1st International Conference on Sensing Technology
November 21-23, 2005 Palmerston North, New Zealand

# Probabilistic Self-Scheduling for Coverage Configuration in Sensor Networks

Jun Lu, Lichun Bao and Tatsuya Suda
Bren School of Information and Computer Sciences,
University of California, Irvine, CA 92697, USA
{lujun,lbao,suda}@ics.uci.edu

## Abstract

Sensing coverage is a critical issue in sensor network deployments. We propose a novel scheme to maintain the sensing coverage in sensor networks, which we call CASS (Coverage-Aware Self-Scheduling). Different from the existing work on coverage maintenance, CASS probabilistically schedules a sensor's sensing activities according to the sensor's contribution to the sensing coverage. CASS is designed to allow sensors with higher coverage contribution to have more chance to be active. In this way, CASS reduces the number of active sensors to maintain certain coverage. Besides the sensing coverage, the connectivity of the network topologies is required for the purposes of communicating among sensors to collect sensing data. Therefore, we describe a generic unifying framework to incorporate different connectivity and coverage maintenance schemes. Simulations are carried out under the framework by integrating CASS with LEACH (the Low Energy Adaptive Clustering Hierarchy). Simulation results show that CASS can considerably improve the energy efficiency of a sensor network with low overhead.

**Keywords**: sensing coverage, sensor scheduling, self-organization, wireless sensor networks

## 1 Introduction

Wireless sensor networks are networks of a large number of small wireless devices, which can sense their surrounding environments and report their sensing data through wireless communications. Wireless sensor networks have emerged rapidly to provide surveillance functions in a variety of applications, such as environment monitoring and target tracking.

Wireless sensors are highly limited in their capabilities (*e.g.*, the processing, computing and communication powers) and constrained resources (*e.g.*, the storage and power capacities). For example, the crossbow mica mote MPR300CB [1] has a low-speed $4MHz$ processor equipped with only $128kB$ flash, $4kB$ SRAM and $4kB$ EEPROM. It has a maximal data rate of $40kbps$ and a transmission range of about 100 feet, using the power of two AA batteries. Due to the limited capabilities and constrained resources of wireless sensors, wireless sensors are usually deployed in high density. Dense deployment not only helps to improve a sensor network's reliability, but also extends its longevity. Moreover, wireless sensor networks are usually employed to monitor a large area, which makes manual deployment impossible. In this research, we assume that a large-scale wireless sensor network is deployed in a random manner.

Given such a randomly and densely deployed wireless sensor network, it is desirable to have sensors autonomously schedule their duty cycles according to local information of connectivity and coverage. This is considered as a self-organization problem, which has drawn intense research attention recently. Wang *et al.* [2] pointed out that the self-organization to maintain network connectivity and sensing coverage are two different issues and both are essential for wireless sensor networks.

Extensive work has been done on the connectivity maintenance issue. The research in [3] focuses on energy conservation by controlling sensor transmission power in order to maintain network connectivity. It demonstrated that the network connectivity can be maintained if each sensor has at least one neighbor in every cone of $2\pi/3$. In ASCENT [4], sensors measure local connectivity and decide whether to join the routing infrastructure based on application requirements. Xu *et al.* [5] proposed two algorithms that can conserve energy by identifying redundant nodes of connectivity.

The other issue, coverage maintenance, has also driven lots of research efforts recently. Tian *et al.* [6] presented a node-scheduling algorithm to turn off redundant sensors if their sensing areas are covered by their neighbors. Randomized as well as coordinated sleep algorithms were proposed in [7] to maintain network coverage using low duty-cycle sensors. A $K$-coverage maintenance algorithm was proposed in [8] so that each location of the sensing area is covered by at least $K$ sensors. A sensor decides whether it is redundant only by checking the coverage state of its sensing perimeter. In [9], the redundancy of the sensing coverage of wireless sensor networks is analyzed, and the relation between the number of neighbors and the coverage redundancy is studied. Abrams *et al.* studied a variant of the NP-hard SET $K$-COVER problem in [10], partitioning the sensors into $K$ covers such that as many areas are monitored as frequently as possible. Yan *et al.* [11] proposed an adaptable energy-efficient sensing coverage protocol, in which each sensor broadcasts a random time reference point, and decides its duty schedule based on neighbors' time reference points.

