International Journal of Current Research and Modern Education (IJCRME)
Impact Factor: 6.725, ISSN (Online): 2455 - 5428
(www.rdmodernresearch.com) Volume 2, Issue 1, 2017

PERFORMANCE EVALUATION OF LOCAL BARRIER COVERAGE IN WIRELESS SENSOR NETWORKS

K. K. Kannan
Assistant Professor, Indus College of Technology, Coimbatore, Tamilnadu

Cite This Article: K. K. Kannan, "Performance Evaluation of Local Barrier Coverage in Wireless Sensor Networks", International Journal of Current Research and Modern Education, Volume 2, Issue 1, Page Number 230-236, 2017.

Abstract:

The main aim of project is to create appropriate model of coverage for movement detection applications such as intrusion detection. However it has been proved that given sensor deployment sensors cannot locally determine whether deployment provides global coverage. Making it impossible to develop localized algorithms thus limiting it Dees use in practice. In this project I am introducing the concept of local barrier coverage to address this limitation. It is motivated by the observation that movements are likely to follow a shorter path in crossing a belt region, local barrier coverage guarantees detection of all movements whose trajectory is confined to a slice of belt region of deployment. To prove that it is possible that individual sensors to locally determine the existence of local barrier coverage, even when the region of deployment is arbitrarily curved. To demonstrate that local barrier coverage can be use to design localized algorithms. For maximization network life time we develop a sleep-wakeup algorithm called localized barrier coverage protocol. LBCP guarantees local barrier coverage and show that LBCP provides close to optimal enhancement in the network life time while providing global barrier coverage most of the time. They out perform an existing algorithm called randomized independent sleeping (RIS) by up to six times.

Introduction:

A mobile ad hoc network (MANET) consists of a group of mobile nodes (MNs) that communicates with each other without the presence of infrastructure. MANETs are used in disaster recovery, rescue operations, military communication and many other applications. In order to provide communication throughout the network, the mobile nodes must cooperate to handle network functions, such as packet routing. The wireless mobile hosts communicate in a multihop fashion. In multi-hop wireless ad-hoc networks, designing energy-efficient routing protocols is critical since nodes have very limited energy, computing power and communication capabilities. The key design challenge is to derive the required global properties based on these localized algorithms. Several important applications of wireless sensors involve movement detection, such as when deploying sensors along international borders to detect illegal intrusion, around a chemical factory to detect the spread of lethal chemicals, on both sides of a gas pipeline to detect potential sabotage, etc. Barrier coverage, which guarantees that every movement crossing a barrier of sensors will be detected, is known to be an appropriate model of coverage for such applications Barrier coverage has several advantages over full coverage, a model requiring every point in the deployment region to be covered. First, barrier coverage requires much fewer sensors than full coverage Second, the sleep-wake up problem, which determines a sleeping schedule for sensors to maximize the network lifetime A major limitation of the barrier coverage model, however, is that unlike full coverage, individual sensors cannot locally determine whether a network does not provide barrier coverage, making it impossible to develop localized algorithms. Consequently, almost all algorithms developed so far for barrier coverage, including the optimal sleep wakeup algorithm, are centralized. A localized algorithm is also more adaptive to changes in the network, which is expected to be quite frequent in wireless sensor network due to the unattended outdoor deployments. Therefore, in order to realize the benefits of the barrier coverage model in movement detection applications, there is a strong need to develop a new model that enables the development of localized algorithms, while essentially retaining the benefits of barrier coverage. Due to a wide diversity of WSN application requirements, however, a general-purpose WSN design cannot fulfill the needs of all applications. Many network parameters such as sensing range, transmission range, and node density have to be carefully considered at the network design stage, according to specific applications. To achieve this, it is critical to capture the impacts of network parameters on network performance with respect to application specifications. Intrusion detection (i.e., object tracking) in a WSN can be regarded as a monitoring system for detecting the intruder that is invading the network domain. The intrusion detection application concerns how fast the intruder can be detected by the WSN. If sensors are deployed with a high density so that the union of all sensing ranges covers the entire network area, the intruder can be immediately detected once it approaches the network area. However, such a high-density deployment policy increases the network investment and may be even unaffordable for a large area. In fact, it is not necessary to deploy so many sensors to cover the entire WSN area in many applications. In this project, i prove a theorem that allows a convenient discretization so that instead of checking each of the infinite bounding boxes to establish that a sensor network provides L-local barrier coverage, one only needs to check if the neighborhood of each sensor is

