An On-Demand Weighted Clustering Algorithm (WCA) for Ad hoc Networks [C]. Proceedings of IEEE GLOBECOM 2000, San Francisco, November 2000, pp.1697-1701.
- [10] Mainak Chatterjee, Sajal K. Das, and Damla Turgut A Weight-Based Distributed Clustering Algorithm for Mobile Ad hoc Networks [C]. Proceedings of Seventh International Conference on High Performance Computing (HiPC), December 2000, pp. 511-521.

ETRI-QM: REWARD ORIENTED QUERY MODEL FOR WIRELESS SENSOR NETWORKS

4.1 Introduction

Once a wireless sensor network is established, it will be able to provide important information to the end users. As one unifying view is to treat the sensor networks as distributed databases, the simplest mechanism to obtain information from this kind of database is to use queries for data within the network. However, most of the devices consisted in wireless sensor networks are battery operated, which highly constrains their life-span, and it is often not possible to replace the power source of thousands of sensors. Thus, how to efficiently query with the limited energy resources on the nodes is a key challenge in these unattended networks.

In the face of this challenge, we present a novel query model (ETRI-QM) to query the data with more important information among the interested data. By using this query model, we can dynamically combine these four constraints (Energy, Time, Reward, and Interest) to provide diverse query versions for different applications. Within our query model, each packet has four parameters: (1) energy consumption of the packet; (2) processing time of the packet; (3) important level of the packet; and (4) interest level of the packet. By using this ETRI-QM, we can achieve the following contributions: (1) Using *interest constraint* as the threshold to filter the uninterested incoming packets to reduce the energy consumption; (2) Using *reward constraint* to choose the high quality information and minimize the queried packet number to minimize the energy consumption but still satisfy the minimum information requirement.

4.2 Related Work

In [1], the authors present a sensor information networking architecture called SINA, which facilitates querying, monitoring, and tasking of sensor networks. To support querying within sensor networks, they design a data structure kept inside the sensor nodes based on the spreadsheet paradigm. In the spreadsheet paradigm, each sensor node maintains a logical datasheet containing a set of cells. By defining the semantic of a cell to specifying scope of the query, the information can be organized and accessed according to specific application needs, and also the number of the packets need to be sent can be reduced, thus the energy consumption will be reduced. However, there exist a tradeoff between

the energy cost to run SINA on each sensor node and the energy reduced by using SINA.

In [2], our work also has some similarities to techniques proposed, the authors introduced a new real-time communication architecture (RAP) and also a new packet scheduling policy called velocity monotonic scheduling (VMS). VMS assigns the priority of a packet based on its requested velocity. This work differs from our work in two aspects: one is that the cost-model is different in the two scenarios—in RAP is primarily reducing the end-to-end deadline miss ratio while we are minimizing energy consumption and maximizing the querying quality; the second one is that RAP intends to maximize the number of packets meeting their end-to-end deadlines without considering their value (reward, importance level), and in our model, we take reward an important constraint to deal with the queries.

Samuel et al discussed the design of an acquisitional query processor (ACQP) for data collection in sensor network in [3]. They provide a query processor-like interface to sensor networks and use acquisitional techniques to reduce power consumption. Their query languages for ACQP focus on issues related to when and how often samples are acquired. To choose a query plan that will yield the lowest overall power consumption, the query is divided into three steps: creation of query, dissemination of query and execution of query. Optimizations are made at each step.

Our ETRI-QM combines four constraints (**energy, time, interest** and **reward**) to maximize the querying quality with minimum energy consumption. In [4, 5]. Cosmin Rusu, *et al.* first time consider **Energy, Time,** and **Reward** these three constraints simultaneously while Reward denotes the important level of tasks. They believe that among a set of tasks of real time applications, some of them are more valuable than the others. So instead of processing several unimportant tasks just consuming less energy, it is more meaningful to process one valuable task consuming more energy. In our query model, we use **reward** to denote the importance level of data, so that we can transmit the data with more valuable information first. By considering the four constraints simultaneously, we make out our target that is to query the most valuable (**reward**) packets from the **interested** area to be transmitted while meeting **time** and **energy** constraints.

4.3 ETRI-QM

Applications may submit queries or register for events through a set of query/event service APIs. The APIs provides a high-level abstraction to applications by hiding the specific location and status of each individual node.

These APIs allow applications to specify the timing constraints as well as other constraints of queries.

ETRI-QM provides the following query/event service APIs.

Query {attribute_list, interested_area, system_value, timing_constraints, querier_loc}

Issue a query for a list of attributes in an interested area with the maximum system value (reward). Attributes refer to the data collected by different types of sensors, such as temperature sensors, humidity sensors, wind sensors, rain sensors etc. Interested area specifies the scope of the query, the area from which data is needed by the users. System value is defined as the sum of selected packets' reward. Timing_constraints can be period, deadline and so on. If a period is specified for a command, query results will be sent from the interested area to the issuer of query periodically. The querier_loc is the location of the base station that sends out the query.

Imagine a heterogeneous network consisting of many different types of sensors: temperature sensors, humidity sensors, wind sensors, rain sensors etc. monitoring the chemical found in the vicinity of a volcano. Suppose the volcano has just broken out, and we want to know which five chemicals found have the highest particle concentration. Obviously, sensors near the volcano will have more valuable data, which means that the importance levels of these data are much higher than those of the data collected by the further sensors. Thus, here we can consider reward to be the distance between sensors and the volcano.

Consider another example: lots of sensors are deployed in some area with different densities. For the EventFound case, take noise into account, the data collected by sensors having higher densities will be more reliable. So, here the reward is changed to be the density of the sensors around the interested area in the network.

There is one more example to make you clearly understand the concept of "reward". In the case of real-time communication for wireless sensor network, meeting the end-to-end deadline seems to be the most importance issue, so that we can consider arriving time to be the reward value. Reward is defined to be the importance level of the data collected by sensors. In the case of different wireless sensor networks, it can be specified to various formats.

4.4 ETRI-PF

After receiving the query message, the sensor nodes will start to collect related data and then send the packets to the cluster head. The cluster can be formed using LEACH or other techniques. In terms of the cluster head, many unprocessed packets are still physically existing in different sensor nodes and waiting for the processing of cluster head. Therefore, in sensor network, except the cluster head, all the other sensor nodes which are going to send packets to the cluster head can logically be considered as a buffer, since all of these packets are waiting for the processing of cluster head. We regard this buffer as the **First Tier Buffer (FTB)**. Actually the FTB is a logical concept for cluster head. The **Second Tier Buffer (STB)** is the buffer that physically exists inside cluster head. Since many sensor nodes will send packets to cluster head, obviously, cluster head needs buffer to store these received packets. Therefore, we propose the Two Tiers Buffer model for wireless sensor network as the figure 4.1 shows.

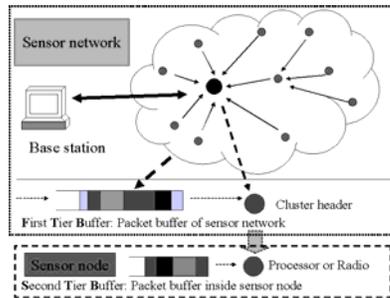


Figure 4.1: Two Tiers Buffer

In FTB, what we want is to **Maximize reward value to meet the Reward constraint (in terms of system_value in the query/event service APIs)**. The key idea of this algorithm is that instead of processing two or more unimportant packets which just consume a small amount of energy we would like to process one important packet which may consume relatively larger amount of energy. Reward value is used to denote the important level of packet. A packet with a larger reward value means that this packet is more important. Therefore, the sensor nodes always accept packets which have the highest reward value. Thus, we can guarantee that the most important packets can be processed first.

After deciding which packets are to be accepted, the algorithm will also arrange the packets according to their value. Packets with the largest value will be sent to STB first, meanwhile, FTB will sum the reward value of all the packets having been sent to STB. If the summation is up to the system_value defined in the query/event service APIs, no more packets will be sent to STB. That is to say, all the packets having been sent to STB is enough to solve the query.

Based on this Two Tiers Buffer model and the algorithms above, we introduce the details of our ETRI packet scheduling principles. The principles of ETRI-PF are as follows:

- (1) Whenever a new packet is accepted, its energy consumption should not exceed the remaining energy;
- (2) Whenever a packet is processed, it must meet its deadline;
- (3) Every packet can under Energy, Timing, Reward, and Interest constraints simultaneously;
- (4) It is not necessary to always under these four constraints at the same time;

We can dynamically compose these constraints to filter and schedule packet for heterogeneous sensor nodes and divers working purposes.

4.4.1 Problem Formulation

We define the interested areas as $\Lambda \subseteq \{A_1, A_2, \dots, A_M\}$. From each interested area A_x the cluster head can accept a subset of packets $P_x \subseteq \{P_{x,1}, P_{x,2}, \dots, P_{x,N}\}$. The processing time of the packet $P_{x,y}$ is denoted by $T_{x,y}$. Associated with each packet $P_{x,y}$ there is an Interest value $I_{x,y}$ and a Reward value $R_{x,y}$. Interest value is used to distinguish the interested packets from different areas. Reward value is used to denote the important level of this packet. The larger reward value means the higher important level. These four constraints of algorithm are defined as follows:

- The *energy constraint* imposed by the total energy E_{max} available in the cluster head. The total energy consumed by the accepted packet should not exceed the available energy E_{max} . In other words, whenever the cluster head accept one packet, the energy consumption $E_{x,y}$ of this packet should not be larger than the remaining energy RE .
- The *time constraint* imposed by the global deadline D . The common deadline of this user's data query is D . Each packet that is accepted and processed must finish before D .

- The *interest constraint* imposed by the interest value threshold IT . Each packet that is accepted and processed must satisfy the interest value threshold $IT_{min} \leq I_{x,y} \leq IT_{max}$.
- The *reward constraint* imposed by the *value ratio* $V_{x,y}$ ($V_{x,y} = R_{x,y} / E_{x,y}$) between reward value $R_{x,y}$ and energy consumption of packet $E_{x,y}$. The larger $V_{x,y}$, the packet has, the more valuable the packet is.

The ultimate goal of ETRI-PF is to query a set of packets $P = P_1 \cup P_2 \cup \dots \cup P_M$ among interested packets to maximize the *system value* which is defined as the sum of selected packets' *value ratio* $V_{x,y}$ to meet the *system_value* defined in the query/event service APIs. Therefore, the problem is to

Maximize

$$\sum_{x \in A, y \in P} V_{x,y} \leq \text{system_value} \quad (1)$$

Subject to

$$\sum_{x \in A, y \in P} E_{x,y} \leq E_{\max} \quad (2)$$

$$\sum_{y \in P} T_{x,y} \leq D \quad (3)$$

$$IT_{\min} \leq I_{x,y} \leq IT_{\max} \quad (4)$$

$$x \in A \quad (5)$$

$$A \subseteq \{A_1, A_2, \dots, A_M\} \quad (6)$$

$$y \in P_x \quad (7)$$

$$P_x \subseteq \{1, 2, \dots, N\} \quad (8)$$

Since $P = P_1 \cup P_2 \cup \dots \cup P_M$, we can have the following equation as:

$$\begin{aligned} \sum_{x \in A, y \in P} V_{x,y} &= \sum_{A_1, y \in P_1} V_{A_1,y} \\ &+ \sum_{A_2, y \in P_2} V_{A_2,y} + \dots + \sum_{A_M, y \in P_M} V_{A_M,y} \end{aligned} \quad (9)$$

From equation (9), we can find that the real problem of ETRI-PF is to find out the minimum subset of $P_x \subseteq \{1, 2, \dots, N\}$ to maximize the *system value* to `system_value` from each interested area A_x . Thus, the problem is changed to

Maximize

$$\sum_{A_x, y \in P_x} V_{x,y} \leq \text{system_value} \quad (10)$$

Subject to

$$E_{x,y} \leq RE \quad (11)$$

$$\sum_{y \in P} T_{x,y} \leq D \quad (12)$$

$$IT_{\min} \leq I_{x,y} \leq IT_{\max} \quad (13)$$

$$x \in A \quad (14)$$

$$A \subseteq \{A_1, A_2, \dots, A_M\} \quad (15)$$

$$y \in P_x \quad (16)$$

$$P_x \subseteq \{1, 2, \dots, N\} \quad (17)$$

Inequality (11) guarantees that the *time constraint* is satisfied. Inequality (12) guarantees that only the interested packets are accepted, and inequality (13) guarantees that the energy budget is not exceeded. In order to solve the problem that is presented by (10)-(17), we give the following steps for our ETRI-PF algorithm.

4.4.2 Steps of ETRI-PF

Before sending the real data of a packet to cluster head, sensor node can send its packet's parameters to the cluster head by including them in a small packet, which just consumes very limited energy. We give a name to this kind of small packet as *Parameter Packet (PP)*. There is a physical buffer that exists inside cluster head to store these PPs. After receiving these *parameter packets*, cluster head can decide which packet to be accepted and which packet should be discarded based on these sent parameters. In terms of this Two Tiers Buffer model, basically, we can define our ETRI-PF algorithm into the following steps:

Step 1: Initialization. After receiving $PP \subseteq \{PP_1, PP_2, \dots, PP_N\}$, we assume that tables exist inside the cluster head for storing parameters of every packet i ($i \in PP$): *energy consumption* $E_{x,y}$, *processing time* $T_{x,y}$, *reward value* $R_{x,y}$, and *interest value* $I_{x,y}$. For each PP_i , there are energy consumption for checking CE_i and a period of time for checking CT_i . We also use two structure arrays, $considered(i)$ and $selected(i)$ of size N , to store the information for all received PP s. Initially, we start with an empty schedule ($selected(i).status = false$) and no PP is considered ($considered(i).status = false$). The set of selected PP s (initially empty) is defined as $S = \{(i) \mid selected(i).status = true\}$. After selecting the PP s, cluster head accepts packets that are corresponded to these selected PP s. Therefore, packet's parameters can be expressed as $considered(i).E_{x,y}$, $considered(i).T_{x,y}$, $considered(i).R_{x,y}$, $considered(i).I_{x,y}$, $selected(i).E_{x,y}$, $selected(i).T_{x,y}$, $selected(i).R_{x,y}$ and $selected(i).I_{x,y}$. We define five variables: 1) *checking energy* ($\sum_{i \in PP} CE_i$) is used to store the total energy consumption for checked PP s; 2) *checking time* ($\sum_{i \in PP} CT_i$) is used to store the total processing time for checked PP s; 3) *processing energy* ($\sum_{i \in PP} selected(i).E_{x,y}$) is used to store the total energy consumption for processed packets; and 4) *processing time* ($\sum_{i \in PP} selected(i).T_{x,y}$) is used to store the total processing time for processed packets. 5) *system value summation* ($\sum_{i \in PP} selected(i).R_{x,y}$) is used to store the total value for packets to be processed in STB. These five variables are all initialized to zero.