We propose a new coverage maintenance scheme called *Coverage-Aware Self-Scheduling (CASS)*. Different from the existing work, CASS analyzes the probabilistic sensing behaviors of sensors caused by signal fading and background noise to schedule sensors' sensing activities, therefore saving the energy consumed for sensing, computing and communicating data. Instead of providing strict guarantee of network coverage (*e.g.*, $K$-coverage), CASS takes a probabilistic approach, in which sensors autonomously adjust their probability of hibernation according to their local coverage information. In order to maintain the network connectivity, a unifying framework is proposed to seamlessly incorporate the operations of connectivity and coverage maintenance. The framework allows fine-tuning requirements for networking and sensing purposes. We simulate CASS with an existing connectivity maintenance scheme called Low Energy Adaptive Clustering Hierarchy (LEACH) [12] in the unifying framework. Using the unifying framework, it is possible to investigate CASS with other connectivity maintenance protocols and the choice of different connectivity protocols shall have little effect on the performance of CASS.

The rest of this paper is organized as follows. Assumptions are described in Section 2. Section 3 specifies CASS in details. In Section 4, we describe the framework to incorporate connectivity and coverage maintenance operations. Simulation results are presented in section 5 for performance evaluations. Section 6 concludes the paper.

## 2 Assumptions

We assume that sensors are static, and that each sensor knows its location as well as its neighbors'. Such assumptions are conveniently taken by other works [6] [8] [11] and are supported by the existing research [13] [14] [15]. The location information can be absolute or relative to neighbors.

We assume that each sensors can *separately* control the states of RF and sensing units, *i.e.*, the state of the RF unit is independent from the sensing unit state.

We assume that the sensing ability model of sensors is available before deployment through calibration process. A sensor detects an event based on its measurement, and the event is detected if the measurement is above a preset threshold. Due to the signal attenuation and noise, a sensor's measurement is modeled by a probability density function, which varies with the type of signals and the propagation channel. In CASS, the sensing ability of a sensor is modeled as the probability of a successful detection of certain events of interests. Apparently, a sensor's sensing ability is a function of the distance between the sensor and the event (a similar concept of Sensor Field Intensity is developed in [16]). Compared with the disk sensing model (*i.e.*, sensors can only detect an event happening within a certain range) assumed by the existing work, the probabilistic sensing model better reflects a sensor's sensing behavior.

We use $S_j(x,y)$ to describe sensor $j$'s sensing ability at the location $(x,y)$. A sensor's sensing range, which is denoted by $SR$, is defined as the range, beyond which the sensor's sensing ability can be neglected. We further assume that sensors' communication range is larger than or equal to $2 \cdot SR$, which is usually true in practice. For example, ultrasonic sensors have a sensing range of approximately $0.2-6m$ [17] while the transmission range of MICA motes is about 30 meters [1]. In the cases that the communication range is less than $2 \cdot SR$, our algorithm can work through multi-hop transmissions.

## 3 Coverage-Aware Self-Scheduling

### 3.1 Coverage metrics of network coverage

Based on our assumptions, the sensing coverage of a sensor network is defined as the probability of detecting an event by any of the sensors in the network, or

$$C(x,y) = 1 - \prod(1 - S_j(x,y)) \tag{1}$$

We then introduce two metrics to evaluate the sensing coverage of a sensor network. The first metric, *gross coverage*, is defined as the summation of the sensing coverage over the target area from (0, 0) to $(X, Y)$, or

$$\begin{aligned} GC &= \int_0^X \int_0^Y C(x,y)dxdy \\ &= \int_0^X \int_0^Y (1 - \prod(1 - S_j(x,y)))dxdy \end{aligned} \tag{2}$$

The second metric, *coverage extensity*, is defined as the percentage of the sub-areas where network coverage

$C(x,y)$ is lower than a certain threshold. When the threshold is set to 0, coverage extensity represents the percentage of the uncovered/blind areas.