barrier covered. Although local barrier coverage does not deterministically guarantee global barrier coverage (when L is less than the length of the deployment region), we show (by simulation) that for thin belt regions, local barrier coverage almost always provides global barrier coverage. This means that for thin belts, checking locally for the existence of local barrier coverage is sufficient to ensure global barrier coverage in practice.

System Description:

Recent advancements in wireless communication and the miniaturization of computers have led to a new concept called the Mobile Ad-hoc Network (MANET), where two or more mobile nodes can form a temporary network without need of any existing network infrastructure or centralized administration. In MANETs mobile nodes act as routers themselves, keeping route information to reach other mobile nodes and helping forward data packets sent from one mobile node to another. Mobility heavily affects data availability; high mobility sometimes increases data availability, e.g. a mobile node relays data between two separated networks and it sometimes decreases, e.g. a mobile node that holds hot data disconnects from the network. However, most of the conventional works assumed a particular mobility model and only examined the influence of the mobility model on the performance of the proposed approach. In other words, they did not give any general insights on the relationship between mobility and data availability. Distributed storage systems have gained popularity for storing and processing even increasing amount of data. These systems are designed to share the storage resources of network nodes and can be deployed in controlled networks such as an enterprise network. In these networks a priori trust relation between nodes is available. In mobile ad-hoc networks, nodes are controlled by users who do not necessarily belong to the same authority; a priori trust relation between nodes is not available. In addition, mobile nodes operate on low-power batteries and use limited power processing and storage capacity. These constraints lead nodes to behave selfishly by discarding some data they have agreed to store for other nodes in order to gain resource for their own usage. To address this issue, several protocols for the verification of data possession have been proposed for peer-to-peer networks and ad hoc networks. These protocols aim at detecting misbehaved data holder nodes that claim to hold data, which they in fact destroyed.

These protocols are generally based on exchanging challenge and response messages between the data owner node and the data holder node. Four categories of protocols can be distinguished:

- ✓ The first one periodically challenges the data holder to send back the original data to the data owner, the holders response in this case is then compared with the original data.
- ✓ In the second one, the data owner periodically challenges its data holders by requesting a block out of the stored data. The response is checked by comparing it with the valid block stored at the verifier's disk space.
- ✓ In the third one, the data holder has to send the Message Authentication Code (MAC) of data as the response to the challenge message. The verifier periodically sends a fresh nonce as the key for the MAC.
- ✓ In the fourth one, the data owner periodically asks the data holder to compute a hash value only over a randomly selected chunk of the data at one time.

The fourth category is the best one among those mentioned above, since it requires less bandwidth consumption and less CPU processing. The remark that all the protocol consist in periodically challenging the data owner, which results in more communication and computational cost to an ad hoc network constrained by its limited resources. Moreover, triggering the challenging-response periodically results in more competition to access the shared wireless channel, and hence the challenge and response messages are likely to be lost. In this case, the probability to falsely identify a well-behaved node as misbehaved increases.

In this project propose new monitoring mechanisms that consider MANETs constraints. The original contributions are the following

- ✓ Propose a probabilistic monitoring mechanism that copes with the limitations of the existing periodic monitors.
- ✓ The probabilistic monitor is further enhanced to be adaptive in the sense that it uses node's reputation and node's local network state to adjust its behavior.
- ✓ Simulation results show that the proposed probabilistic and adaptive monitors offer a good tradeoff between exposure time and the rate of false positives. It comes second after the periodic one in terms of exposure time, and it is the closest competitor to the probabilistic one in terms of the rate of false positives.