Step 2: In FTB, we filter and accept packets based on the ETRI constraints. A packet that can be accepted should satisfy all the following criteria:

- ❑ This packet's PP is not considered before ($considered(i).status = false$).
- ❑ The current schedule is feasible ($checking\ time + processing\ time \leq D$).
- ❑ By accepting this packet to current schedule, the energy budget is not exceeded ($checking\ energy + processing\ energy + considered(i).E_{x,y} \leq E_{max}$).
- ❑ This packet is intentionally queried by end user ($IT_{min} \leq considered(i).I_{x,y} \leq IT_{max}$).
- ❑ Among all the PP s that satisfy the above criteria, select the one that has the largest $considered(i).V_{x,y} = considered(i).R_{x,y} / considered(i).E_{x,y}$.
- ❑ By accepting this packet to current schedule, the summation of the system value is just up to the $system_value$ defined in the query/event service APIs. ($\sum_{i \in PP} selected(i).V_{x,y} \leq system_value$)

After choosing the *PP*, cluster head can send Acknowledge back to accept new packet. In addition, for those packets which end user is not interested in, their corresponded sensor nodes will discard them. In this case, we refuse and discard the unnecessary data; consequently, we can reduce the energy consumption by reducing the data transmitting and receiving.

Step 3: In STB, we transmit accepted packets to base station by using Velocity Monotonic Scheduling:

As the algorithm that has been presented in [2], which assigns the priority of a packet based on its requested velocity. VMS minimizes the deadline miss ratios of sensor networks by giving higher priority to packets with higher requested velocities, which also reflects the local urgency. VMS embodies with both the timing constraint and location constraint.

The flowchart and source code of ETRI-PF principles are showed in figure 4.2 and 4.3.

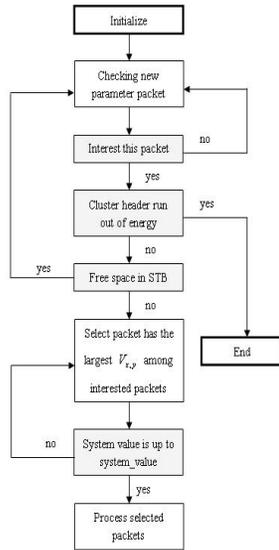


Figure 4.2: Flowchart of ETRI-PF

```

Step1 initialize: selected[i].status = false; considered[i].status = false;
           ∀ i ∈ {1,2,...,M}; checking_energy = 0; checking_time = 0;
           processing_energy = 0; processing_time = 0;
Step2 if(considered[i].interest_value >= interest_threshold) then
           considered[i].status = true;
           if(cluster_header_buff has free space) then
               receive_pp();
           else
               if(current_pp.ratio > min(pp.ratio in buff)) then
                   receive_current_pp; discard min(pp.ratio in buff);
               else
                   discard current_pp;
               end if
           end if
           calculate_checking_time();
           calculate_checking_energy();
           for i to buff_size
               if(checking_time + processing_time > deadline) then
                   stop_working();
               else
                   if(total_ratio < system_value) then
                       if(checking_energy + processing_energy <= total_energy)
                           selected[i].status = true;
                       else
                           change selected[i] with
                               max(selected[i].processing_energy in select);
                       end if
                   end if
               end if
           next
Step3 for i to buff_size
           if(selected[i].status = true) then
               receive_packet(); process_packet();
               calculate_processing_time(); calculate_processing_energy();
           end if
Next
Step4 End
  
```

Figure 4.3: Pseudo code of ETRI-PF

Another aspect: Replace or drop a packet in the STB. A new packet is always accepted if possible. When receiving new *PP* from sensor node, if the STB is full, we can replace or drop a packet based on the following criteria:

- This packet's *PP* is selected ($selected(i).status = true$).

- Among all selected packet's PP_s , find out the one that has the smallest $selected(i).V_{xy} = selected(i).R_{xy} / selected(i).E_{xy}$.
- If this found one is not the new packet that is going to be accepted, we use this new packet to replace this found one, otherwise, we drop this new packet.

4.5 Simulation Result

For the simulation work, we randomly deploy eleven different sensor nodes. And we randomly initialize these sensor nodes with: the total energy of sensor nodes (scope: from 111 to 888), the buffer size of sensor nodes (scope: from 6 to 9). Ten of these eleven sensor nodes are chosen to be the packet generators which randomly create these ten different packets and send to the remaining one. The remaining one works as the cluster head. For this cluster head, we design five parameters: the *total energy* = 666, the *buffer size* = 6, the *deadline* = 5, the *system_value* = 10 and the *interest threshold* = 5. The meaning of threshold is that we just accept the packets when their interest value are larger than 5. Packets from those areas are what the end users are interested in.

In addition, we design ten different packets that are randomly initialized with the following four parameters: energy consumption (scope: from 3 to 10), processing time (scope: from 3 to 10), reward value (scope: from 3 to 10) and interest value (scope: from 3 to 10).

These ten sensor nodes are organized into three groups based on their created packets' interest values. The packets that have the interest values belong to {8, 9, 10} are considered as group A, the packets that have the interest values belong to {6, 7} are considered as group B, and the packet that have interest values belong to {3, 4, 5} are considered as group C. Suppose the cluster head just accepts the packets from area A and B, moreover, within these interested packets it accepts the packet that has the largest $V_{xy} = R_{xy} / E_{xy}$ first. And we also design that this cluster head works in the STB by using the *Velocity Monotonic Scheduling*.

In terms of energy consumption, we mainly consider the following two parts that have strong relationship with our proposed ETRI-PF, which are *processing energy* $\{E_{(Returning_ACK)} + E_{(Receiving_packet)} + E_{(Processing)} + E_{(Broadcasting_event)} + E_{(Listening)} + E_{(Accepting_ACK)} + E_{(Sending_packet)}\}$ and *checking energy* $\{E_{(Accepting_event)} + E_{(Deciding)}\}$. The checking energy is designed to be 0.3, which is 10% of the minimum packet consumption 3; also the checking time is designed to be 0.3, which is 10% of the minimum processing

time 3. Besides ETRI-PF, we provide two different existing packet scheduling algorithms to run on the cluster head for comparison as follows:

1) Compared Algorithm one (CA 1):

- a) In FTB: No *interest constraint* and no *reward constraint*
- b) In STB: Minimizing the packet deadline miss ratio (*Velocity Monotonic Scheduling*)

The cluster head doesn't set any threshold to reduce the incoming packets, but just simply receives packets and relays them. Once it gets a packet, it will process this packet based on the *Velocity* determined by *time constraint* and *location constraint*.

2) Compared Algorithm two (CA 2):

- a) In FTB: Consider *interest constraint*, but no *reward constraint*
- b) In STB: Minimizing the packet deadline miss radio (*Velocity Monotonic Scheduling*)

The cluster head always accepts the packet that has the interest value larger than the interest threshold. Once it gets a packet, it will process this packet based on the *Velocity* determined by *time constraint* and *location constraint*.

We used the following metrics to capture the performance of our routing approach and to compare it with other algorithms:

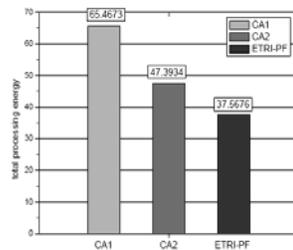


Figure 4.4: Total Processing Energy

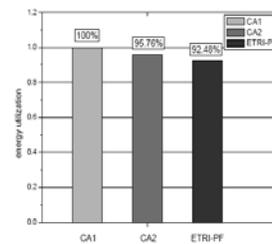


Figure 4.5: Energy Utilization

1) *total processing energy* of cluster head, 2) *energy utilization* of cluster head ($energy\ utilization = processing\ energy / (checking\ energy + processing\ energy)$), 3) *discarded packets ratio* in sensor nodes ($discarded\ packets\ number / total\ created\ packets\ number$ by sensor nodes), 4) *total time consumption* of cluster head ($checking\ time + processing\ time$), 5) *average interest value* per packet, 6) *average reward value* per packet. The simulation results and comparisons are showed as the following figures.

From figure 4.4, we can find that algorithm CA1 costs a lot of *processing energy* and our ETRI-PF algorithm costs only about half of that. The reason is that the cluster header just simply receives the packets and relays them without reducing any incoming packets, neither interest nor reward constraint is considered in algorithm CA1. Take a look at figure 4.5, we find that the *energy utilization* ($= \text{processing energy} / (\text{checking energy} + \text{processing energy})$) of our ETRI-PF algorithm is a little bit lower than the other two algorithms. Remember that we used both interest and reward constraints, which would definitely cost some checking energy, however, we still reduce the energy consumption of whole sensor networks. The saved energy comes from the normal sensor nodes but not from the cluster head.

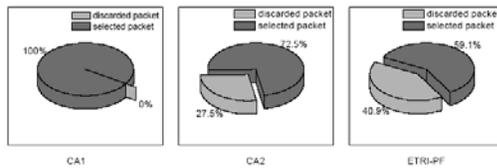


Figure 4.6: Discarded Packet Ratio

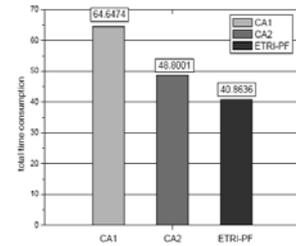


Figure 4.7: Total Time Consumption

Same conclusion can also be drawn from figure 4.6, by analyzing the *discarding ratio* ($\text{discarding ratio} = \text{discarded packets} / \text{total created packets}$), we can see that the discarding ratio of our ETRI-PF is much higher than others. The lower *discarding ratio* the sensor nodes have, the more uninterested packets the sensor nodes send. Thus, the more unnecessary energy is consumed. In conclusion, by using the ETRI-PF, the sensor nodes can reduce the unnecessary transmission of uninterested data to reduce the energy consumption.

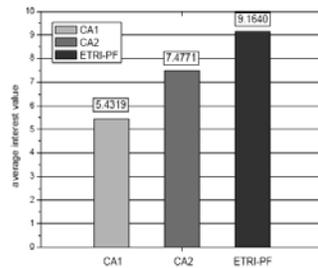


Figure 4.8: Discarded Packet Ratio

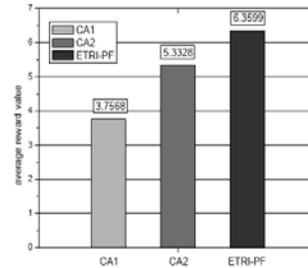


Figure 4.9: Total Time Consumption

Consequently we get figure 4.7 showing the *total time consumption*, even though we need more checking time, we reduce the total time consumption by processing

only part of the packets. For this part, the packets have larger reward than that of the rest packets.

As we presented in foregoing paragraph, we design the *interest threshold* to accept packets that have larger interest values, therefore, the desired *average interest value* should be larger than that of other algorithms. Figure 4.8 shows that the *average interest value* of ETRI-PF is much larger than others, which means the ETRI-PF can exactly process the interested packets well. Figure 4.9 shows the comparison among three algorithms' *average reward values*. In the algorithm CA 1, because we do not intentionally maximize the *value ratio* ($V_{x,y} = R_{x,y} / E_{x,y}$), as a result, the *average reward value* of CA 1 is relatively smaller than others. Compared with CA 2, even though we add the *interest constraint* to CA 2, still no reward constraint is considered, thus the *average reward values* of our ETRI-PF is the largest one. Once again, we demonstrate that our ETRI-QM can deal with the queries more efficiently and get more important information to solve the queries.

4.6 Conclusion and Future Work

Wireless sensor networks consist of nodes with the ability to measure, store, and process data, as well as to communicate wirelessly with nodes located in their wireless range. Users can issue queries over the network. Since the sensors have typically only a limited power supply, energy-efficient processing of the queries over the network is an important issue. In this chapter, we proposed a novel query model ETRI-QM dynamically combining the four constraints: Energy, Time, Reward and Interest. By considering these four constraints simultaneously, we can maximize the system value among the interested packet while satisfying the time and energy constraints by using our ETRI-PF algorithm. In this algorithm, we choose to process packets which have the highest reward value. A packet with a larger reward value means that this packet is more important. Based on our simulation results, we find out that our ETRI-QM and ETRI-PF algorithm can improve the quality of the information queried and also reduce the energy consumption.

However, as we mention the ETRI-QM principle that sensor nodes can know the reward value and interest value of packets well. In the simulation we randomly design the interest value and reward value for 10 different packets. But we do not mention the method that how to design the reward value and interest value for different packets based on each packet's content. Therefore, as a challenge issue to be solved in the future, we are going to explore the appropriate measure methods to evaluate the interest level and important level of different packets.