### 3.2 Coverage contribution

The coverage contribution of a sensor is defined as the difference of the gross coverage of the sensor network with and without the sensor. For sensor $m$ with $N$ neighbors within $2 \cdot SR$, its coverage contribution can be calculated by

$$\begin{aligned} CC_m &= GC - GC' \\ &= \int_0^X \int_0^Y S_m(x,y) \prod_{j=1}^{N} (1 - S_j(x,y)) dxdy \end{aligned} \quad (3)$$

where $GC$ and $GC'$ are the gross coverage with and without sensor $m$, respectively.

Since the existence of a sensor only affects the area covered by the sensor, its coverage contribution can be calculated by only considering the area within its $SR$. For computation convenience, Eq. (3) is converted into polar coordinates:

$$\begin{aligned} CC_m &= GC - GC' \\ &= \int_0^{2\pi} \int_0^{SR} S_m(\theta,r) \prod_{j=1}^{N} (1 - S_j(\theta,r)) r d\theta dr \end{aligned} \quad (4)$$

where $SR$ is the sensing range.

According to Eq. (4), the coverage contribution can be calculated by each sensor in a distributive way given the states of its neighbors within $2 \cdot SR$.

### 3.3 The self-scheduling algorithm

CASS is designed to maximize the gross coverage per unit of consumed energy while satisfying the application requirement on coverage extensity. To achieve this goal, each sensor measures its coverage contribution in a distributed fashion. Sensors with higher coverage contribution are more preferred to be active. The details of CASS are described in the rest of the section.

Initially, each sensor is active and maintains an active neighbor list containing all the neighbors within its transmission range, which is approximated by $2 \cdot SR$. Afterward, each sensor sets a back-off timer to make its decision. When a sensor times out, the sensor computes its coverage contribution according to Eq. (4) using the active neighbor list. The coverage contribution is used to compute the probability to turn off the sensing unit,

$$P_m = \frac{(CC_{base} - CC_m)}{CC_{base}} \quad (5)$$

where $CC_m$ is the coverage contribution of sensor $m$, and $CC_{base}$ is a system parameter defined by

$$CC_{base} = \varepsilon \int_0^{2\pi} \int_0^{SR} S_m(\theta,r) r d\theta dr \quad (6)$$

where $\varepsilon$ is a tunable parameter and the double integral is the maximum possible coverage contribution when sensor $m$ has no neighbor. With a larger $\varepsilon$, $CC_{base}$ is larger and a sensor is more aggressive to turn off the sensing unit.

According to $P_m$, if the sensor decides to stop sensing, it turns off the sensing unit and broadcasts a `HIBERNATE` message to its neighbors. When a sensor receives a `HIBERNATION` message before the timer expires, it removes the sensor from the active neighbor list.

According to Eq. (5), sensors with relatively high coverage contribution have more chance to keep its sensing unit active. Please note that when $CC_m$ is larger than $CC_{base}$, $P_m$ is negative, which means sensor $m$ has a negative probability to turn off its sensing unit and should maintain the sensing unit active.

## 4 The Connectivity and Coverage Maintenance Framework

The self-organization in sensor networks involves two different issues, the connectivity maintenance and the coverage maintenance. In [2], an integrated approach for coverage and connectivity maintenance is proposed, in which a sensor should stay active if it is needed *either* for communication or coverage purposes.

In a real sensor network deployment, however, an application may have separate requirements on coverage and connectivity. For instances, an application may require low-quality monitoring with high-bandwidth data transmission, or high-quality monitoring with low-bandwidth data transmission. Furthermore, sensors may have various combinations of sensing and communication capabilities. For instance, a sensor's transmission range may be larger or smaller than its sensing range. The above observations imply that the necessary sensor densities to fulfill the coverage and connectivity requirements are usually different. Thus, the integrated solution proposed in [2] may inevitably cause redundant active RF or sensing units. In contrast to [2], we propose to separate the control of RF units (*i.e.*, connectivity maintenance) from the management of sensing units (*i.e.*, coverage maintenance). Under our framework, the connectivity maintenance protocol decides the active/inactive state of RF units and the coverage maintenance protocol determines the active/inactive state of sensing units. Jointly, there are four possible sensor states:

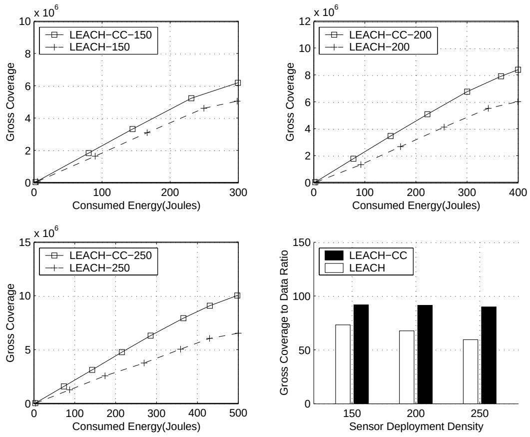


Figure 1: Gross Coverage.

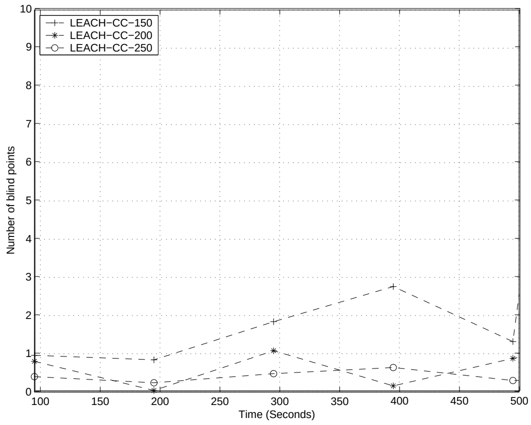


Figure 2: Number of Blind Points - LEACH-CC.

1. Relaying. Relaying nodes can relay data from other sensors to keep global connectivity. Sensing units of relaying nodes are off.
2. Sensing. Sensing nodes maintain the sensing coverage of the network. The RF units of sensing nodes are usually off but can be activated when there are sensing data to transmit. When there is no data collection node within one hop, sensing nodes send the data to relaying nodes. Since RF units are off most of the time, sensing nodes cannot relay data.
3. Active. Active nodes perform the functions of both sensing and relaying.
4. Hibernating. Both the RF unit and sensing unit of a hibernating node are off. Hibernating nodes perform no function of sensing or relaying.

The advantages of the framework are: 1) compared with the integrated approach, more energy is conserved by only keep active the necessary RF units and sensing units; 2) under the framework, any connectivity and coverage maintenance schemes can coexist and work independently, so that different requirements on connectivity and coverage can be easily satisfied.

## 5 Simulation Evaluations

To verify the validation of the framework proposed in Section 4, we integrate CASS with LEACH [12] and evaluate their performance through simulation experiments.

Note that we do not compare CASS with other existing work on coverage maintenance (*e.g.*, [6], [8] and [11]) because they assume the disk sensing model and cannot work under the probabilistic sensing model.

### 5.1 Experiment setup

For simulation simplicity, we assume a virtual sensing model for the sensors:

$$S_j(x,y) = \frac{1}{(1+\alpha\sqrt{(X_j-x)^2+(Y_j-y)^2})^{\beta}} \qquad (7)$$

where $(X_j, Y_j)$ is the coordinate of sensor $j$ and the square root represents the Euclidean distance from sensor $j$ to location $(x,y)$.

If not explicitly specified, $\alpha$ and $\beta$ are default set to 0.1 and 3, respectively. We regard the sensing ability less than 6% as negligible and set the $SR$ of the default model to 15 meters. In the simulations, $\varepsilon$ is set to 0.75 by default and $CC_{base}$ is calculated according to Eq. (6).

Each instruction consumes about $1pJ$ (pico-Joule) of energy [18]. Assuming that the calculation of coverage contribution requires about 100,000 instructions, we derive that the calculation of coverage contribution consumes about $100nJ$ (nano-Joule). We assume thermal sensors in the simulations and each thermal sampling of 10 bits costs $4nJ$ [18]. The sampling rate is set to $25Hz$. And the initial energy of each sensor is set to $2J$.