Localized Sleep-Wake Up Protocols:

Local barrier coverage concept to design a localized sleep-wake up algorithm, called LBCP, for barrier coverage to maximize the network lifetime. The protocols usually have close-to-optimal performances and provide global barrier coverage most of the time for thin belt regions.

Assumptions:

First state some assumptions. Assume that each node has a unique ID as is common in newer platforms. We also assume that the network has been localized so that each node knows its own location. In the

event of localization inaccuracies, the identified 2d-zone of a node u may not contain the real 2d-zone. However, the error of the location, denoted by, only slightly affects the performance of LBCP. Further, we assume that two nodes u and v are able to communicate with each other if u is in v's (identified) 2d-zone or v is in u's (identified) 2d-zone. We also assume that every node is able to estimate its remaining lifetime (if it stays active) by observing its battery drainage. Battery drainage rate can be observed in recent mote platforms. Finally, we assume that the MAC protocol does not introduce too much latency; all LBCP packets are sent or received almost immediately.

Protocol LBCP:

- ✓ Initially every node is active and calls procedure INITIAL once and only once.
- ✓ After initialization every active node periodically invokes procedure ACTIVE to decide whether to stay active or go to sleep. In the latter case the node becomes sleeping.
- ✓ Every sleeping node periodically wakes up enters the waking – up state and invokes procedure WAKE-UP.

Procedure Initial:

- ✓ Node u calculates the value of d according to the desired value of and initializes sets A[u] = Φ and Nu=Φ
- ✓ Node u invokes procedure IDENTIFY (u) to identify its 2d-zone.If IDENTIFY reports that some node's 2d-zone is not k-barrier covered; u informs the base station of this fact. Otherwise u broadcasts over 2d-zone (u) an inform packet containing u's ID position and lifetime.
- ✓ When node u receives another node u's informI packet, it records u's ID position and lifetime and sets NV=NV Ů {u}.Furthermore if u Є 2dzone (v), then v replies with an inform packet containing v's id position and lifetime.
- ✓ When node u receives another node v's informb packet it records the contained information and sets Nu = Nu = Ů {v}

Procedure Identity:

- ✓ Node u computes the values of r1, r2, r3 using the values of W,d,r as described . Let r' = max {r1, r2}.
- ✓ Node u then two nodes p and q such that r' ≤d9p, u0≤r3, r′≤ d (q, v) ≤r3 and puq≥π/2.If it cannot find two such nodes then it stops and reports that at least one node's 2d-zone is not k-barrier covered by the network.

Procedure Active:

- ✓ An active node u checks if 2d-zone (u) will be k-barrier covered without u itself and the nodes in A (u); if so u sends a query a packet to the nodes in Nu.
- ✓ Whenever an active node v receives a node u,s query A packet if 2d-zone (v) will be k-barrier covered without A(v) Ů {u} then v adds u to A(v) and replies with not required a message. Otherwise v replies with with a required a message.
- ✓ After issuing a query a if v receives a not required a packet from every active node that is in Nu then it decides to go to sleep .In that case u sends a Decision Sleep packet to the nodes in Nu and goes to sleep until t time later or until the first active node in Nu is expected to die whichever occurs earlier. Otherwise vu stays active and sends to the nodes in Nu a decision continue packet containing u's ID position and lifetime.
- ✓ Whenever an active node v receives a node u's decision sleep packet v removes u from its set of active nodes and removes u from A(u); but if v receives u's Decision Continue packet , v only removes u from A(v) and renews u's information.