REFERENCES

- [1] Chien-Chung Shen, Chavalit Srisathapornphat, and Chaiporn Jaikaeo. Sensor information networking architecture and applications. *IEEE Personal Communications*, 8(4):52-59, August 2001.
- [2] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks, In *IEEE RTAS 2002*.
- [3] Samuel R. Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. The Design of an Acquisitional Query Processor for Sensor Networks. *SIGMOD*, June 2003, San Diego, CA
- [4] C. Rusu, R. Melhem, D. Mosse, “Maximizing Rewards for Real-Time Applications with Energy Constraints”, *ACM TECS*, vol 2, no 4, 2003.
- [5] C. Rusu, R. Melhem, D. Mosse, “Multi-version Scheduling in Rechargeable Energy-aware Real-time Systems”, to appear in *Journal of Embedded Computing*, 2004.

VIRTUAL – IP BRIDGE: CONNECTING UBIQUITOUS SENSOR NETWORKS WITH TCP/IP NETWORK

5.1 Introduction

Wireless sensor networks are based on collaborative efforts of many small wireless sensor nodes, which are collectively able to form networks through which sensor information can be gathered. Such networks usually cannot operate in complete isolation, but must be connected to an external network through which monitoring and controlling entities can reach the sensor networks. As TCP/IP, the Internet protocol suite has become the de facto standard for large scale network, it is quite reasonable to connect wireless sensor networks with TCP/IP network to provide meaningful services for large number of Internet users.

Furthermore, in the desired 4G paradigm [1], each mobile device will have global unique IPv6 address, all kinds of heterogeneous wireless networks and current existing IP based Internet should be integrated into one pervasive network to provide transparent pervasive accessibility and mobility for users. Internet users can seamlessly access and use the services provided by heterogeneous wireless networks without knowing which kind of wireless networks they are. Sensor networks as a family member of wireless networks should also be integrated.

In the new appeared pervasive computing paradigm [2], by using ubiquitous sensor networks as the underlying infrastructure, middleware which is considered as the key solution to realize the ubiquitous computing paradigm has been invested in many famous research projects, such as Gaia, Context Toolkit, Aura, TOTA, etc [3]. Ubiquitous sensor networks play an important role in our daily life to provide the seamless pervasive accessibility to users.

However, even though we know it is very important to connect sensor networks with TCP/IP network, the nature limitations of sensor networks, such as limited energy resource and low processing capability make it very difficult to deploy full IP protocol stack in sensor nodes. Therefore, in this chapter we propose a novel bridge based approach to connect ubiquitous sensor networks with TCP/IP network.

In next section, we present a short survey on related researches. Section 5.3 discusses the suitable communication paradigms of sensor networks for connecting with TCP/IP network. In section 5.4, we present the major principle of designing new solution. Section 5.5 presents the key idea and detailed description of our *Virtual – IP Bridge*. An example is presented to make readers easily understand our idea in section 5.6. In section 5.7, we present the comparison between related researches and our approach; in addition, we show that we can easily integrate several different sensor networks into one virtual sensor networks by using our *Virtual – IP Bridge*. Finally, we conclude this chapter in section 5.8.

5.2 Related Work

Since the attention of present research community is mostly paid to other issues of sensor networks, such as energy efficiency and security, very limited numbers of related researches have been performed till now. Basically, these researches can be categorized into two different approaches: 1) *Gateway-based approach*; 2) *Overlay-based approach*.

Gateway-based approach: This is the common solution to integrate sensor networks with an external network by using *Application-level Gateways* [4] as the interface. Different protocols in both networks are translated in the application layer as the figure 5.1 shows. The main role of this gateway is to relay packets to different networks. The advantage of this research work is that the communication protocol used in the sensor networks may be chosen freely. However, this approach has a drawback: Internet users cannot directly access any special sensor node.

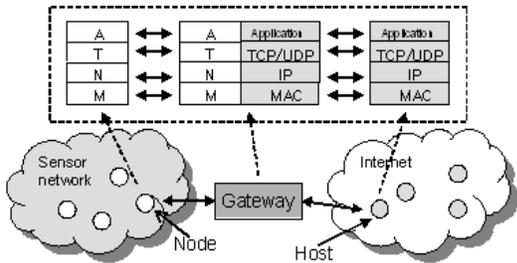


Figure 5.1: Application-level Gateway

Another research work, *Delay Tolerant Network* [5], also follows this *Gateway-based approach*. The key different point from [4] is that a *Bundle Layer* is deployed in both TCP/IP network and non-TCP/IP network protocol stacks to store and forward packets, as figure 5.2 shows. It is very easy to integrate with different heterogeneous wireless networks by deploying this *Bundler Layer* into their

protocol stacks. But the drawback also comes from the deployment of *Bundle Layer* into existing protocols, which is a costly job.

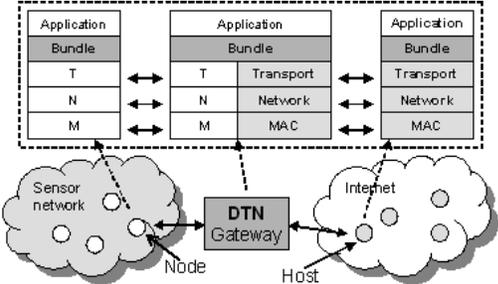


Figure 5.2: Delay Tolerant Network

Overlay-based approach. There are two kinds of overlay-based approaches for connecting sensor networks with TCP/IP network: 1) *TCP/IP overlay sensor networks*; 2) *sensor networks overlay TCP/IP*. Research work in [6, 7] provides a solution to implement IP protocol stack on sensor nodes which is named as **u-IP**. The key advantage of this solution is that Internet host can directly send commands to some particular nodes in sensor networks via IP address. However, this **u-IP** can only be deployed on some sensor nodes which have enough processing capabilities. Another problem is that the communication inside sensor networks based on IP address will bring more protocol overhead, e.g. tunneling. We show this **u-IP** approach in figure 5.3.

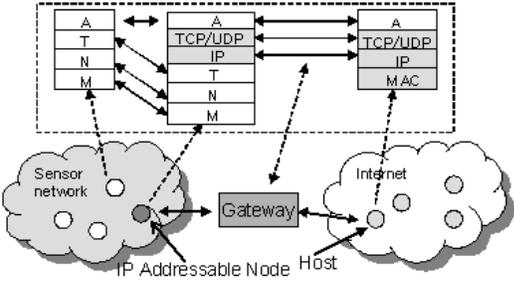


Figure 5.3: TCP/IP Overlay Sensor Networks

The *sensor networks overlay TCP/IP* is proposed in [8]. As figure 5.4 shows, sensor networks protocol stack is deployed over the TCP/IP and each Internet host is considered as a virtual sensor node. By doing so, the Internet host can directly communicate with sensor node and Internet host will process packets exactly as sensor nodes do. The problem of [8] is that it has to deploy an additional protocol stack into the Internet host, which brings more protocol header overhead to TCP/IP network. In addition, it loses the consistency with current

IP based working model, which makes it not suitable to meet requirements of Next Generation Network paradigm.

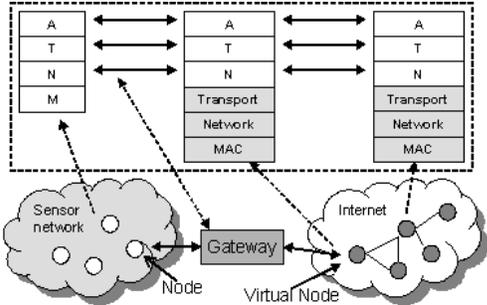


Figure 5.4: Sensor Networks Overlay TCP/IP

By analyzing these related researches and considering requirements of Next Generation Network (NGN) paradigm, it is not difficult to figure out that *we must propose a new approach which can cover all the advantages of existing researches and still has the consistency with IP based working model to realize the NGN paradigm*. So, what are the major principles of designing this new approach? Before presenting the major design principles let us have a look at the different communication paradigms of sensor networks for more detailed analysis.

5.3 Communication Paradigms of Sensor Networks

Typically, there are three kinds of communication paradigms in sensor networks: 1) *Node-Centric*, all sensor nodes are labeled with some names (IDs) and routing is performed based on these names (IDs), e.g. some table-driven-routing protocols; 2) *Data-Centric*, trying to make sensor networks answer “Give me the data that satisfies a certain condition”, e.g. Directed-Diffusion [9]; 3) *Location-Centric*, using the location of sensor nodes as a primary means of address and routing packets, e.g. CODE [10].

Then, which communication paradigm is suitable for connecting sensor networks with TCP/IP network? In nowadays Internet, every network entity such as personal computer, router, or printer has its own IP address for identifying itself from others. Commercial databases used to provide diverse services for Internet users are stored in different computers. Internet users can access these services by using the IP addresses of those computers. However, the difficulty of remembering IP address for service motivates the using of Domain Name, which probably uses the name of this service. Internet users can easily use the Domain Name to access the corresponding service, with the assumption that this service’s domain name or IP address can be known by users in advance. The

routing in Internet is also IP address based. *This kind of working model is similar with those of Node-Centric and Location-Centric.*

Data-Centric approach presented in paper Directed Diffusion [9] has its foremost different assumption from the IP based Internet working model: **users don't know the exact locations of their interested sensors or data in advance.** In order to find the needed data, users request the gateway to broadcast the *Interest packet* to all the sensor nodes of sensor networks and look for the data source. On the other side, the sensor nodes which have the needed data also broadcast the *advertisement packet* to tell other nodes that they have this kind of data. Once the *Interest packet* and *advertisement packet* meet each other in certain sensor node, the transmission path from data source to gateway will be set up. If we consider the data provided by these sensor nodes as the services, we realize that the working approach of *Data-Centric* is more like a Service (Data) Discovery approach.

Now we can easily answer that “*In order to provide the consistency between the working models of sensor networks and TCP/IP network, the Node-Centric and Location-Centric communication paradigms are more suitable for connecting sensor networks with TCP/IP network.*”

After having these aforementioned analyses, we can present our major design principles in the following section now.

5.4 Major Design Principles

These following principles of designing our new approach must be clearly figured out, so that we can successfully deploy a comprehensive approach to connect sensor networks with TCP/IP network.

- **Consistency:** The new approach should be IPv6 based, because it should have the consistency with the working paradigm of Next Generation Network.
- **Transparency:** By using IP based approach, non-system-designer users should be able to use services provided by sensor networks without knowing that “these services are provided by certain sensor networks.”
- **Energy efficiency:** Sensor networks should be able to freely choose routing protocol to optimize energy efficiency and performance.

- **Direct accessibility:** Some sensor nodes should be able to be accessed and operated by Internet users directly by using IP address to identify them from others.
- **No overlay approach:** Because both of *TCP/IP overlay sensor networks* and *sensor networks overlay TCP/IP* require modification on protocol stacks.
- **Easy integration between different sensor networks:** Several locating in different place's sensor networks should be easily integrated into one virtual sensor networks based on IP addresses.
- **Taking the advantage of knowing sensor node's label (ID) or location address:** Because both sensor nodes' label (ID) and location addresses are unique information inside sensor networks, it can be used to identify different sensor nodes.

5.5 V – IP Bridge

5.5.1 Key Idea

Taking all of these foregoing principles into consideration, we create our key idea *Virtual – IP Bridge: Basing on Node-Centric or Location-Centric communication paradigm, mapping the node label (ID) or location address with IP address in bridge. The IP address will not be physically deployed on sensor node, but just store in bridge as a virtual IP address for Internet users.* Packets that come from one side will be translated into corresponding packet formats and sent to another side by this *Virtual – IP Bridge*. We describe the system model of *Virtual – IP Bridge* in the following subsection.

5.5.2 System Model of Virtual – IP Bridge

In this *Virtual – IP Bridge*, there are two major components to translate packets for both sides, as figure 5.5 shows: 1) *TCP/IP Network -> Sensor Networks (T->S) Packet Translation*, translating packets from TCP/IP network into the packet format of sensor networks; 2) *Sensor Networks -> TCP/IP Network (S->T) Packet Translation*, translating packets from sensor networks into the packet format of TCP/IP network. We use *T->S Packet* to represent the packet that comes from

TCP/IP network, and *S->T Packet* to represent the packet that comes from sensor networks.

The packet format of original *T->S Packet* has four major fields:

- 1) *User IP*, used to represent the IP address of user's who sends this packet;
- 2) *Sensor IP/Bridge IP*, used to represent the destination of this packet, which can be the bridge IP address or some special sensor node's IP address;
- 3) *Q/O*, used to represent packet type, which can be *Query Command* or *Operation Command*;
- 4) *Complicated/Simple Data Request / Operation Command*, used to represent the real content that is carried by this packet.

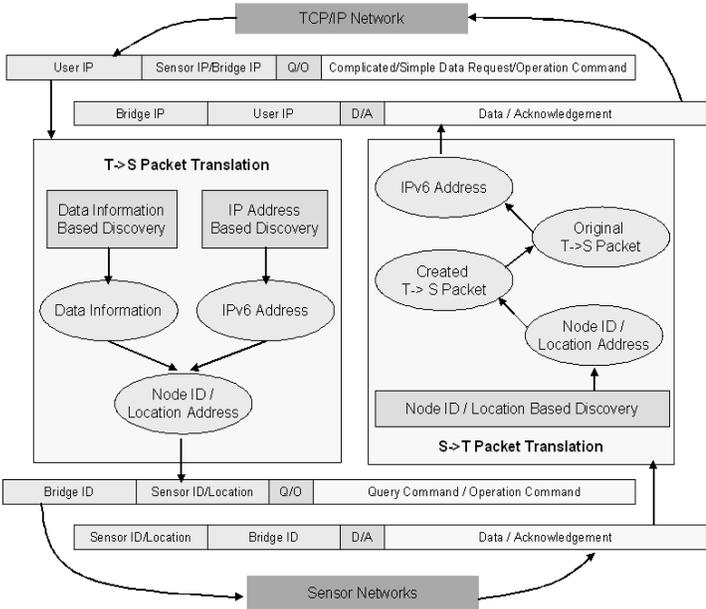


Figure 5.5: Architecture of V – IP Bridge

The packet format of created *T->S Packet* has the following four major fields:

- 1) *Bridge ID/Location*, used to represent the ID or location address of Bridge, which sends the packet to sensor networks;

- 2) *Sensor ID/Location*, used to represent the node ID or location address of data source;
- 3) *Q/O*, used to represent packet type, which can be *Query Command* or *Operation Command*;
- 4) *Complicated/Simple Data Request / Operation Command*, used to represent the real content that is carried by this packet.