In our simulations, LEACH is extended with CASS, and is referred as *LEACH-CC*. In LEACH-CC, each sensor runs the CASS algorithm right before the cluster formation in each round. Following the unifying framework, RF units are controlled by LEACH, and sensing units are managed by CASS. As described in Section 4, the role of a sensor is jointly decided by the states of the RF and sensing units, and one of the four states is taken, *i.e.*, relaying, sensing, active and hibernating states.

In the simulations, sensors are randomly deployed over a square area from (0, 0) to (100, 100). We ran the simulations in three different network densities, *i.e.*, 150-node, 200-node and 250-node. In each network density, 5 scenarios are randomly generated, and simulations are run 10 times per scenario. The simulation results show the average performance of LEACH and LEACH-CC in these 50 runs.

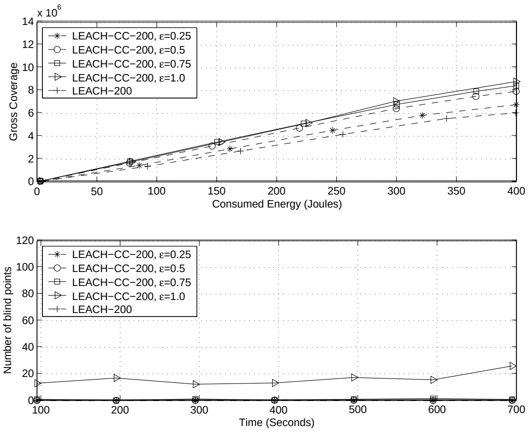


Figure 3: The Effect of $CC_{base}$.

## 5.2 Result analysis

We evaluate the performance of LEACH and LEACH-CC in terms of the sensing coverage performance and protocol overhead.

### 5.2.1 Sensing coverage

Two metrics are collected in all the scenarios for the sensing coverage statistics - the gross coverage, and the coverage extensity. In the simulations, the gross coverage computations are carried out in the following way. For each aggregated data packet received from a cluster-head, the data collection node calculates the gross coverage of the network that consists of the sensors contributing to the aggregated packet, using Eq. (2). The gross coverage is accumulated for each aggregated data packet received.

In figure 1, the gross coverage of LEACH and LEACH-CC is compared. The first three diagrams show the gross coverage achieved by LEACH and LEACH-CC in different network densities (*i.e.*, 150-node, 200-node and 250-node). We can see that LEACH-CC achieves more gross coverage than LEACH in all the cases. In LEACH-CC, sensors adjust their sensing behaviors according to their local coverage information (*i.e.*, coverage contribution). Thus, LEACH-CC improves the gross coverage achieved per unit of energy consumed. The fourth diagram of figure 1 shows the ratio of gross coverage to the number of data packets. We can see that the ratio of LEACH decreases with the increment of network density, while the ratio of LEACH-CC almost keeps stable because sensors become more aggressive to turn off sensing units due to the lower coverage contribution in a higher network density. As a result, LEACH-CC achieves about 22% more gross coverage than LEACH in 150-node networks, while about 53% improvement is observed for 250-node networks. We have carried out simulations with higher network densities, and similar observation has been made, *e.g.*, about 100% improvement in 450-node networks.

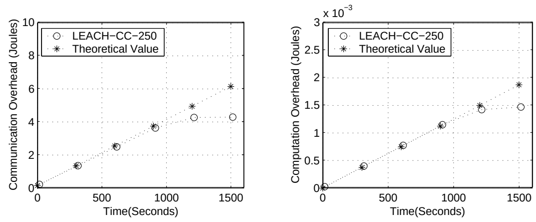


Figure 4: Communication and Computation Overhead - LEACH-CC.

The coverage extensity is measured by setting the network sensing accuracy threshold to 0, and computing the number of blind points out of $10,000$ sampling points. Figure 2 shows that LEACH-CC presents less than $0.03\%$ blind points in different network densities. Intuitively, by powering off more sensing units, the spatial correlation of data packets decreases (*i.e.*, sensors have less probability to detect the same event), thus better gross coverage performance can be expected. But coverage extensity becomes worse since more uncovered sub-area may be exposed by turning off more sensing units. So there is a tradeoff between the two coverage performance metrics.

