Optimizing Coverage in a K-Covered and Connected Sensor Network Using Genetic Algorithms

Kasım Sinan YILDIRIM
Ege University
Department of Computer Engineering
35100 Bornova-Izmir
TÜRKİYE
sinan.yildirim@ege.edu.tr

Tahir Emre KALAYCI
Ege University
Department of Computer Engineering
35100 Bornova-Izmir
TÜRKİYE
tahir.kalayci@ege.edu.tr

Aybars UĞUR
Ege University
Department of Computer Engineering
35100 Bornova-Izmir
TÜRKİYE
aybars.ugur@ege.edu.tr

Abstract: Sensing coverage of a sensor network characterizes how well an area is monitored or tracked by sensors. Connectivity is an important requirement that shows how nodes in a sensor network can effectively communicate. Some hotspot areas in the network are more important than other areas and need to be covered by more sensors. We are interested in an initial deployment strategy that maximizes the coverage area of wireless sensor network while preserving connectivity between nodes provided that all given hotspot regions are covered by at least k sensors. We propose a genetic algorithm based solution to find an optimal sensor node distribution. Experimental results are presented to evaluate our algorithm.

Key–Words: Wireless Sensor Networks, Genetic Algorithm, Coverage, Connectivity, k-Coverage

1 Introduction

Sensor networks are dense wireless networks of small, low-cost sensors, which collect and disseminate environmental data. Wireless sensor networks facilitate monitoring and controlling of physical environments from remote locations with better accuracy. They have applications in a variety of fields such as environmental monitoring, military purposes and gathering sensing information in inhospitable locations [1].

Coverage is one of the fundamental problems in sensor networks which sensing coverage characterizes the monitoring quality provided by a sensor network on a designated region and reflects how well a sensor network is monitored or tracked by sensors [3]. Consequently, coverage can be considered as the measure of quality of the service of a sensor network [11]. While some applications may only require that every location in a region be monitored by one node, other applications require that some areas in the network are more important than other areas and need to be covered by more sensors. These important regions are called hotspots[3]. The coverage requirement on hotspot regions also depends on the number of faults that must be tolerated. Another important point to consider in sensor networks is providing connectivity between sensor nodes. Without connectivity, nodes may not be able to coordinate effectively or transmit data back to base stations. Thus, combination of connectivity and coverage is an important concept in sensor networks[4].

In a large scale sensor network that the sensor nodes are random scattered, some redundant sensor nodes are needed to make sure a satisfying coverage on the sensing area. In sensor networks that sensor nodes can be set specially, we can reduce not only the redundancy nodes, routing request and maintenance overhead, power consuming but also expend the network's sensing range. So how to get the optimized node distribution is an important problem in the wireless sensor networks [7].

A genetic algorithm (GA) is a search technique used in computer science to find approximate solutions to combinatorial optimization problems. Some of the GA's merits are that it can be easily developed because it does not require detailed knowledge about the problem, it can search globally, and it can adapt to the changing conditions in the problem[2]. It is difficult to derive a solution for finding a better coverage for a k-covered and connected sensor network deployment. The space of all feasible solutions can be quite large. GA can efficiently be used to search an optimal solution to this problem as a powerful technique.

The main contribution of this paper to be the first application employing genetic algorithms that satisfies three conditions for finding an optimal sensor placement :

- maximizes the coverage area of a sensor node distribution
- all of the given hotspot areas are covered by at

least k sensors (k-covered)

- maintains connectivity between sensor nodes

This paper is organized as follows. Section 2 describes the related work with respect to coverage problem in sensor networks. Section 3 contains constraints and description of the problem. Section 4 presents genetic algorithm which is used to optimize coverage in a connected and k-covered sensor network. Section 5 shows experimental results and Section 6 is the conclusion.

2 Related Work

Recently, many researchers have been investigating and developing techniques for achieving an optimal sensor distribution that maximizes the coverage of the sensing area and preserving sensor node connectivity.