The *Query Command* is used to request data from sensor networks, it can be as simple as query data just from one special sensor node, or it can be as complicated as query data from many sensor nodes at the same time. *Operation Command* is used to remote control one special sensor node's working status.

Similarly, the packet format of *S->T Packet* also has four major fields:

- 1) *Sensor ID/Location*, used to represent the node ID or location address of data source;
- 2) *Bridge ID/Location*, used to represent the ID or location address of Bridge, which is the destination of this packet;
- 3) *D/A*, used to represent packet type, which can be *Data Packet* or *Acknowledgement Packet*;
- 4) *Data/Acknowledgement*, used to represent the real content that is carried by this packet.

The packet format of created *S->T Packet* has the following four major fields:

- 1) *Bridge IP*, used to represent the IP address of Bridge, which sends the packet to TCP/IP network;
- 2) *User IP*, used to represent the IP address of user's who will receive this packet;
- 3) *D/A*, used to represent packet type, which can be *Data Packet* or *Acknowledgement Packet*;
- 4) *Data/Acknowledgement*, used to represent the real content that is carried by this packet.

The *Data Packet* corresponds to the *Query Command*, and the *Acknowledgement Packet* corresponds to the *Operation Command*.

A *Node ID/Location Address* is the node ID or location address of a sensor node. A *Data Information* is a description about what kind of data can be provide by this sensor node. An *IPv6 Address* is the assigned IP address for this special sensor node. *Virtual – IP Bridge* will actively collect *Node ID/Location Address*, *Data Information* for all sensor nodes, and also actively assign *IPv6 Address* for these sensor nodes. All these information are stored in a database which physically locating in the *Virtual – IP Bridge*. Furthermore, ***bridge will map these three different kinds of information with each other.***

In next subsection, we will present the detailed workflow of two translation components to explain how we translate different packets for both sides.

5.5.3 Workflow of Both Translation Components

TCP/IP Network -> Sensor Networks Packet Translation: After receiving packets from TCP/IP network, there are two different ways to translate them into the packet format that used by sensor networks: 1) *Data Information Based Discovery*; 2) *IPv6 Address Based Discovery*.

The translation workflow is showed in figure 5.6. Bridge will analyze these received packets based on the field “*Q/O*” to categorize them into *Query Command* and *Operation Command*.

If a packet is an *Operation Command*, then bridge can base on the *Sensor IP* to search the database to find out the corresponding *Node ID/Location Address* of this sensor node through the mapping between *IPv6 Address* and *Node ID/Location Address*.

If a packet is a *Query Command*, then bridge can base on *Complicated/Simple Data Request* to search the database to find out the corresponding *Node ID/Location Address* of this sensor node through the mapping between *Data Information* and *Node ID/Location Address*.

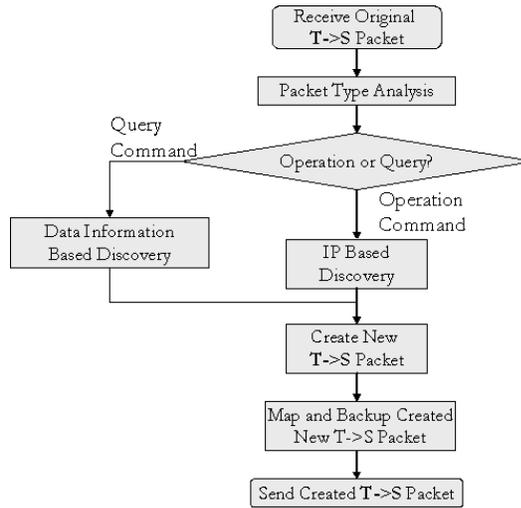


Figure 5.6: Translation workflow of T -> S

After knowing *Node ID/Location Address* of this sensor node, we can easily create the new packet for sensor networks. Before sending new created packet to sensor networks, we backup this new *T->S packet*, and map it with the original *T->S packet* in bridge. These saved packets will be used when we translate packets that come from sensor networks into the packet format of TCP/IP network.

Sensor Networks -> TCP/IP Network Packet Translation: The workflow of *S->T translation* is different from the workflow of *T->S translation* as figure 5.7 shows.

After receiving the *S->T Packet* from sensor networks, bridge first bases on packet's *Sensor ID/Location* to find out the created *T->S Packet*, then through the mapping between the created *T->S Packet* and the original *T->S Packet*, bridge can easily find out the original *T->S Packet*.

By analyzing the original *T->S Packet*, bridge can get the *User IP*, and then create the new *S->T Packet*. Before sending this new *S->T Packet*, bridge will delete the corresponding original and created *T->S Packets* to save the storage space of the database.

To make readers easily understand our idea, we are going to present an example which is named as *G-IP Approach* to connect Grid based (*Location-Centric*) sensor networks with TCP/IP network. The example for *Node-Centric* will be presented in our future work.

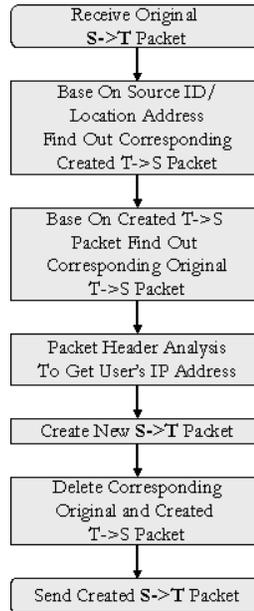


Figure 5.7: Translation workflow of S -> T

5.6 An Example: G – IP Approach

We use our previous research work CODE [10] as the routing protocol in this *G-IP approach*. In CODE, we assume sensor networks are homogeneous and sensor nodes have the knowledge about their residential locations. The basic idea of CODE is to divide sensor networks field into grids, and grids are indexed based on their geographical locations. Each grid contains one sensor node works as the coordinator or cluster head to intermediately cache and relay data.

5.6.1 Active Data Discovery and Registration

After building up grids, each coordinator actively senses its local environment and registers the *Data Information* about the sensed data to bridge. The *Data Information* of a grid means what kind of data that can be provided by this grid. For example, sensor nodes within grid [1.0] can sense temperature data for its local environment, then the coordinator of this grid registers in bridge that grid [1.0] can provide some temperature information. By doing so, bridge can have the location address (Grid ID) about the interested data (sensors) in advance. For example, whenever the Internet users want to get some temperature data, bridge can forward the Query packet to the grid [1.0] directly.

5.6.2 Data Information & Grid ID & IP address Mapping

After active data discovery and registration, bridge can have *Data Information* and grid ID for whole sensor networks. In this step, we assign global unique IP address for each grid in bridge. Technically, it is possible to assign IPv6 address to every sensor node because IPv6 can provide enough IP address for whole sensor networks.

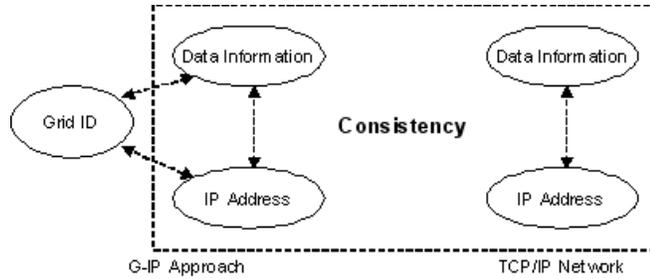


Figure 5.8: Hide the Grid ID to make the consistency with TCP/IP network (Internet)

However, if we assign IP address to every sensor node, the cost for maintenance could be very high. Therefore, we only assign the IP address to each grid's coordinator to reduce the cost for setting up and management. Furthermore, we make the data, grid ID, IP address mapping in bridge as the left part of figure 5.8. By doing this kind of mapping, whenever Internet users want to get some data, they can easily find out its exact location through the corresponding IP address and Grid ID. However, we are trying to use IP address instead of Grid ID. Because once we can hide the Grid ID from Internet users, we can have the consistency between traditional IP based Internet and our G-IP approach, as figure 5.8 shows. In order to clearly differentiate G-IP approach from related researches, we present our system model in figure 5.9. A *Grid ID – IP Address Mapping Layer* is deployed above network layer to translate packets for both sides.

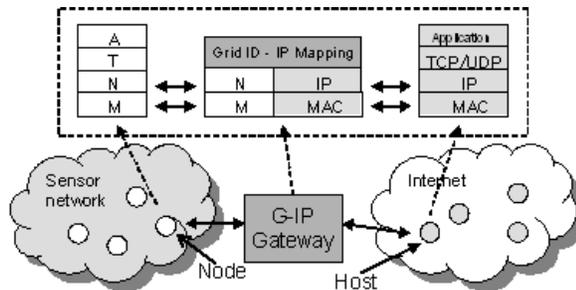


Figure 5.9: G – IP Approach

5.6.3 Packets Translation in Grid ID – IP Address Mapping Layer

In this subsection, we present several different kinds packet translations that can be done by *Grid ID – IP Address Mapping Layer*.

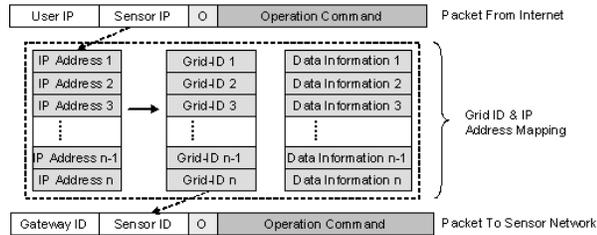


Figure 5.10: Directly Operate Coordinator

In figure 5.10, one Internet user sends an Operation Command to one special coordinator to change its working status. The *Target IP* is the IP address of this special coordinator. Here, we assume that this Internet user can get this *Target IP* from sensor networks developer. After receiving the packet from Internet, the bridge will search the table inside the *Grid ID – IP Address Mapping Layer* to find out the Grid ID of this coordinator, and then create another packet for routing in sensor networks. In this chapter, we just provide the solution to directly access coordinator or cluster head. The solution of how to directly access a normal sensor node in grid is planned in our future work, since it needs a more detailed design in *inside-grid* level.

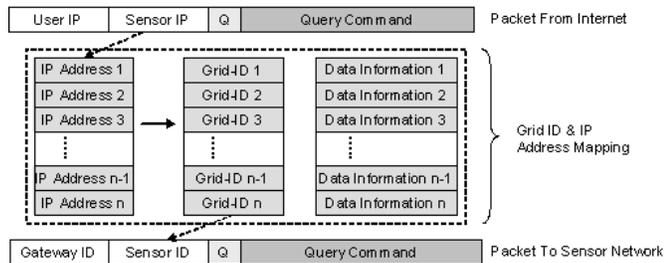


Figure 5.11: Directly Query based on IP address

Similar with the operating coordinator, we also can directly query data from the interested coordinator as figure 5.11 and 5.12. Internet users can directly query the data basing on *IP address* or *Data Information*. However, generally the *Data Request* from Internet does not want to simply get data from one special sensor

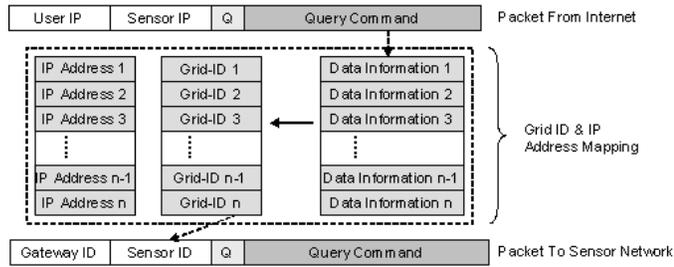


Figure 5.12: Directly Query based on Data Information

node, but needs the collaboration result of several sensor nodes. Therefore, we also can perform some attribute based discovery mechanism inside bridge as figure 5.13 shows. The G-IP mapping layer can create several sub-query commands for different requested data based on the *Complicated Data Request*.

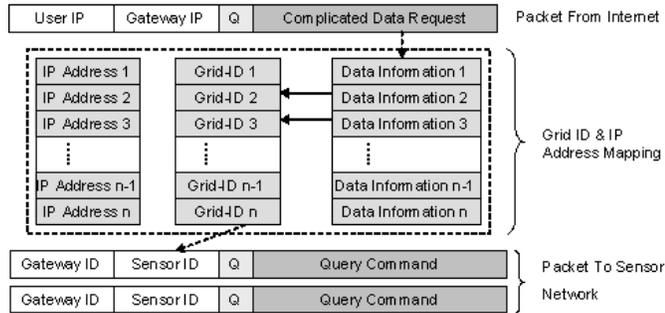


Figure 5.13: Complicated Data Request from several coordinators

After querying, packets that originally come from the sensor networks can also follow the following procedure to be sent back to Internet user, as figure 5.14 shows. Bridge first bases on packet's *Sensor ID/Location* to find out the created *T->S Packet*, then through the mapping between the created *T->S Packet* and the original *T->S Packet*, bridge can easily find out the original *T->S Packet*. By analyzing the original *T->S Packet*, bridge can get the *User IP*, and then create the new *S->T Packet*.

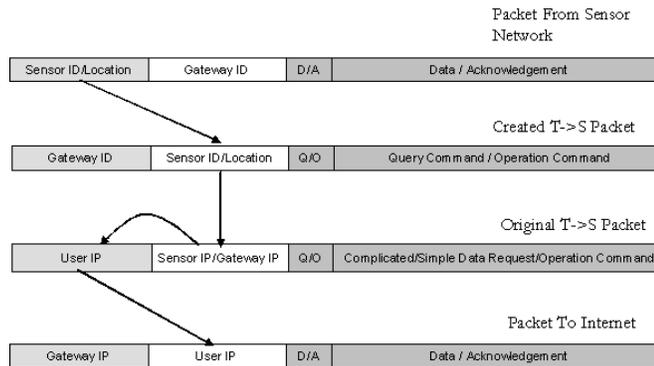


Figure 5.14: Send $S \rightarrow T$ Packet to TCP/IP network

5.7 Discussion

5.7.1 Comparison with Related Researches

After understanding our idea, we think that a table based comparison with related researches is essentially necessary to prove that our solution can cover most of the benefit of related researches, as Table 5.1 shows.