The tradeoff is demonstrated by figure 3. From the upper diagram, the gross coverage becomes better with the increase of $\varepsilon$. This is because sensors become more aggressive to turn off sensing units with a larger $CC_{base}$. In the lower diagram, more blind points are presented with the increase of $\varepsilon$. Thus, it is very important to choose an appropriate $\varepsilon$ value for CASS to maximize gross coverage achieved while not degrading coverage extensity too much. Empirically, we can see from figure 3 that $0.75$ is an appropriate value for $\varepsilon$. Although not studied in the current stage of this research, it is possible to have CASS dynamically adjust $\varepsilon$ value according to local factors, such as the event happening frequency.

### 5.2.2 Communication and computation overhead

The first diagram of figure 4 depicts the estimated and measured communication overhead of LEACH-CC in 250-node networks. The measured communication overhead is about $4.43J$, or 0.8% of the total energy consumption. The second diagram of figure 4 demonstrates the estimated and measured computation overhead of LEACH-CC in 250-node networks. The measured computation overhead is about $1.47mJ$. The estimated curve and measured curve grow apart after about 1000 seconds because a part of sensors dies due to power outage. The overall overhead of LEACH-CC including the communication and computation overhead is about $4.43147J$, or only 0.8% of the total energy consumed.

## 6 Conclusions

We have described a new coverage maintenance scheme called CASS, which allows sensors to decide the state of their sensing units in a distributed fashion according to their local coverage information. A unifying framework is proposed to incorporate different connectivity and coverage maintenance schemes. Under such a framework, we evaluate the performance of CASS by integrating CASS with an existing connectivity maintenance protocol - LEACH. The simulation results verify the validation of the proposed framework and show that CASS achieves considerable improvements to the energy efficiency for coverage maintenance with small overhead.

## References

[1] "http://www.xbow.com/Support/Support_pdf_files/MPR-MIB_Series_Users_Manual.pdf", XBOW Inc., MPR/MIB User's Manual, Last visited on 06/30/2005.

[2] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks", *Proceedings SenSys*, 2003.

[3] R. Wattenhofer, L. Li, P. Bahl, and Y. Wang, "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks", *Proceedings InfoCom*, 2001.

[4] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring Sensor Network Topologies", *Proceedings InfoCom*, 2002.

[5] Y. Xu, S. Bien, Y. Mori, J. Heidemann, and D. Estrin, "Topology Control Protocols to Conserve Energy in Wireless Ad Hoc Networks", *Technical Report 6, University of California, Los Angeles*, 2003.

[6] D. Tian and N.D. Georganas, "A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks", *Proceedings WSNA*, 2002.

[7] C. Hsin and M. Liu, "Network Coverage Using Low Duty-Cycled Sensors: Random & Coordinated Sleep Algorithms", *Proceedings IPSN*, 2004.

[8] C. Huang and Y. Tseng, "The Coverage Problem in a Wireless Sensor Network", *Proceedings WSNA*, 2003.

[9] Y. Gao, K. Wu, and F. Li, "Analysis on the Redundancy of Wireless Sensor Networks", *Proceedings WSNA*, 2003.

[10] Z. Abrams, A. Goel, and S. Plotkin, "Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks", *Proceedings IPSN*, 2004.

[11] T. Yan, T. He, and J.A. Stankovic, "Differentiated Surveillance for Sensor Networks", *Proceedings SenSys*, 2003.

[12] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", *Proceedings Hawaii International Conference on System Sciences (HICSS)*, 2000.

[13] J. Albowicz, A. Chen, and L. Zhang, "Recursive Position Estimation in Sensor Networks", *Proceedings IEEE International Conference on Network Protocols*, 2001.

[14] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less Low Cost Outdoor Localization for Very Small Devices", *IEEE Personal Communications*, 2000.

[15] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System", *Proceedings MobiCom*, 2000.

[16] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure In Wireless Ad-Hoc Sensor Networks", *Proceedings MobiCom*, 2001.

[17] "http://www.robosoft.fr/SHEET/02Local/1001LAUN/LAUN.html", Ultrasonic Sensors, Last visited on 06/30/2005.

[18] J. Pister, "Energy and Performance Considerations for Smart Dust", *International Journal of Parallal and Distributed System and Networks*, 2001.