Megerian et. al. [5] presents an optimal polynomial time worst and average case algorithm for coverage calculation by combining computational geometry and graph theoretic techniques (Voronoi diagram and graph search algorithms). Sensor coverage of the field are characterized by Maximal Breach Path and Maximal Support Path and these parameters can be used for future deployment or reconfiguration schemes for improving the overall quality of the service. A geometric analysis of the relationship between coverage and connectivity is provided in [4]. A Coverage Configuration Protocol (CCP) is presented which can provide different degrees of coverage requested by applications meanwhile maintaining communication connectivity when the sensing range of sensors is no more than half of the communication range. But both of these two work do not address k-coverage problem and do not provide an initial both k-covered and connected sensor network deployment.

Some researchers have focused on optimizing coverage of sensor network using Evolutionary Computation techniques. An optimal distribution based on Genetic Algorithm in the initial planning of sensor network and a new optimizing algorithm of sensor node distribution by utilizing node topology in sensor network are proposed in [7]. [6] presents an Integer Linear Programming formulation and evolutionary algorithm to find a configuration that maintains the coverage of the monitoring area, accomplishes the management of the network resources and minimizes energy consumption. [11] modeled the coverage problem as two sub-problems: floorplan and placement. Two sub-problems are combined into one optimization problem so that it can achieve the maximum possible coverage. Evolutionary computation techniques presented in these papers do not offer a solution to find an optimal coverage while preserving k-coverage and connectivity.

There are some other studies that target k-coverage problem in sensor networks. [3] formulate k-coverage problem as a decision problem, whose goal is to determine whether every point in the service area of the sensor network is covered by at least k sensors, where k is a predefined value. A polynomial-time algorithm is provided that checks the perimeter of the sensing range of each sensor. [9] considers the problem of selecting minimum size connected K-cover. [10] propose a heuristic algorithm for efficiently scheduling the sensors, such that monitored region can be k-covered with the purpose of maximizing the network lifetime. [8] present efficient approximation algorithm for selecting the minimum set of sensors to activate from an already deployed set of sensors. However, none of these research efforts try to find an optimal initial sensor network distribution that has k-covered hotspot areas and sensor connectivity.

3 Problem Definition

We are given an obstacle-free 2D area $A(width, height)$, N sensors with sensing radius r_s and communication radius $r_c = 2 * r_s$, h hotspot areas and a k value. We assume that hotspot areas are unit disk shaped and they all have identical radius that is equal to the sensing radius of sensors. Fig. 1 presents the input parameters of the problem.

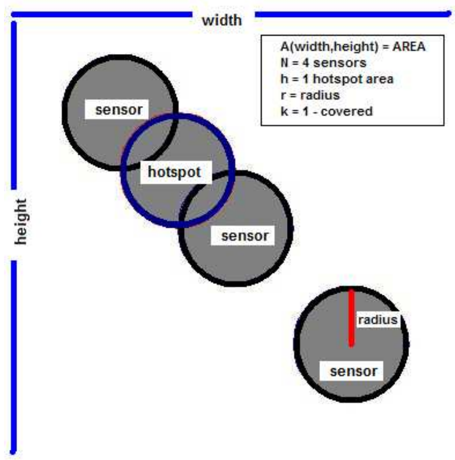


Figure 1: Example Sensor Network

We are requested to maximize total covered area of sensor network under the following constraints:

- All sensors can communicate with each other (connectivity)
- h hotpot areas must be covered by at least k sensors ($k - covered$)

- The sensing and communication ranges of all sensors are identical
- Centers of sensors must be reside in the limited area A

After defining parameters and constraints, we can summarize the problem as:

Definition 1 (Problem Definition) *Given parameters A, N, r_s, r_c, k; try to increase coverage area of the sensor network without breaking the property that all hotspot areas are k-covered and all sensors are connected*

4 Evolutionary Approach for Solving the Problem

We implemented a GA to find an optimal solution to the coverage problem. First of all, GA requires some important parameters to be defined listed as below:

- T Population size
- P_m Mutation probability
- P_c Crossover probability
- G Generation count

Next, GA is started with a random generated population. At each iteration, the best solutions from the current population are used to form a new population such that the new population is expected to be better than the old one. GA continues to run until the number of the iterations exceeds the predetermined termination criteria (total number of generated populations which given as a parameter). Outline of the implemented GA is as follows:

- **Start** Generate a random population of chromosomes
- **Loop** Repeat until termination criteria is reached
  - **Fitness** Calculate fitness and sort individuals
  - **New Population** Crossover and mutation are applied to current population to form a new population
  - **Elitism** The best individual is selected and applied to current sensor deployment
- **Solution** Return optimal solution