After the integration of sensor networks and TCP/IP network, we can still keep the consistency with the IP based working model by hiding the Grid ID (node ID). Because in the view of Internet users, the sensor networks is IP based, they don't need to know which kind of routing protocol is used in sensor networks. In other words, sensor networks are transparent to Internet Users. However, for *sensor networks overlay TCP/IP*, users always have to deploy corresponding sensor networks routing protocol on Internet hosts, which means that users must know what kind of sensor networks they are.

Since we only deploy virtual IP addresses in bridge, rather than bring any modification to sensor networks protocols, sensor networks can still freely choose the optimized routing protocol which is *Node-Centric* or *Location-Centric* based. However, the *TCP/IP overlay sensor networks* must modify the protocol stack of sensor networks.

Table 5.1 Comparison with related researches

	Application level gateways	Delay Tolerant Network	TCP/IP overlay sensor networks	Sensor networks overlay TCP/IP	Virtual IP
Consistent with Internet working model	No	No	Yes	No	Yes
Transparent for Internet users	Yes	Yes	Yes	No	Yes
Freely choose routing protocol in sensor networks	Yes	Yes	No	Yes	Yes
Directly accessibility some special sensor node	No	No	Yes	Yes	Yes
Easy to integrate different sensor networks	No	Yes	No	Yes	Yes

Furthermore, Internet users can easily and directly access some special sensor nodes via *virtual IP addresses*. Since sensor networks can be *virtual-IP based*, it is very easy to integrate several locating in different place's sensor networks into one *virtual sensor networks*. Because we consider the integration of different sensor networks as a new research issue in the field of ubiquitous sensor networks, we are going to have more discussion about it in the following subsection.

5.7.2 Integration of Different Sensor Networks

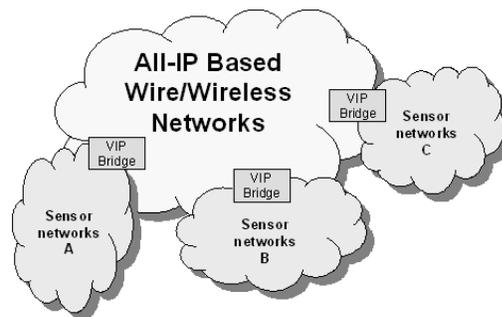


Figure 5.15: Several sensor networks deployed in different locations

Sensor networks which are physically located in different locations may use totally different routing protocols for their specific applications, as figure 5.15 shows. Sometimes these sensor networks should be integrated into one virtual sensor

networks over wired/wireless networks to provide comprehensive services for end users.

In *Delay Tolerant Network*, because they deployed an additional *Bundle Layer* in both TCP/IP network and non-TCP/IP network protocol stacks, it is very easy to integrate different networks into one virtual network. However, it requests a lot of effort to modify existing routing protocols to deploy this new *Bundle Layer*.

In *sensor networks overlay TCP/IP*, if several sensor networks are only physically located in different locations but still use the same routing protocol, users can deploy this routing protocol to overlay TCP/IP networks, so that these sensor networks can be integrated into one virtual sensor networks. If these sensor networks are using different routing protocols, then this *sensor networks overlay TCP/IP* is not suitable to integrate them into one virtual sensor networks.

Compared with our *Virtual – IP Bridge* approach, either *Delay Tolerant Network* or *sensor networks overlay TCP/IP* needs to deploy or modify current existing protocol stacks. If these sensor networks have bridges which have *virtual IP addresses*, then it is very easy to integrate them into one virtual sensor networks, because *virtual IP address* can hide all the heterogeneities of different sensor networks for upper layers.

5.7.3 Drawbacks of Virtual – IP Bridge

Even though we claim that our *Virtual – IP Bridge* can cover most of the benefits of related researches, through the prototyping work we realize that our approach also has several drawbacks:

- 1) ***Single point of failure***: once this *Virtual – IP Bridge* is failure, the sensor networks that connected to this bridge will not be able to be used any more.
- 2) ***Bottleneck problem***: because of these packets need to be translated into different packet formats when they are sent to different sides, if the processing capability of this *Virtual – IP Bridge* is not powerful, it's easy to occur the bottleneck problem, which slows down the performance of whole system.

One major limitation of our *Virtual – IP Bridge* is that the routing protocols in sensor networks must be *Node-Centric* or *Location-Centric* based, which means many *Data-Centric* based routing protocols will not be supported.

5.8 Conclusion and Future Work

Pervasive network which is considered as the next generation of current networks requests us to integrate all kind of heterogeneous networks into one global network. Sensor networks as a family member of wireless networks should also be integrated. In this chapter we present a new solution to connecting ubiquitous sensor networks with TCP/IP network. By comparison with related researches we claim that our new solution can cover most of the benefits of related researches.

Through our prototyping work, we find out that our *Virtual – IP Bridge* still has some limitations and drawbacks, which we are going to solve in our future work. Here, we want to clearly point out that how to analyze one *Complicated Data Request* and create several *sub-Simple Data Requests* is another research issue, which is currently under the investigation of another research team in our group. The major contribution of this chapter is that we present a comprehensive new solution to connect ubiquitous sensor networks with TCP/IP network.

REFERENCES

- [1] Gary Legg, “Beyond 3G: The Changing Face of Cellular”, <http://www.techonline.com-/community/home/37977>
- [2] Saha, D., Mukherjee, A. “Pervasive Computing: A Paradigm for the 21st Century”, IEEE Computer, Volume: 36 Issue: 3, March 2003 Page(s): 25 – 31
- [3] N.Q. Hung, N.C. Ngoc, L.X. Hung, Shu Lei, and Sungyoung Lee, “A Survey on Middleware for Context-Awareness in Ubiquitous Computing Environments”, Korean Information Processing Society Review ISSN 1226-9182 July 2003.
- [4] Z. Z. Marco, K. Bhaskar, “Integrating Future Large-scale Wireless Sensor Networks with the Internet”, *USC Computer Science Technical Report CS 03-792*, 2003.
- [5] K. Fall, “A Delay-Tolerant Network Architecture for Challenged Internets”. *In Proceedings of the SIGCOMM 2003 Conference*, 2003
- [6] A. Dunkels, J. Alonso, T. Voigt, “Making TCP/IP Viable for Wireless Sensor Networks”, In *Proceedings of the First European Workshop on Wireless Sensor Networks (EWSN'04), work-in-progress session*, Berlin, Germany, January 2004.
- [7] A. Dunkels, J. Alonso, T. Voigt, H. Ritter, J. Schiller, “Connecting Wireless Sensornets with TCP/IP Networks”, In *Proceedings of the Second International Conference on Wired/Wireless Internet Communications (WWIC2004)*, Frankfurt (Oder), Germany, February 2004.
- [8] H. Dai, R. Han, “Unifying Micro Sensor Networks with the Internet via Overlay Networking”, in *Proc. IEEE Emmets-I*, Nov. 2004.

- [9] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", Proc. ACM MobiCom'00, Boston, MA, 2000, pp. 56-67
- [10] L.X. Hung, and S.Y. Lee, "A Coordination-based Data Dissemination for Wireless Sensor Networks", Proceeding of International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 05) Melbourne, Australia December, 14-17, 2004.

INDIVIDUAL CONTOUR EXTRACTION FOR ROBUST WIDE AREA
TARGET TRACKING IN VISUAL SENSOR NETWORKS

6.1 Introduction

All Intelligent surveillance or monitoring is an emerging area for sensor networks application. An important task to be performed by a network of visual sensors (e.g. camera) or multimode sensors (e.g. camera with microphone and thermometer) distributed in a geographic area is to track target in a local area in order to monitor unusual activities. We use visual sensors and multimode sensors other than one single type of sensor because with the increase of computing power of hardware it becomes more feasible to utilize richer representations of features other than that used in current Decentralized Data Fusion algorithms. A lot of research works have been done based on common wireless sensor networks (WSNs) equipped with only acoustic sensors or temperature sensors, and they work well in ideal environment without much interference or noise. In case of real environment, however, the noise or uncertainty often has bad effect on target tracking results.

Before tracking, the sensors close to the predicted path of the target need to be alerted. The target can be a moving vehicle or can be a phenomenon such as an approaching fire. It is assumed that each individual sensor node is equipped with multimode sensory devices in order to detect the target based on the sensed data. The sensors that are triggered by the target collaborate to localize the target in the physical space [1]. One of the central issues for collaborative signal and information processing to be addressed is energy constrained dynamic sensor collaboration: how to dynamically determine who should sense, what needs to be sensed, and who the information must be passed on to.

Target tracking in visual sensor networks (VSNs) is a challenging problem that requires acquiring and processing data from multiple camera views (a single camera often cannot see every object). Besides, any practical sensing device has limitations on its sensing capabilities (eg. resolution, bandwidth, efficiency, etc.). The descriptions or physical models built on the data sensed by a device are, unavoidably, only approximations of the real nature. These uncertainties, coupled with the reality of occasional sensor failure greatly compromise reliability and reduce confidence in sensor measurements. In addition, the spatial or physical limitations of sensor devices often mean that only partial information can be

provided by a single sensor. Thus, target tracking process should be robust because the existence of faulty sensors deteriorates the difficulty of target tracking problem. False alarms waste network resource as well as cause other negative effect. In WSN environment, the traditional double or triple redundancies are not adequate solutions due to their power consumption, space, and cost. We attempt to consider qualitatively fault tolerance in sensor collaboration using a simple but efficient method.

In case of existence of multiple objects including both target and uninterested background objects, which also complicates tracking problem, genetic fitting is proposed to distinguish them especially when the target is closely located with other object (s).

All in all, in this paper, our contribution is that we considered fault tolerant sensor collaboration in target tracking process by a low computation cost method in wireless VSN environment and a B-spline contour fitting approach based on genetic algorithm (GA) for efficient contour extraction is proposed as inter-scene vision method to detect the target. It produces accurate detection and tracking especially when the interesting target is closely located beside other objects. Compared with the existing classic contour extraction method B-spline, and existing graph matching method also based on VSNs, our method shows significant improvement in terms of success rate of target tracking.

This chapter is organized as follows: related work and VSNs environment are sketched in section 2 and section 3, respectively. The method to extract individual contour is proposed in section 4, followed by experimental results in section 5. Some concluding remarks and future work are provided in section 6.

6.2 Related work

The problems of target detection and tracking have been explored in [2] on an individual node basis. There is, however, little research on distributed detection and tracking within WSNs. Topics of target tracking have been studied and developed extensively but primarily in the domain of active and passive radar. Graphical modeling techniques such as Kalman filtering and Hidden Markov Models [3, 4] have been employed successfully in this domain. Complex multiple hypothesis testing techniques are incorporated into their frameworks that rigorously evaluate every possible origin of the measurements received. However, they assume that all the measurements are available for processing at a centralized node. In [5], the authors proposed a distributed energy-aware collaborative target-tracking algorithm using Kalman filtering technique. However, their algorithm

was only suitable for one single target case and it can move solely in a straight line. It is not practical in most cases as multiple targets may exist and they often move in at least a 2-D space. Another drawback is that they choose initial active sensor randomly for sensing information. It cannot be energy efficient and fault tolerant when the active sensor is located farthest away from the target. In [2], the authors used a classification algorithm to disambiguate closely located targets. However signals received from targets are correlated and we cannot recover the uncorrelated signals always. Since we do not know in advance the number of targets around each sensor, the problem is ill-posed and very challenging even for a highend computer.

There are also some researches for target tracking based on VSN environment. In [13], the authors proposed a real remote monitoring system for all day outdoor observation using wireless communication, but the difference with ours is that this system was used for small scope area target tracking because it has only one camera equipped. If equipping multiple cameras for wide area tracking, the collaboration among sensors and cameras should be reconsidered. G. Kogut, etc [6] established wide-area camera network with non-overlapping fields of view in which a wide-area tracking algorithm is tested to track moving objects. This algorithm employs a variety of vehicle features in a feature vector: color, shape, and velocity. The color features are calculated by using partial implementation of the AutoColor Matching System to compensate for differences between illumination at cameras sites and between cameras. The tracking algorithm uses the color model and blob centroids from the segmentation module to help solve the data association problem. However their simulation results show that the graph matching method is not accurate enough for target tracking.

For fault tolerant target detection or tracking, Clouqueur, etc [7] seek algorithms to collaboratively detect a target region. Each sensor obtains the target energy (or local decision) from other sensors, drops extreme values if faulty sensors exist, computes the average, and then compares it with a pre-determined threshold for final decisions. For these algorithms, the challenge is the determination of the number of extreme values. This is unavoidable when using “mean” for data aggregation. In [8], the authors explored the utilization of “median” to effectively filter out extreme values for target region detection and claimed their algorithm more computationally inexpensive than [7]. However only readings of neighbors are included for computing the median value, and a sensor S_i is determined as an event sensor if the estimated value is larger than a predefined threshold. The drawback of this algorithm is that they omit the case when S_i is just the faulty sensing node.

In this chapter, we attempt to solve the target-tracking problem on VSN platform using fault tolerant sensor collaboration, and genetic fitting method as in-network data fusion, especially when the target is closely located with other object (s).