Procedure Wakeup:

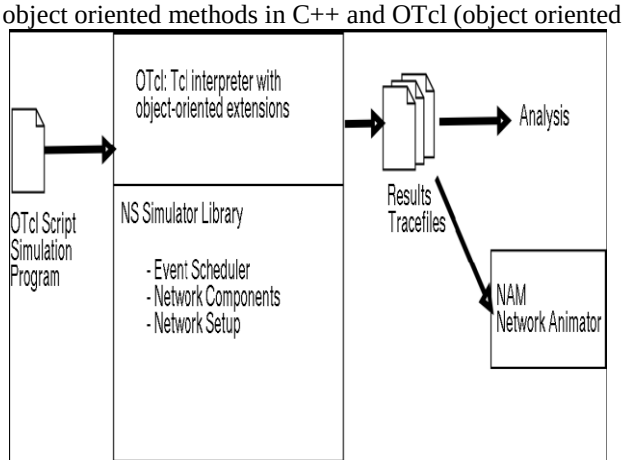
- ✓ A waking –up node u resets its record of active nodes to null and sends a Query packet to the nodes in Nu.
- ✓ When an active node v receives a query w packet from a node u, if u is in 2d-zone (v) and the latter is currently not k-barrier covered, then v replies with a required W packet containing its ID, position and lifetime. Otherwise v replies with a not required w packet containing its ID position and lifetime.
- ✓ If u receives any required w packet or if u does not receive any reply – Required W or not Required W to its query w packet (meaning there is no active node in Nu), then u becomes active and sets A (u) =Φ. Otherwise u goes back to sleep .Upon receiving a packet of either type u records the ID, position and lifetime contained in the packet.
- ✓ If u decides to go back to sleep, u sleeps until t times units later or until the first active node in Nu is expected to die whichever occurs earlier.
- ✓ If u decides to become active, u sends to the nodes in Nu a Decision Active packet containing u's ID, position and lifetime.
- ✓ When an active node v receives u's decision active packet v adds u to its list of active nodes and records ID, position and lifetime.

Software Description:

Network Simulator (NS2) is a discrete event driven simulator developed at UC Berkeley. It is part of the VINT project. The goal of NS2 is to support networking research and education. It is suitable for designing new protocols, comparing different protocols and traffic evaluations. NS2 is developed as a collaborative environment. It is distributed freely and open source. A large amount of institutes and people in development and research use, maintain and develop NS2. This increases the confidence in it. Versions are available for FreeBSD, Linux, Solaris, Windows and Mac OS X.

Structure of NS2:

NS2 is built using object oriented methods in C++ and OTcl (object oriented variant of Tcl.



Simplified User's View of Ns. NS2 interprets the simulation scripts written in OTcl. A user has to set the different components (e.g. event scheduler objects, network components libraries and setup module libraries) up in the simulation environment. The user writes his simulation as a OTcl script, plumbs the network components together to the complete simulation. If he needs new network components, he is free to implement them and to set them up in his simulation as well. The event scheduler as the other major component besides network components triggers the events of the simulation (e.g. sends packets, starts and stops tracing). Some parts of NS2 are written in C++ for efficiency reasons. The data path (written in C++) is separated from the control path (written in OTcl). Data path object are compiled and then made available to the OTcl interpreter through an OTcl linkage (tclcl) which maps methods and member variables of the C++ object to methods and variables of the linked OTcl object. The C++ objects are controlled by OTcl objects. It is possible to add methods and member variables to a C++ linked OTcl object.

NS was built in C++ and provides a simulation interface through OTcl, an object-oriented dialect of Tcl. The user describes a network topology by writing OTcl scripts, and then the main NS program simulates that topology with specified parameters.

Ns are now developed in collaboration between a number of different researchers and institutions, including SAMAN (supported by DARPA), CONSER (Collaborative Simulation for Education and Research). It is currently maintained by volunteers. Long-running contributions have also come from Sun Microsystems and the cited by the ns homepage for wireless code additions.