4.1 Chromosome Encoding

Sensors are represented using 2D coordinates ($S(x, y)$) on a 2D plane ($A(width, height)$) of the form

$$\{S(x, y) : 0 \le x \le width, 0 \le y \le height\} \quad (1)$$

Solutions (Chromosomes) are represented with a movement array

$$\begin{array}{l} C_0 = \{M_{S_0}, M_{S_1}, M_{S_2}, M_{S_3}, ..., M_{S_N}\} \\ C_1 = \{M_{S_0}, M_{S_1}, M_{S_2}, M_{S_3}, ..., M_{S_N}\} \\ C_2 = \{M_{S_0}, M_{S_1}, M_{S_2}, M_{S_3}, ..., M_{S_N}\} \\ \\ C_i = \{M_{S_0}, M_{S_1}, M_{S_2}, M_{S_3}, ..., M_{S_N}\} \\ \\ C_T = \{M_{S_0}, M_{S_1}, M_{S_2}, M_{S_3}, ..., M_{S_N}\} \end{array} \quad (2)$$

Every gene in this array represents next movement amount of corresponding sensor. Population has T chromosome and every chromosome has N sensor movement. Movement amount of sensor is

$$M_{S_i}(x_{inc}, y_{inc}) : \{-r_s * 0.01 \le x_{inc}, y_{inc} \le r_s * 0.01\} \quad (3)$$

All these representations are implemented using basic data structures which are shown in Fig. 2.

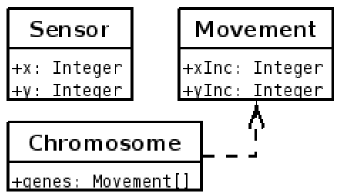


Figure 2: Data structures

4.2 Crossover, Mutation and Elitism

Crossover generates new offsprings from the best parents among the current population. With crossover probability (P_c), the best individuals are selected and paired using one point crossover (Fig. 3). A random cut point is selected on parents and genes beyond this cut point is swapped between parent chromosomes. The procured chromosomes are the children and replaced with the worst individuals in current population to form a new and hopefully better population.

After crossover is finished, with mutation probability (P_m), the individuals are mutated by replacing two randomly selected genes. After mutation and crossover, the next step is choosing the elite individual. If the best solution of current generation is better than the elite individual of all previous populations,

Figure 3: One point crossover

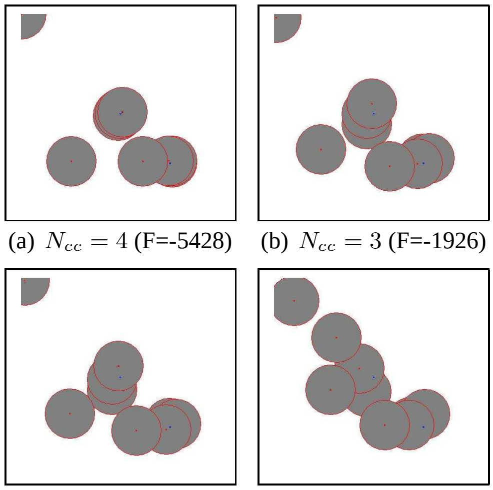
then it is saved as elite individual. At the next generation, elite individual is applied to sensor coordinates to generate the new sensor locations.

4.3 Fitness function

Fitness function helps to discern quality of individuals and to separate them to different quality groups. Our main objective is to maximize coverage area. The area of coverage is only useful to classify individuals if these individuals are totally connected and satisfy k-coverage. Otherwise we have to calculate fitness function using other considerations. Quality of every individual of the population has been calculated using three criteria.

We assume that at every iteration of the GA, all hotspots in the current sensor node distribution remains k-covered. Thus, we consider that the worst individuals are the ones that do not satisfy k-coverage property. First criteria is that, if current individual does not satisfy k-coverage property, we assign a negative fitness (defined as $-INFINITY$) to this individual. We used the polynomial-time algorithm proposed in [3] to check the perimeter of each hotspot area. If the perimeter of each hotspot area is covered by at least k sensors, then we decide that all hotspot areas are at least k-covered and when we apply the individual to the current sensor node distribution, k-coverage property is still satisfied.