6.3 Visual sensor networks: architecture

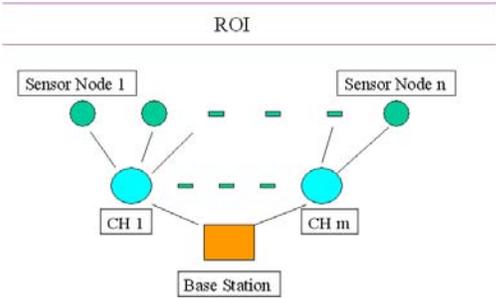


Figure 6.1: Visual sensor networks architecture for target detection

We envision a system that covers a remote area with wireless sensors, cluster heads (CHs) [9] and base station as shown in Fig. 1, which has nearly uniform sensor distribution with optimal coverage and optimal CHs positions to reduce transmission energy consumption proposed by our previous work [10]. We assume that each sensor has a camera and other simple sensing devices equipped. Sensor nodes communicate with their CHs, and CHs transmit compressed image data to the base station, which is located far away. CH usually works based on an event driven method, i.e., it works according to the wireless sensor’s signal indicating the existence of target. Only sensors positioned in the interested field of view and among them only those who have robust data, can be selected to send images to CH. In this way, large amount of energy can be saved. Fusion of information from different sensors would allow for tracking the interesting target. In addition to triggering appropriate responses, results from such an analysis would be stored in a database of base station [11]. This would allow statistical analysis of past events. Having multimode sensors provides the ability to reduce the probability of detecting some types of fault information, and to provide more information than with a single camera.

The use of multimode-sensor architectures guarantees a proper coverage of all possible conditions of operation, thus satisfying the desired requirements in terms of system’s global performance.

6.3.1 Image acquisition

In the image acquisition stage, the input to each single-sensor node is a stream of raw images. There are no requirements of the properties of the images. However the quality of the acquired data may affect the overall performance and quality of the processing steps. The rate of image acquisition may also influence processing results, depending on the velocity of the moving objects being tracked, and the perspective by which the objects are being viewed. Sensor location tends to be a contexts-dependent problem, determined largely by the specific application requirements.

6.3.2 Sensor Collaboration

Deploying VSNs aims to cover a wider area than possible with a single camera. Viewing the same object from different positions has two main advantages [12]. Firstly, event detection can be made more robust by cross-validating information. For example, if two camera-equipped sensors in a surveillance system see at the same time a moving object approximately located in the same position in space, then the confidence of this event detection increases. Secondly, range information can be estimated by triangulation, allowing for a very powerful geometric scene description.

For all kinds of weather and darkness, the camera can acquire images using the functions of auto focus, auto iris, lens zooming, and night vision. The camera can automatically change into an infrared one, depending on the lightness [13].

To achieve robust sensor collaboration, we proposed four steps fault tolerant method:

- (1) Discover neighbors within cluster range.
- (2) At every node in the cluster, compute the “median” value among the readings of node i and all its neighbors for eliminating the extreme values. This is different from average method [7] which firstly drops extreme values and then computes the average.
- (3) Choose those whose readings are above a particular threshold (e.g., “temperature is higher than A”, “acoustic signal intensity is higher than B”) as the event nodes
- (4) The chosen sensor nodes indicate the presence of a target in the vicinity. Then, binary messages are sent by the chosen nodes to CH. When such a message is received by CH, it demands the image data to be sent from the originator.

If a faulty sensor constantly reports wrong information (obviously different from other sensors), it should finally be excluded from the sensor network. This

proposed fault tolerant idea has low computation and communication overheads compared with related fault tolerant methods, by analysis in a qualitative way.

6.3.3 Inter-scene tracking

Inter-scene tracking is the fusion of tracking data received from multiple individual sensors to allow target to be tracked as they move along nodes with or without overlapping fields of view. This type of tracking is impossible with data from single nodes working on independently. The input of the inter-scene tracking data is the pre-processed output from the intra-scene tracking process [14]. Suitable communication protocol is assumed in the VSN, so the detailed design of novel protocol is out of the scope of this chapter.

6.4 Object tracking by extracting individual contour

We propose the individual contour extraction approach based on GA in the inter-scene tracking stage. GA is a probabilistic technique for searching an optimal solution. The optimal solution is described by the value of vector X , which is called as the “chromosome” in GA, and it can be obtained by minimizing an objective function $f(X)$. Hence, the definition of the objective function significantly affects the solution state X . In the proposed algorithm, a chromosome consists of n control points of B-spline, which is similar to the chromosome design of MacEachern [15]. Since the chromosome represents a complete contour and a gene uses the actual location of a control point, the search algorithm has neither ambiguity on the contour location nor potential bias to particular shapes. To reduce the size of a gene, we use the index value as a gene, instead of two coordinate values. The chromosome representation using the indices enables us to use the search area with any shape and produce shorter chromosomes, which generally result in faster convergence rate.

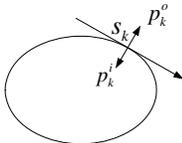


Figure 6.2: Definition of inner and outer regions

The fitting process to extract a contour starts with the generation of the initial population. The initial population consists of a set of chromosomes selected randomly from the search windows for control points. A new generation is evolved through evolutionary operations. The end of the evolutionary operation

is determined by checking the fitness values, which represent the goodness of each chromosome in the population [16]. In this paper, we introduce a fitting function that rejects the contours of the nearby object specially designed for the contour extraction in interesting target image sequences.

To compute the fitting value for a possible solution, we first generate the contour points of B-spline and trace the contour as shown in Figure 2. At the k'th contour point $\mathbf{r}(s_k)$, compute a normal vector $\mathbf{n}(s_k)$, and identify the inner region and outer region pixel location \mathbf{p}_k^i and \mathbf{p}_k^o from the curve by using equation (1) and (2).

$$\mathbf{p}_k^o = \mathbf{r}(s_k) + \mathbf{n}(s_k) \quad (2)$$

$$\mathbf{p}_k^i = \mathbf{r}(s_k) - \mathbf{n}(s_k) \quad (2)$$

The fitting value can be determined based on gradient magnitude information at each contour point by using the equation (3) and (4).

$$f(\mathbf{x}) = \left| \sum_{k=0}^{M-1} grad_k \right| \quad (3)$$

$$grad_k = \begin{cases} |\nabla I(\mathbf{r}(s_k))| & I(\mathbf{p}_k^i) - I(\mathbf{p}_k^o) > 0 \\ -|\nabla I(\mathbf{r}(s_k))| & I(\mathbf{p}_k^i) - I(\mathbf{p}_k^o) \leq 0 \end{cases} \quad (4)$$

where $I(\mathbf{p}_k^i)$ and $I(\mathbf{p}_k^o)$ are grayscale values of the inside and outside of the k'th contour point. Hence, by judging the sign of the intensity difference, the proposed fitting function can subtract the gradient magnitude values of the contour pixel that lies on the neighboring object boundary from the accumulated value, to avoid detections of other objects from background.

6.5 Individual contour extraction: experimental Results

Using the VSN architecture aforementioned in section 3, we tested the proposed contour segmentation method with two sequences of images in each of which the moving target of interest passes by another adjacent static object. The sensors switch working by turn. Only event sensors selected by fault tolerance method send images to CHs. After receiving several image sequences from sensors, CH will take the responsibility to execute the proposed algorithm to process the image sequence. The test data are prepared to reveal the capability of the proposed algorithm in finding an accurate boundary among similar objects nearby if there are any existing. To generate the results, we construct a B-spline contour with 8 control points and select 20 initial solutions for each 30x30 window.

Fig. 3 (a) and (b) are tracking results with contour marked as red for 2 image frame sequences of a moving car in the Region of Interest (ROI). Each 3 images of (a) and (b) are taken by 3 event sensors selected by proposed fault tolerant sensor collaboration method respectively. As shown in Fig. 3, the proposed algorithm can successfully track the moving target of interest under unfavorable conditions, for example, with some other background objects nearby. Although we tested this algorithm only in case of single target, it is also applicable to multiple targets case.

Table 1 lists parts of numerical results of the segmentation algorithm for each image of the test set. FPE (False Positive Error) is the percent of area reported as a target by the algorithm, but not by manual segmentation. FNE (False Negative Error) is the percent of area reported by manual segmentation, but not by the algorithm. Similarity and dissimilarity indices [16], which show the amounts of agreement and disagreement between the area of the algorithm and manual segmentation, are computed by:

$$S_{agr} = 2 \frac{A_{man} \cap A_{alg}}{A_{man} + A_{alg}}, S_{dis} = 2 \frac{A_{man} \cup A_{alg} - A_{man} \cap A_{alg}}{A_{man} + A_{alg}} \quad (5)$$

In this expression, A_{man} and A_{alg} are the sets of pixels classified as measurement area of manual segmentation and the algorithm, respectively. These indices are calculated for validation on every image along the movement of target. Values computed are shown in Table 1 and we conclude that proposed method for segmentation isolates individual region of target successfully.

Table 6.1: Segmentation results for 8 images

Image Number	FPE [%]	FNE [%]	S_{agr}	S_{dis}
1	3.87	3.45	0.965	0.161
2	7.76	4.49	0.947	0.122
3	6.46	8.47	0.896	0.218
4	2.81	8.29	0.909	0.198
5	2.47	7.89	0.933	0.124
6	4.21	6.79	0.935	0.089
7	4.67	1.42	0.975	0.057
8	3.73	6.57	0.955	0.122



(a)



(b)

Figure 6.3: Tracking results of the proposed method for a sequence of interesting target images (a) tracking result for the 1st original image frame sequence of a moving car in one area (b) tracking result for the 2nd original image frame sequence of the moving car in another area

As the segmentation is performed piece by piece, in contrast with the result of proposed method, mal-fitting error contained in results of existing classic contour

extraction method increases. Figure 4 shows the test result in terms of the similarity index, Sagr along with the sequence of pictures. The plots are made using the proposed segmentation algorithm, and B-spline snake respectively. It shows that B-spline snake algorithm fails when the number of pictures containing adjacent objects increases while the proposed algorithm detects the accurate contours.

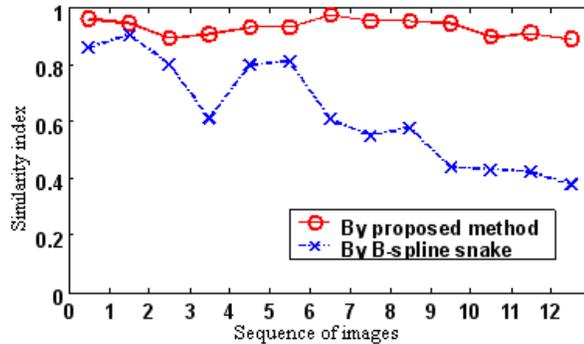


Figure 6.4: Comparison between proposed method and B-spline snake on the similarity index values at each picture

We also compared with the experimental results of a wide area target tracking system in a VSN proposed in [6] as shown in Table 2, which has similar assumptions with ours. Their algorithm success rate was defined versus a manually tracked ground truth. The trial data consisted of several data sets: the deliberate running of an electric cart along various paths through the sensor network, the recording of full-sized vehicles traveling through the network, and the recording and tracking of pedestrians in the areas which adjacent vehicle and pedestrian traffic. The experiments involved simultaneous calculations on five video streams, and ran on three separate computers. Comparing our results shown in Table 1 with the results of [6] shown in Table 2, we can see clearly the significantly higher success rate of our algorithm which is represented by Sagr. We didn't compare our results with any other tracking approaches which are based on common WSN equipped with only acoustic sensors or temperature sensors, although many of their algorithms show successful tracking results. The reason is our totally different assumptions. Their approaches work well in ideal environment without much interference or noise. In case of real environment, however, the noise or uncertainty always has negative influences on target tracking results. With visual sensors in addition to other types of sensors, we can get richer information from the target..

Table 6.2: Experiment results of graph matching in visual sensor network platform [6]

<i>Data Set</i>	<i>#Nodes Entered</i>	<i>Mean Platoon Size</i>	<i>#Successful Reident.</i>	<i>Success Rate%</i>
Electric Cart	18	1.0	11	61%
Full-size Vehicles	27	1.3	14	52%
Pedestrians	33	2.8	19	58%

6.6 Conclusions and future work

In VSN environment, fault tolerant sensor collaboration in target tracking process by a low computation cost method has been considered. A B-spline contour fitting approach based on GA for efficient contour extraction is used as inter-scene vision method to detect the target. Our proposed method produces accurate detection and tracking especially when the interesting target is closely located beside other objects. Compared with the existing classic contour extraction method B-spline and graph matching based on VSNs, our method shows significant improvement in terms of success rate of target detection and tracking. Although we have used target tracking as an example to carry out the discussions, the method of individual contour extraction is quite general.

Our future work includes accurate target position identification, image data compression and more flexible architectural design.

REFERENCES

- [1] Rahul Gupta and Samir R. Da, "Tracking Moving Targets in a Smart Sensor Network", *Proc. of Vehicular Technology Conference*, Vol.5, 2003, pp. 3035-3039.
- [2] D. Li, K. D. Wong, Y. H. Hu, and A. M. Sayeed, "Detection, Classification and Tracking of Targets", *IEEE Signal Processing Magazine*, Vol. 19, 2002, pp. 17-29
- [3] F. Maainerie, "Data Fusion and Tracking Using HMMs in a Distributed Sensor Network", *IEEE Trans on Aerospace and Electronics System*. Vol. 33, No. 1, 1997, pp. 11 - 28.
- [4] D. Reid, "An Algorithm for Tracking Multiple Targets", *IEEE Transactions on Automatic Control*, Vol, 24, No. 1 ,1979, pp. 843- 854.
- [5] S. Balasubramanian, I. Elangovan, S. K. Jayaweera and K. R. Namuduri, "Distributed and Collaborative Tracking for Energy-Constrained Ad-hoc Wireless Sensor Networks", *WCNC/ IEEE Communications Society*, 2004, pp. 1732-1737.