Network simulation software enables us to predict behavior of a large-scale and complex network system such as Internet at low cost under different configurations of interest and over long period. Many network simulators, such as NS2, Openet, Qualnet, etc., are widely available. We'll use NS2 for this project. NS2 is a discrete event simulator written in C++, with an OTcl interpreter shell as the user interface that allows the input model files (Tcl scripts) to be executed. Most network elements in NS2 simulator are developed as classes, in object-oriented fashion. The simulator supports a class hierarchy in C++, and a very similar class hierarchy in OTcl. The root of this class hierarchy is the Tcl Object in OTcl. Users create new simulator objects through the OTcl interpreter, and then these objects are mirrored by corresponding objects in the class hierarchy in C++. NS2 provides substantial support for simulation of TCP, routing algorithms, queuing algorithms, and multicast protocols over wired and wireless (local and satellite) networks, etc. It is freely distributed, and all source code is available.

Developing new networking protocols and creating simulation scripts are complex tasks, which require understanding of the NS2 class hierarchy, C++, and Tcl programming. However, in this project, you only need to design and run simulations in Tcl scripts using the simulator objects without changing NS2 core components such as class hierarchy, event schedulers, and other network building blocks. Put all your results in the files with given names. For this report, do not need to rewrite the assumptions given in this document, but should provide any information not already given to you.

An ad-hoc network is a local area network or other small network, especially one with wireless or temporary plug-in connections having some of the network devices are part of the network only for the duration of a communications session in the case of mobile or portable devices, while in some close proximity to the rest of the network. Ad hoc literally means for this purpose only and thus usually temporary. Ad hoc has been applied to future office or home networks in which new devices can be quickly added. This Blue tooth technology in which devices communicate with the computer and perhaps other devices using wireless transmission is an example for Ad hoc network.

An application of ad-hoc network technology is that one vendor offers people to come to a conference room and using infrared transmission or radio frequency (RF) wireless signals, join their notebook computers with other conferees to a local network with shared data and printing resources. Each user has a unique network address that is immediately recognized as part of the network. The technology would also include remote users and hybrid wireless/wire connections. Three types of ad hoc networks are currently used. They are

- ✓ Mobile Ad-hoc networks
- ✓ Vehicular Ad-hoc networks
- ✓ Intelligent Vehicular Ad-Hoc Network.

Vehicular Ad-Hoc Networks (VANet) are a form of MANets used for communication among vehicles and between vehicles and road side equipment. Intelligent Vehicular Ad-Hoc Network (In VANET) is a kind of Intelligence in Vehicles which provide multiple autonomic intelligent solutions to make automotive vehicles to behave in intelligent manner during vehicle-to-vehicle collisions, accidents, drunken driving etc. In VANET uses Wi-Fi IEEE 802.11b/IEEE802.11g and IEEE 802.16 for providing easy, accurate, effective communication between multiple vehicles on dynamic mobility. Effective measures to track the automotive vehicles, media download /upload, conference between vehicles are also preferred. In VANET can also be applied for artillery vehicles during warfare / Battlefield / Peace operations.

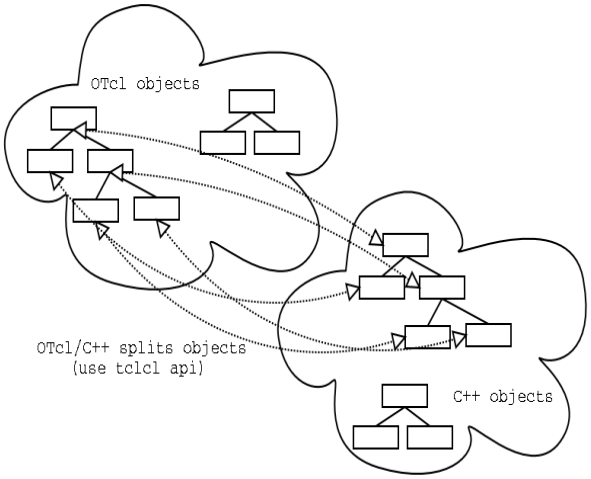
A mobile ad-hoc network is a kind of wireless ad-hoc network, and is a self-configuring network of mobile routers connected by wireless links the union of which forms an arbitrary topology. The routers are free to move randomly and organize themselves arbitrarily, thus the network's wireless topology may change rapidly and unpredictably. Such a network may operate in a standalone fashion, or can be connected to the larger Internet.