If hotspots areas are k-covered then the second criteria is checking how well the sensors are connected. This check has been done as finding the number of connected components applying depth first search to the network graph. If there are more than one connected component, this means this sensor network is not connected and we assign as negative fitness value to this individual because it is not a feasible solution. In a disconnected sensor network, fitness is calculated as product of number of connected components (N_{cc}) and total of shortest distances between all connected components (D_{cc}). This formula generates distinction between similar disconnected sensor networks (Fig. 4). Fig. 4(b) has a better fitness then Fig. 4(a) since it has lower number of connected components. Fig. 4(d) and Fig. 4(c) has the same number of connected components but the total distance between connected components in the former is smaller and as a result it has a better fitness. Consequently, by assigning such fitness values to individuals, we are trying to connect a disconnected and k-covered sensor network.



(a) $N_{cc} = 4$ (F=-5428) (b) $N_{cc} = 3$ (F=-1926)

(c) $N_{cc} = 2$ (F=-510) (d) $N_{cc} = 2$ (F=-224)

Figure 4: Disconnected sensor nodes

After constructing a connected k-covered sensor network, we try to maximize its coverage. The coverage area of sensors is the last criteria for grouping individuals based on their quality and it is estimated using an image processing technique. All sensors are drawn on a 2D finite plane and after producing 2D graphics, all pixels which have different color from background are counted to estimate the coverage area (A_c)

$$A_c = \sum_{i=0}^{width} \sum_{j=0}^{height} \begin{cases} 0, & \text{if } Color_{ij} \text{ is White} \\ 1, & \text{else} \end{cases} \tag{4}$$

All these calculations are summarized at equation (5).

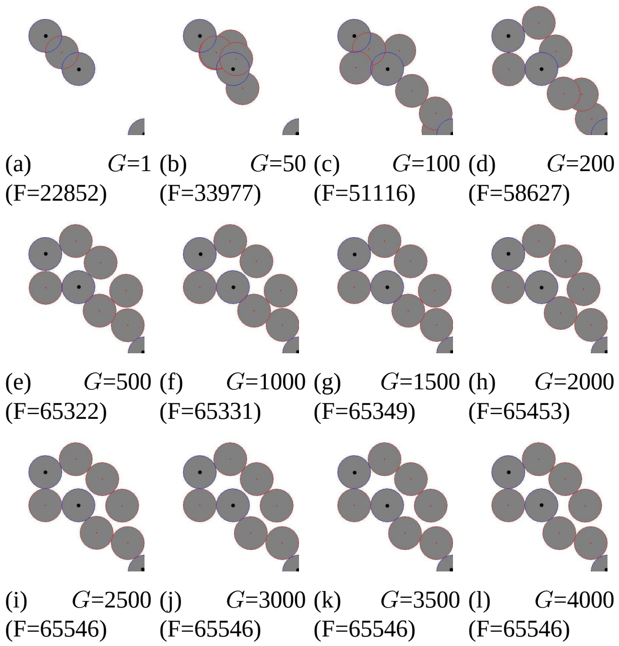
$$F(C_i) = \begin{cases} -INFINITY, Q_0 \\ -(D_{cc} * N_{cc}), Q_1 \\ A_c, Q_2 \end{cases} \tag{5}$$

Q_0=not k-covered (not feasible)
Q_1=k-covered, not connected (not feasible)
Q_2=k-covered, connected (feasible)

5 Experimental Results

We have implemented the coverage optimization problem in Java programming language using evolutionary approach. For all our experiments, we assigned the population size, the number of generation, the crossover rate and mutation rate to be 100, 4000, 0.7, and 0.2 respectively. All experiments are executed on a PC platform that has Ubuntu GNU/Linux operating system installed.

Before starting Genetic Algorithm, we build an initial node placement that has all hotspot areas k-covered. In order to achieve this, we place k sensors on every hotspot and remaining sensors are placed randomly. After we build an initial k-covered sensor graph, we start the Genetic Algorithm. Fig. 5 illustrates a sample simulation. We are given 3 hotspots ($h_1(100, 100)$, $h_2(200, 200)$,$h_3(400, 400)$), 12 sensors that have 50 pixel sensing range, coverage value k as 2 and a 400x400 square object area. The initial distribution given as a start point to the genetic algorithm is shown in Fig. 5(a) (We have $3 * 3 = 9$ sensors located at centers of hotspot areas $(100, 100), (200, 200), (300, 300)$ that makes our hotspot areas k-covered and remaining 3 sensors are placed randomly). Centers of hotspot areas are shown with small black dots. At each generation, Genetic Algorithm tries to construct a connected graph and then it maximizes the total coverage area of the sensor network. Fig. 5(l) shows the algorithm can effectively adjust node positions to get an optimal sensor node distribution which has the best network coverage satisfying connectivity and k-coverage for hotspot areas. Fig. 6 shows the change in total covered area (fitness) through the generations.