- [6] G. Kogut and M. Trivedi, "A Wide Area Tracking System for Vision Sensor Networks", *9th World Congress on Intelligent Transport Systems*, Chicalgo, Illinois, 2002.
- [7] T. Clouqueur, K.K. Saluja, and P. Ramanathan, "Fault Tolerance in Collaborative Sensor Networks for Target Detection", *IEEE Transactions on Computers*, Vol. 53, No. 3, 2004, pp. 320-333.
- [8] Min Ding, Dechang Chen, Andrew Thaeler and Xiuzhen Cheng, "Fault-Tolerant Target Detection in Sensor Networks", *Wireless Communications and Networking Conference*, IEEE Vol. 4, 2005, pp. 2362-2368.
- [9] Wendi B. Heinzelman, Anantha P. Chandrakasan, and Hari Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4, 2002, pp. 660-670.
- [10] Wu Xiaoling, Shu Lei, Yang Jie, Xu Hui, Jinsung Cho and Sungyoung Lee, "Swarm Based Sensor Deployment Optimization in Ad hoc Sensor Networks", *ICESS/ LNCS (SCIE)*, Springer, 2005, pp. 533-541.
- [11] Mohan M. Trivedi, Ivana Mikic, and Greg Kogut, "Distributed Video Networks for Incident Detection and Management", *Proceedings of IEEE ITSC*, 2000, pp. 155-160.
- [12] K. Obraczka, R. Manduchi, and J. J. Garcia-Luna-Aveces, "Managing the information flow in visual sensor networks", *Int. Symp. Wireless Personal Multimedia Communications*. Vol. 3, 2002, pp. 1177-1181.
- [13] Shinichi Masuda and Tetsuo Hattori, "Flexibly Configurable Multivision Remote Monitoring System", *The 47th IEEE International Midwest Symposium on Circuits and Systems*, 2004, pp. 301-304.
- [14] Greg T. Kogut, Mohan M. Trivedi, "Real-time wide area tracking: Hardware and software infrastructure", *Proceedings of the IEEE 5th International Conference on Intelligent Transportation Systems*, 2002, pp. 587-593.
- [15] L. A. MacEachern, T. Manku, "Genetic algorithms for active contour optimization", *IEEE Int. Sym. for Circuits and Systems*, 1998, pp. 229-232.
- [16] Hoon Heo, M. Julius Hossain, JeongHeon Lee, OkSam Chae, "Visualization of Tooth for 3-D Simulation", *Asia Simulation Conference, Proceedings of LNCS (SCIE)*. Vol.3398, 2004, pp. 645-684.

IMPACT OF PRACTICAL MODELS ON POWER AWARE
BROADCAST PROTOCOLS FOR WIRELESS AD HOC AND SENSOR
NETWORKS

7.1 Introduction

Wireless ad hoc and sensor networks have emerged recently because of their potential applications in various situations such as battlefield, emergency rescue, and conference environments [1-4]. Ad hoc and sensor networks are without a fixed infrastructure; communications take place over a wireless channel, where each node has the ability to communicate with others in the neighborhood, determined by the transmission range. In such network, broadcast is a frequently required operation needed for route discovery, information dissemination, publishing services, data gathering, task distribution, alarming, time synchronization, and other operations. In a broadcasting task, a message is to be sent from one node to all the other ones in the network. Since ad hoc and sensor networks are power constrained, the most important design criterion is energy and computation conservation, broadcast is normally completed by multi-hop forwarding. There exist a lot of power aware broadcast protocols and their proposals are as following: first set up broadcast tree, and then at each transmission the transmission nodes will adjust their transmission radius to the distance between trans-mission nodes and relay nodes.

The existing power aware broadcast protocols for wireless ad hoc and sensor networks assume the impractical model where two nodes can communicate if and only if they exist within their transmission radius. In this paper, we take practical models into consideration. For physical layer, we employ a universal and widely-used statistic shadowing model, where nodes can only indefinitely communicate near the edge of the communication range. For MAC layer, we consider two models: EER (end-to-end retransmission without acknowledgement) and HHR (hop-by-hop retransmission with acknowledgement). In addition, power aware broadcast protocols in networks with omni-antennas and networks with directional antennas are dealt with separately. Based on above practical models, we improve the reception probability function proposed in [7] and analyze how to choose the transmission radius between transmission nodes and relay nodes. We show how the practical physical layer and MAC layer impact the selection of transmission radius in power aware broadcast protocols and present the trade off between maximizing probability of delivery and minimizing energy consumption

in the selection of transmission radius. From our analysis, we have derived the optimal transmission range. The results presented in this paper are expected to improve the performance of power aware broadcast protocols in practical environments.

The remainder of the paper is organized as follows: Section 2 presents related work and offers some critical comments. In Section 3, we introduce our system model, including practical physical layer and MAC layer protocol model. In Section 4 we show the impact of practical physical layer on packet reception and energy consumption, and also present the improved approximation reception probability model and expected energy consumption. Section 5 presents the impact of practical models on power aware broadcast protocols focused on the selection of transmission radius. In Section 6, we present our conclusions and future work.

7.2 Introduction

In wireless ad hoc and sensor networks, the most important design criterion is energy and computation conservation since nodes have limited resources. Except reducing the number of needed emissions, radius adjustment is a good way to further reduce the energy consumption. For example, the well-known centralized algorithm is a greedy heuristics called BIP (Broadcast Incremental Power) [5]. It is a variant of the Prim's algorithm that takes advantage of the broadcast nature of wireless transmissions. Basically, a broadcast tree is computed from a source node by adding nodes one at a time. At each step, the less expensive action to add a node is selected, either by increasing the radius of an already transmitting node, or by creating a new emission from a passive one.

Our work has been inspired by recent research work made in [6-9]. Mineo Takai, et al [6] focused on the effects of physical layer modeling on the performance evaluation of higher layer protocols, and have demonstrated the importance of the physical layer modeling even if the evaluated protocols do not directly interact with the physical layer. The set of relevant factors at the physical layer includes signal reception, path loss, fading, interference and noise computation, and preamble length. I. Stojmenovic, et al [7-9] presented guidelines on how to design routing and broadcasting in ad hoc and sensor networks taking physical layer impact into consideration. They apply the log normal shadow fading model to represent a realistic physical layer to derive the approximation for probability $p(d)$ of receiving a packet successfully as a function of distance d between two nodes. Since successful reception is a random variable related to distance d , they redefine

the transmission radius r as the distance at which $p(r) = 0.5$. They proposed several localized routing schemes for the case when position of destination is known, optimizing expected hop count (for hop by hop acknowledgement), or maximizing the probability of delivery (when no acknowledgements are sent). They considered localized power aware routing schemes under realistic physical layer. Finally, they mentioned broadcasting in ad hoc network with realistic physical layer and propose new concept of dominating sets to be used in broadcasting process.

7.3 System Model

7.3.1 Physical Layer Model

Existing results in ad hoc wireless broadcasting are based on free-space or two-ray ground propagation models which represent the communication range as an ideal circle. In reality, the received power at certain distance is a random variable due to multi-path propagation effects, which is also known as fading effects. Therefore we take reality into consideration and employ shadowing model [10] as practical model which is expected to be more similar to reality.

The shadowing model consists of two parts. The first one is known as path loss model which predicts the mean received power at distance d , denoted by $\overline{P_r(d)}$. It uses a close-in distance d_0 as a reference. $\overline{P_r(d)}$ is computed relative to $P_r(d_0)$ as follows.

$$\frac{P_r(d_0)}{\overline{P_r(d)}} = \left(\frac{d}{d_0}\right)^\beta \quad (1)$$

β is called the path loss exponent and is usually empirically determined by field measurement; $\beta = 2$ is for free space propagation. Larger values of β correspond to more obstructions and hence faster decrease in average received power as distance becomes larger. $P_r(d_0)$ can be computed from free space model. The path loss is usually measured in dB. So from Eq. (1) we have

$$\left[\frac{P_r(d)}{P_r(d_0)} \right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right). \quad (2)$$

The second part of the shadowing model reflects the variation of the received power at certain distance. It is a log-normal random variable, that is, it is of Gaussian distribution if measured in dB. The overall shadowing model is represented by

$$\left[\frac{P_r(d)}{P_r(d_0)} \right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) + X_{dB}, \quad (3)$$

where X_{dB} is a Gaussian random variable with zero mean and standard deviation σ_{dB} . σ_{dB} is called the shadowing deviation, and is also obtained by measurement. Eq. (3) is also known as a log-normal shadowing model.

The shadowing model extends the ideal circle model to a richer statistical model; nodes can only probabilistically communicate near the edge of the communication range.

7.3.2 MAC Layer Protocol Model

In this section, we introduce two kinds of MAC layer protocols: HHR (hop-by-hop retransmission) protocol where the sender of a packet requires an acknowledgement from receiver and EER (end-to-end retransmission) protocol where the sender of a packet does not.

In EER case, the sender sends a packet and the receiver may or may not receive the packet which depends on the reception probability. For HHR case, we employ a MAC layer communication protocol between two nodes proposed in [7-9]. After receiving any packet from sender, the receiver sends u acknowledgements. If the sender does not receive any acknowledgement, it will retransmit the packet. They also derive the expected number of messages in this protocol as measure of hop count between two nodes. The count includes transmissions by sender and acknowledgments by receiver. They assume both the acknowledgement and data packets are of the same length.

Let S and A be the sender and receiver nodes respectively, and let $|SA| = d$ be the distance between them. Probability that A receives the packet from S is $p(d)$. Probability that S receives one particular packet from A is $p(d)$ and the probability that it does not receive the packet is $1 - p(d)$. Therefore, the probability that S does not receive any of the u acknowledgements is $(1 - p(d))^u$. Thus, the probability that S receives at least one of u acknowledgements from A is $1 - (1 - p(d))^u$. Therefore, $p(d)(1 - (1 - p(d))^u)$ is the probability that S receives acknowledgement after sending a packet and therefore stops transmitting further packets. Thus, the expected number of packets at S is $1/[p(d)(1 - (1 - p(d))^u)]$. Each of these packets is received at A with probability $p(d)$. If received correctly, it generates u acknowledgements. The total expected number of acknowledgements sent by A is then $u p(d) / [p(d)(1 - (1 - p(d))^u)] = u / (1 - (1 - p(d))^u)$. The total expected hop count between two nodes at distance d is then $1/[p(d)(1 - (1 - p(d))^u)] + u / (1 - (1 - p(d))^u)$.

7.4 Impact of Practical Models on Packet Reception and Energy Consumption

7.4.1 Reception Probability Model

In shadowing model, nodes can only probabilistically communicate near the edge of the communication range. I. Stojmenovic, et al [7-9] derives the approximation for probability of receiving a packet successfully as a function of distance d between two nodes. The model is having in mind packet length $L = 120$ and an error within 4% $p(r, d) = (1 - (d/r)^{2\beta})/2$ for $d < r$ and $((2r - d)/r)^{2\beta}/2$ for all other d , where β is the power attenuation factor, with fixed value between 2 and 6, r is transmission radius with $p(r, d=r) = 0.5$ and $d < 2r$.

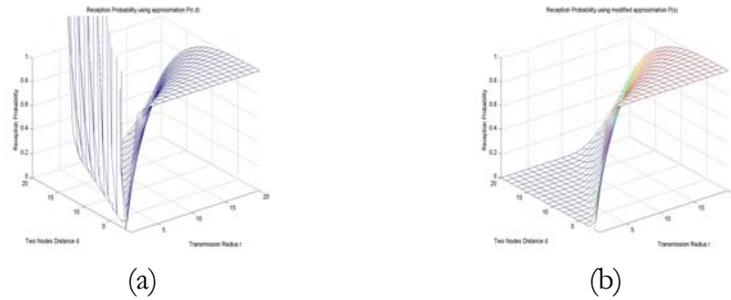


Figure 7.1: Reception probability with approximation and modified approximation $p(r, d)$

Fig. 7.1(a) shows the reception probability with approximation $p(r, d)$ when β is 2. From Fig. 7.1(a), we can see there are some error results since probability value cannot be larger than 1. The following shows our precise analysis:

$$p(r, d) = \begin{cases} 1 - \frac{(d/r)^{2\beta}}{2} & 0 \leq d < r \\ \frac{(2r-d)^{2\beta}}{2} & d \geq r \end{cases} \quad 0 \leq p \leq 1 > \begin{cases} 0 \leq 1 - \frac{(d/r)^{2\beta}}{2} \leq 1 & (0 \leq d < r) \\ 0 \leq \frac{(2r-d)^{2\beta}}{2} \leq 1 & (d \geq r) \end{cases}$$

$$\begin{cases} 0 \leq d \leq 2^{\frac{1}{2\beta}} r & (0 \leq d < r) \\ (2 - 2^{\frac{1}{2\beta}}) r \leq d \leq (2 + 2^{\frac{1}{2\beta}}) r & (d \geq r) \end{cases} \Rightarrow \begin{cases} 0 \leq d < r \\ r \leq d \leq (2 + 2^{\frac{1}{2\beta}}) r \end{cases}$$

While when d increases to $2r$, the probability has been zero which means the distance between two nodes has been too far, therefore d should be less than $2r$. At last, the modified probability model is

$$p(r, d) = \begin{cases} 1 - \frac{(\frac{d}{r})^{2\beta}}{2} & 0 \leq d < r \\ \frac{(\frac{2r-d}{r})^{2\beta}}{2} & r \leq d \leq 2r \\ 0 & \text{others} \end{cases}$$

The figure of our modified approximation $p(r, d)$ when β is 2 is shown in Fig. 7.1(b).

7.4.2 Expected Energy Consumption

Assume now that two nodes are at distance d , but a packet is sent with transmission radius r ; let E represent energy for processing signals at both transmitter and receiver nodes. The exact transmission power is then r^β multiplied by a constant, which is assumed to be 1 for simplicity. Therefore the energy needed by sending node is $E + r^\beta$, while energy at receiving node is E , for a combined energy $2E + r^\beta$. The reception probability at distance d is $p(d) = p(rd/r) = g(d/r)$, where we defined $g(y) = p(ry)$.