Mobile ad-hoc networks became a popular subject for research as laptops and 802.11/Wi-Fi wireless networking became widespread in the mid- to late 1990s. Many of the academic researches evaluate protocols and abilities assuming varying degrees of mobility within a bounded space, usually with all nodes within a few hops of each other, and usually with nodes sending data at a constant rate. Different protocols are then evaluated based on the packet drop rate, the overhead introduced by the routing protocol, and other measures.

Functionalities of NS2.33:

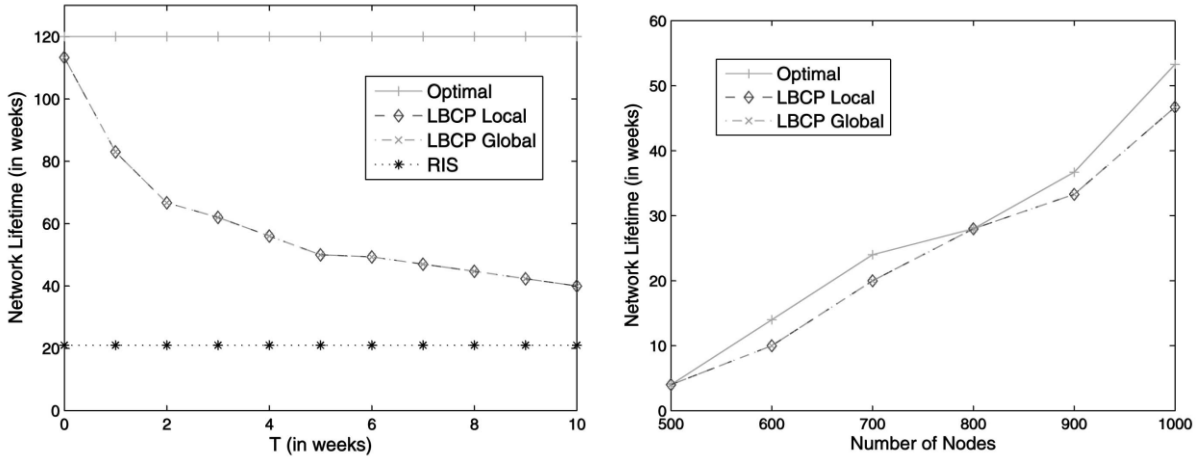
Functionalities for wired, wireless networks, tracing, and visualization are available in NS2.

- ✓ Support for the wired world include
  - Routing DV, LS, and PIM-SM.
  - Transport protocols: TCP and UDP for unicast and SRM for multicast.
  - Traffic sources: web, ftp, telnet, cbr (constant bit rate), stochastic, real audio.
  - Different types of Queues: drop-tail, RED, FQ, SFQ, DRR.
  - Quality of Service: Integrated Services and Differentiated Services.
  - Emulation.
- ✓ Support for the wireless world include
  - Ad hoc routing with different protocols, e.g. AODV, DSR, DSDV, TORA
  - Wired-cum-wireless networks
  - Mobile IP
  - Directed diffusion
  - Satellite
  - Senso-MAC
  - Multiple propagation models (Free space, two-ray ground, shadowing)
  - Energy models
- ✓ Tracing
- ✓ Visualization
  - Network Animator (NAM)
  - Trace Graph
- ✓ Utilities
  - Mobile Movement Generator
  - OTcl and C++: the duality



The Performance for General Belts:

Investigate the performance of LBCP combined with the heuristic method developed for determining the barrier coverage in a 2d-zone for general belts. Concatenate two identical semicircular belts to form an S-shaped belt. The middle line in each semicircle is 550 m long. So, the whole belt's middle line is 1; 100 m long. All other parameters (r, d, W, node lifetime) are the same as described in the beginning. Only compare the performance of LBCP with the optimal algorithm. We vary the number of nodes from 500 to 1,000. The simulation results are shown in