(a) G=1 (F=22852) (b) G=50 (F=33977) (c) G=100 (F=51116) (d) G=200 (F=58627)

(e) G=500 (F=65322) (f) G=1000 (F=65331) (g) G=1500 (F=65349) (h) G=2000 (F=65453)

(i) G=2500 (F=65546) (j) G=3000 (F=65546) (k) G=3500 (F=65546) (l) G=4000 (F=65546)

Figure 5: Elite individuals and increase in coverage during GA run (k=2, h=3, N=12, $A(400, 400)$)

Fig. 7 shows how is the total coverage affected when we change the k parameter and illustrates sensor distribution through 0-100th generations when k=1,2 and 3 respectively. Since genetic algorithm preserves k-coverage for all hotspot areas and does not move sensors that k-covers the hotspot area, when the coverage value is bigger the number of sensors that GA can move decreases. As shown in Fig. 7(a), when k=1 we still do not have a connected graph in the 100th generation because genetic algorithm has many sensors that can be moved and search domain is large. Fig. 7(f) shows the 100th generation when k=2 and it is connected. When k=3, we have a lower number of sensors that can move to make the graph connected. Also the sensors that are moved to connect the graph are far away from the other connected component then when k=2 as shown in Fig. 7(i).

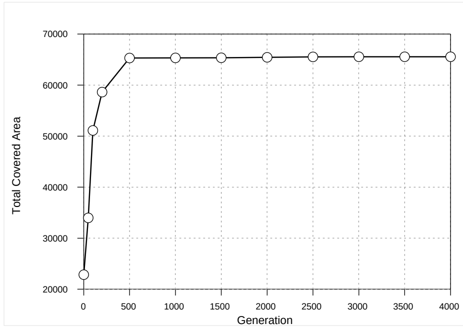
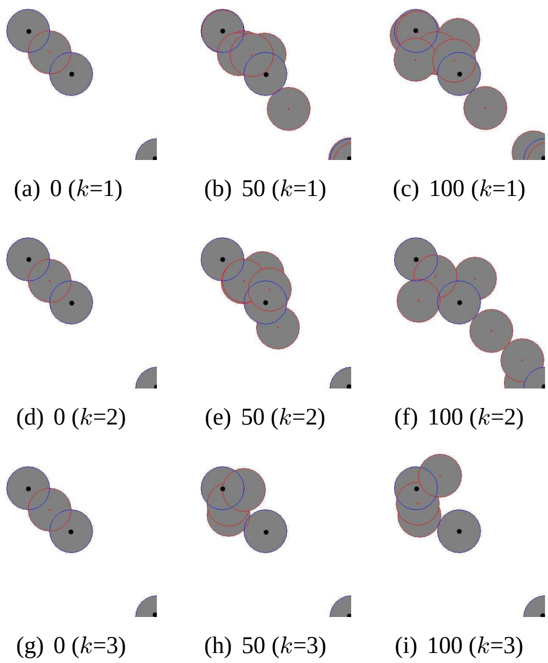


Figure 6: Total covered area change during GA run



(a) 0 (k=1) (b) 50 (k=1) (c) 100 (k=1)

(d) 0 (k=2) (e) 50 (k=2) (f) 100 (k=2)

(g) 0 (k=3) (h) 50 (k=3) (i) 100 (k=3)

Figure 7: Change in coverage with different k values

Finally, Fig. 8 shows a snapshot of genetic algorithm simulation environment. Application uses XML file for getting parameters of the problem and users can interact with graphical user interface that visualizes sensor network node distribution on each generation of GA.