In EER case, the sender sends a packet and the receiver may or may not receive the packet, which depends on the probability of receiving. Therefore, the expected energy consumption is $(2E + r^\beta)g(d/r) = (2E + d^\beta(r/d)^\beta)g(d/r)$.

In HHR case, a message is retransmitted between two nodes until it is received and acknowledged correctly; after receiving any packet from the sender, the receiver sends u acknowledgements. Transmissions and acknowledgements in general do not need to be done with the same transmission powers. However, since they use the same probability function, we can argue that the optimal power is achieved when both of them use the same power. Then, because the expected number of transmitted packets (for $u = 1$) is $1/g^2(d/r)$ and the expected number of acknowledgements is $1/g(d/r)$, the total expected energy consumption is $(2E + r^\beta)(1/g(d/r) + 1/g^2(d/r))$, which is a function of one variable that needs to be optimized for r as function of d . The formula is as following $(2E + d^\beta(r/d)^\beta)(1/g(d/r) + 1/g^2(d/r))$.

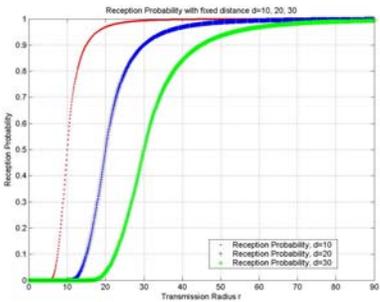
7.5 Impact of Practical Models on Power Aware Broadcast Protocols

For broadcast with practical models, first we set up broadcast tree using power aware broadcast protocols under impractical model; and then, choose the optimal transmission radius for every retransmission. As for the metric to decide the optimal transmission radius, there exists a trade-off or negotiation between maximizing probability of delivery and minimizing energy consumption. We

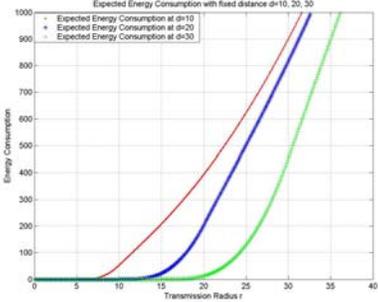
propose the following rules: for broadcasting in wireless network with omni-antennas, minimizing energy consumption is the primary metric; otherwise, for network with directional antennas, maximizing probability of delivery will be the primary metric, since transmission coverage overlapping is much fewer than that in networks with omni-antennas.

7.5.1 EER Case

In EER case, a sender sends a packet and a receiver may or may not receive the packet which depends on the reception probability. The reception probability function is $p(r,d)=(1-(d/r)^{2\beta}/2)$ for $d < r$, $((2r-d)/r)^{2\beta}/2$ for $r < d < 2r$, and 0 for all the other d . For network with directional antennas, since maximizing probability of delivery is our primary metric, at least we have to guarantee the reception probability no less than 0.5; however if the reception probability is near 1, the energy consumption will be too high. Therefore, we choose [0.5 0.9] as the acceptable reception probability scope. From Fig. 1 we can find that if $r > d$, the scope of reception probability is [0.5, 1]; otherwise, if $r < d$, reception probability will be less than 0.5. Since we should guarantee the reception probability no less than 0.5, we will only use $p(r,d)=(1-(d/r)^{2\beta}/2)$ for $d < r$. For any value of $\beta, 2 \leq \beta \leq 6$, if we want to get the relationship of d and r ($r > d$) for certain reception probability α , we can set up the formula as $1-(d/r)^{2\beta}/2 = \alpha$, then we get $r = [2(1-\alpha)]^{-1/2\beta} d$. Therefore, in order for reception probability to be [0.5 0.9], the transmission radius should be $[d (1/5)^{-1/2\beta} d]$. We can verify it through Fig. 2, where $\beta=2, d=10, 20$ and 30 . According to our proposal, we can choose the transmission radius in the scope of [10 15], [20 30] and [30 45] respectively. In Fig. 2(a), the according reception probability is in the scope of [0.5 0.9]; in Fig. 2(b), the according expected energy consumption is in the scope of [53 208], [203 817] and [453 1830] respectively.



(a) Reception probability



(b) Expected energy consumption

Figure 7.2: Reception probability and expected energy consumption with fixed distance d

For network with omni-antennas, minimizing expected energy consumption is primary metric. We know as transmission r increases, the expected energy consumption will also increase. Therefore, we want to choose the transmission radius r value as small as possible. Whereas, even minimizing energy consumption is the primary metric, we still cannot neglect the reception probability. According our proposal above, which is selecting r in the scope $[d (1/5)^{-1/2\beta} d]$, and getting the reception probability scope $[0.5 0.9]$, by guaranteeing reception probability not less than 50%, we decide to choose d as the transmission radius r .

7.5.2 HHE Case

In HHR case, a message is retransmitted between two nodes until it is received and acknowledged correctly; after receiving any packet from sender, the receiver sends u acknowledgements. Considering the characteristic of MAC layer in HHR case, it's better to be employed in networks with directional antennas, which represent one to one transmission model. In addition, we can find the MAC layer has already guaranteed successful reception, therefore our research moves to minimizing the expected hop number and expected energy consumption between two nodes.

According to the MAC layer protocol in HHR case, $p(d)(1-(1-p(d))^u)$ is the probability that sender S receives acknowledgement after sending a packet and therefore stops transmitting further packets. Each of these packets is received at A with probability $p(d)$. When u equals 1, reception probability at sender S and receiver A is respectively $p^2(d)$ and $p(d)$, that is $g^2(d/r)$ and $g(d/r)$. Since the expected packets number is respectively $1/g^2(d/r)$ and $1/g(d/r)$, our work is transferred to maximize the reception probability at sender S and receiver A .

For any value of β , $2 \leq \beta \leq 6$, for receiver A , the relationship of d and r ($r > d$) for certain reception probability α is $r = [2(1-\alpha)]^{-1/2\beta} d$, then in order for reception probability to be $[0.5 0.9]$, the transmission radius should be $[d (1/5)^{-1/2\beta} d]$; however, for sender S , the relationship of d and r ($r > d$) for certain reception probability α is $r = [2(1-\alpha^{1/2})]^{-1/2\beta} d$, then in order for reception probability to be $[0.5 0.9]$, the transmission radius should be $[[2(1-(0.5)^{1/2})]^{-1/2\beta} d \quad [2(1-(0.9)^{1/2})]^{-1/2\beta} d]$. Therefore considering the reception probability of both sender S and receiver A , our proposal can be extended as the following: in HHR case, we choose r from the scope of $[[2(1-(0.5)^{1/2})]^{-1/2\beta} d \quad (1/5)^{-1/2\beta} d]$, where for sender S the scope of

reception probability is [0.5 0.9) and for receiver \mathcal{A} the scope of reception probability is within (0.5 0.9]. We can verify it through Fig. 3, where $\beta=2$, $d=10$, 20 and 30. The reception probability at sender \mathcal{S} and at receiver \mathcal{A} with fixed distance d when β is 2 is showed in Figure 3.

According to our proposal, we can choose the transmission radius in the scope of [11.4, 15], [22.9 30] and [34.3 45] respectively. In Fig. 3, for sender \mathcal{S} , the scope of reception probability is [0.5 0.8] and for receiver \mathcal{A} , the scope of reception probability is within [0.7 0.9].

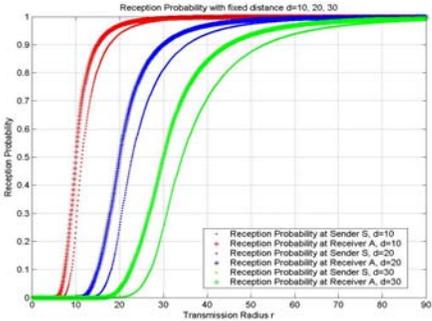
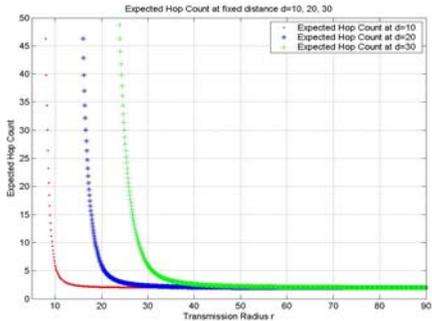
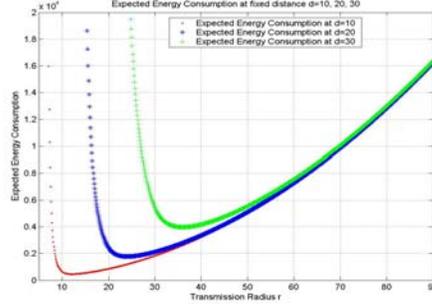


Figure 7.3: Reception probability at sender \mathcal{S} with fixed distance $d=10, 20, 30$

In HHR case, because of the characteristic of MAC layer, the number of transmission between two nodes is more than one, therefore expected hop count and expected energy consumption will be higher than that in EER case. Fig. 4 shows the total expected hop count and energy consumption including sender \mathcal{S} and receiver \mathcal{A} when β is 2.



(a) Total expected hop count



(b) Total expected energy consumption

Figure 7.4: Total expected hop count and energy consumption with fixed distance d

We can verify whether our proposal of choosing r from the scope of $[[2(1-(0.5)^{1/2})]^{-1/2\beta}d (1/5)^{-1/2\beta}d]$ is reasonable or not. The total expected hop count and energy consumption with fixed distance $d=10, 20, 30$ when β is 2 is showed in Figure 4. According to our proposal, we can choose the transmission radius in the scope of $[11.4 15]$, $[22.9 30]$ and $[34.3 45]$ respectively. Fig. 4(a) shows that if the transmission radius r is not less than the distance 10, 20 and 30 respectively, expected hop count will be less than 5 and also at last decrease to a constant number. Fig. 4(b) shows that the expected energy consumption can get minimum value when r is around 11.4, 22.9 and 34.3 respectively; whereas if r is larger than those values, the expected energy consumption will increase. Therefore, even if r is larger than 15, 30 and 45 respectively, we can get the minimum expected hop count, but because the expected energy consumption will be larger, so we still cannot choose r larger than 15, 30 and 45 respectively. In a word, our proposal for HHR case is to choose the transmission radius r in the scope of $[[2(1-(0.5)^{1/2})]^{-1/2\beta}d (1/5)^{-1/2\beta}d]$, which can get optimal performance at expected hop count and energy consumption.

7.6 Conclusions

To the best of our knowledge, this is the first work that considers the impact of practical physical layer and MAC layer model on power aware broadcast protocols. We investigated power aware broadcast protocols with and without acknowledgements and presented the trade off between maximizing probability of delivery and minimizing energy consumption for ad hoc wireless networks with practical models. We show how the practical physical layer and MAC layer impact the selection of transmission radius in power aware broadcast protocols. In EER case, for network with omni-antennas, minimizing energy consumption is the primary metric, and by guaranteeing reception probability no less than 50%, we decide to choose the distance d as transmission radius, where d is the distance

between transmission node and relay node; in network with directional antenna, we propose to choose the transmission radius in the scope of $[d (1/5)^{-1/2\beta} d]$ to maximize the probability of delivery. In HHR case, the MAC layer protocol is not suitable to one-to-all communication; therefore we only consider networks with directional antennas. Since the MAC layer has already guaranteed successful reception, our research moves to minimize the expected hop number and expected energy consumption between two nodes. For networks with directional antennas we propose to choose the transmission radius in the scope of $[[2(1-(0.5)^{1/2})]^{-1/2\beta} d (1/5)^{-1/2\beta} d]$, which can get optimal performance at expected hop count and energy consumption. Currently, we are designing new power aware broadcast protocols based on our analysis.

REFERENCES

- [1] S. Basagni, M. Conti, S. Giordano, I. Stojmenovic (eds.), “Mobile Ad Hoc Networking”, IEEE Press/Wiley, July 2004.
- [2] J. M. Kahn, R. H. Katz, and K. S. J. Pister, “Next century challenges: Mobile networking for ‘smart dust’”, International Conference on Mobile Computing and Networking (MOBICOM), 1999, pp. 271–278.
- [3] G. Giordano, I. Stojmenovic (ed.): Mobile Ad Hoc Networks. Handbook of Wireless Networks and Mobile Computing, Wiley, 2002, pp. 325-346.
- [4] G.J. Pottie and W.J. Kaiser, “Wireless integrated network sensors”, Communications of the ACM, 2000, vol. 43, no. 5, pp. 551–558.
- [5] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, “On the construction of energy-efficient broadcast and multicast trees in wireless networks”, Proc. IEEE INFOCOM, March 2000, pp. 585-594.
- [6] Mineo Takai, Jay Martin and Rajive Bagrodia, “Effects of wireless physical layer modeling in mobile ad hoc networks”, Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, 2001, pp. 87-94.
- [7] I. Stojmenovic, A. Nayak, J. Kuruvila, F. Ovalle-Martinez and E. Villanueva-Pena, “Physical layer impact on the design of routing and broadcasting protocols in ad hoc and sensor networks”, Journal of Computer Communications (Elsevier), Special issue on Performance Issues of Wireless LANs, PANs, and Ad Hoc Networks, Dec 2004, to appear.
- [8] Johnson Kuruvila, Amiya Nayak and Ivan Stojmenovic, “Hop count optimal position based packet routing algorithms for ad hoc wireless networks with a realistic physical layer”, IEEE Journal on selected areas in communications, June 2005, vol.23.
- [9] J. Kuruvila, A. Nayak and I. Stojmenovic, “Greedy localized routing for maximizing probability of delivery in wireless ad hoc networks with a realistic physical layer”, Algorithms for Wireless And mobile Networks (A_SWAN)

Personal, Sensor, Ad-hoc, Cellular Workshop, at Mobiquitous, Boston, August 2004, pp. 22-26.

[10] Network Simulator: <http://www.isi.edu/nsnam/ns/>.