Two key observations:

- ✓ LBCP provides close to the optimal network lifetime and
- ✓ LBCP always provides global barrier coverage although it only strives to provide local barrier coverage, indicating that our heuristic works well

Conclusion and Future Work:

This project will propose a new notion of coverage called local barrier coverage that is more appropriate for applications than the existing global barrier coverage. I provided a localized algorithm for sensors to determine whether the sensor network provides local barrier coverage. In simulations, i observed that for thin belt regions, the network provided global barrier coverage whenever it provided local barrier coverage. The concept of local barrier coverage to develop the first localized sleep-wake up protocol for movement detection applications that provided close to optimal enhancement in the network lifetime. I proved that our protocol guarantees local barrier coverage. I showed that in addition to ensuring global coverage most of the time, local barrier coverage also ensured connectivity under some mild assumptions. By enabling the development of localized algorithms for barrier coverage, work might have opened up many interesting research problems. For instance, localized algorithms for other tasks such as barrier coverage network repair may now be explored. The concept of L-local barrier coverage can also be used to measure the quality of barrier coverage provided by a sensor network.

Future Enhancements:

In our proposed method all the sensor nodes in the 2d zone are able to communicate with each other nodes directly or indirectly using the intermediate nodes. The routin is based on the shortest path. In our future work we find the intermediate node with the maximum energy and used that node for the communication in order to increase the network lifetime of the sensor nodes.

- ✓ Node Activation
- ✓ Intruder Detection
- ✓ Connectivity Improvement

Node Activation:

In this module we activate the failure node from the base station and make the active node which is turned to ACTIVE state during the failure of the failure node. The Activation information is passed to all the sensor nodes in the belt region.

Intruder Detection:

In this module we find the Intruder enter into the belt region. The nearby ACTIVE sensor detects the intruder and its id and gives the information to the base station

Connectivity Improvement:

In order to improve the connectivity in the network during the failure of the sensors we introduce transmitters in the region to transfer information to the base station.

References:

1. Ai Chen, Santosh Kumar, and Ten H. Lai (2010) "Local barrier coverage in wireless sensor networks".
2. P. Balister, B. Bolloba´s, A. Sarkar, and S. Kumar, (2007)"Reliable Density Estimates for Coverage and Connectivity in Thin Strips of Finite Length," Proc. ACM MobiCom.
3. M. Cardei, M. Thai, and W. Wu, (2005) "Energy-Efficient Target Coverage in Wireless Sensor Networks," Proc. IEEE INFOCOM.
4. B. Carbunar, A. Grama, J. Vitek, and O. Carbunar, "Redundancy and Coverage Detection in Sensor Networks," ACM Trans. Sensor Networks (TOSN), vol. 2
5. Chen, S. Kumar, and T.H. Lai, (2008) "Local Barrier Coverage with Wireless Sensor Networks," Technical Report OSU-CISRC-9/08- TR49, Ohio State Univ. (OSU).
6. Chen, T.H. Lai, and D. Xuan,(2008) "Measuring and Guaranteeing Quality of Barrier-Coverage in Wireless Sensor Networks," Proc. ACM MobiHoc.
7. T. He et al., (2004) "Energy-Efficient Surveillance System Using Wireless Sensor Networks," Proc. ACM Mobisys.
8. Huang and Y. Tseng, (2003) "The Coverage Problem in a Wireless Sensor Network," Proc. ACM Int'l Workshop Wireless Sensor Networks and Applications (WSNA).
9. S. Kumar, (2006) "Foundations of Coverage in Wireless Sensor Networks," technical report, PhD thesis, Ohio State Univ. (OSU).
10. S. Kumar, T.H. Lai, and A. Arora, (2005) "Barrier Coverage with Wireless Sensors," Proc. ACM Mobi Com.