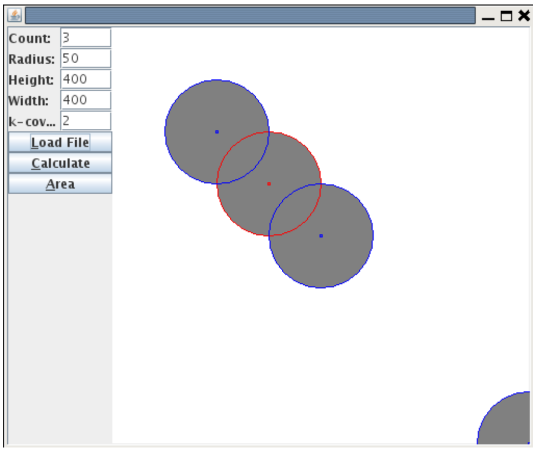


Figure 8: Genetic Algorithm Simulation Environment for Sensor Networks

Experimental results show that implemented genetic algorithm can provide an optimal sensor network distribution that maximizes total coverage area while preserving k-coverage for hotspot areas and connectivity. Implemented algorithm can also be applied to an existing k-covered but not necessarily connected sensor network graph to make it connected and to maximize its total coverage area.

6 Conclusion and Future Work

Coverage and connectivity are important metrics to characterize the quality of sensor networks. In this paper, a genetic algorithm is proposed and a specific fitness function is designed to find a sensor network distribution that has optimal coverage by preserving node connectivity and k-coverage property for hotspot areas. Experimental work shows that this algorithm can achieve optimal node distribution in a feasible time. We are studying on developing a method to find minimum number of sensors for connected k-coverage problem. We plan to continue development of simulation tool and extend its functionality to support more application areas.

Acknowledgements: Tahir Emre KALAYCI and Kasım Sinan YILDIRIM acknowledge TÜBİTAK for supporting this work through a domestic Ph.D. scholarship (2211 - Yurt İçi Doktora Burs Programı).

References:

[1] A. Bharathidasan and V.A.S. Ponduru, Sensor networks: An overview. Online. Available: http://wwwcsif.cs.ucdavis.edu bharathi/sensor/survey.pdf

[2] H. Sengoku, I. Yoshihara, A Fast TSP Solver Using GA on Java, In Proc. of the 3rd Int. Symp. on Artificial Life and Robotics (AROB-III), 1998, pp. 283–288.

[3] C. Huang, and Y. Tseng, The coverage problem in a wireless sensor network, In Procs. of the 2nd ACM international Conference on Wireless Sensor Networks and Applications (WSNA ’03), 2003, pp.115–121.

[4] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, In Proc. of the 1st international Conference on Embedded Networked Sensor Systems (SenSys ’03), 2003, pp.28-39.

[5] S. Megerian, F. Koushanfar, M. Potkonjak, M.B. Srivastava, Worst and best-case coverage in sensor networks, Transactions on Mobile Computing , vol.4, no.1, Jan.-Feb. 2005, pp.84-92

[6] F.P. Quintao, F.G. Nakamura, G.R. Mateus, Evolutionary algorithm for the dynamic coverage problem applied to wireless sensor networks design, Evolutionary Computation, 2005, vol.2, pp.1589-1596

[7] J. Zhao, Y. Wen, R. Shang and G. Wang, Optimizing Sensor Node Distribution with Genetic Algorithm in Wireless Sensor Network, Advances in Neural Networks (ISNN’04), 2004, LNCS vol. 3174, pp.242-247

[8] M. Hefeeda and M. Bagheri, Efficient k-Coverage Algorithms for Wireless Sensor Networks, Technical Report TR 2006-22, Simon Fraser University School of Computing Science, 2006

[9] Z. Zhou, S. Das, H. Gupta, Connected K-coverage problem in sensor networks, Proc. of 13th International Conference on Computer Communications and Networks, 2004. ICCCN 2004), 11-13 Oct. 2004, pp. 373-378

[10] S. Gao, C.T. Vu, and Y .Li, Sensor Scheduling for k-Coverage in Wireless Sensor Networks, 2nd International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2006), Hong Kong, China, December 13-15, 2006.

[11] S. Habib, M Safar, Sensitivity Study of Sensors’ Coverage within Wireless Sensor Networks, Proc. of 16th International Conference on Computer Communications and Networks (IC-CCN 2007), 13-16 Aug. 2007, pp.876-881