

Multi-Channel and Multi-Timeslot Convergecast for Barrier Coverage in Wireless Sensor Networks

Jehn-Ruey Jiang, Chieh-Chun Huang, and Shih-Yi Chan

Department of Computer Science and Information Engineering

National Central University, Jhongli City, 32001, Taiwan

Keywords: wireless sensor network, barrier coverage, optimization, multi-channel and time-slot scheduling, IEEE 802.15.4.

Abstract. We propose algorithms performing multi-channel and multi-timeslot convergecast for barrier coverage in wireless sensor networks (WSNs). We also simulate the proposed algorithms and a related algorithm on the basis of the IEEE 802.15.4 protocol with OPNET and MATLAB for the sake of performance comparisons. By the simulation results, the proposed algorithms significantly outperform the related algorithm in terms of the notification success rate and the notification delay.

Introduction

A *wireless sensor network* (WSN) consists of many sensor nodes and few sink nodes [8]. The sensor nodes can sense physical phenomena, such as temperature, humidity, light luminosity, sound intensity, etc. They can also *convergecast* [1, 7] the sensed data to one of the sink nodes through a chain of intermediate nodes with wireless communications. WSNs have been widely used in many areas, such as battlefield surveillance, animal tracking, wildfire detection, environmental monitoring, and healthcare, attracting much research attention recently.

Some of the WSN research focused on the *barrier coverage problem* concerning how to deploy or select WSN sensor nodes in a monitored belt-shaped region to form a virtual barrier so that any intruder crossing the region can be detected by sensor nodes [4-6, 11]. This paper is most related to barrier coverage optimization research [4-5]. In [4], Kumar et. al. studied the *k-barrier coverage optimization problem* related to how to deploy or select WSN sensor nodes to form a virtual barrier of the highest degree k so that at least k sensor nodes can detect an intruder crossing the barrier. Lai and Jiang [5] investigated the *sink-connected barrier coverage optimization (SCBCO) problem*, trying to find the minimum number of *detecting nodes* for forming a virtual barrier of the highest degree k and to find the minimum number of *forwarding nodes* for forwarding packets to sink nodes so as to make the detecting nodes *sink-connected*. To solve this problem, they proposed an algorithm, called the *optimal node selection algorithm (ONSA)*, which is based on concepts of the *node disjoint/edge transformation* and the *maximum flow minimum cost (MFMC)* algorithm.

The ONSA successfully solves the SCBCO problem to construct a sink-connected barrier coverage with the maximum degree by using the minimum number of sensor nodes for detecting intrusion events. However, it does not resolve the problem concerning how to convergecast intrusion notifications to sink nodes reliably and timely. This is a challenging problem, as many sensor nodes will detect an intruder and send notifications to sink nodes at almost the same time, causing interference, which in turn prevents sink nodes or sensor nodes from receiving the notifications successfully [10, 12]. This motivates us to solve such a problem.

In this paper, we call the aforementioned problem the *interference-avoiding barrier coverage convergecast (IVBCC)* problem. We intend to solve the problem by proposing algorithms for achieving the following two goals for a given set of detecting nodes of k -barrier coverage of a WSN.

Goal-1: Forming convergecast trees to ensure every detecting node has a routing path for sending intrusion notifications to one of the sink nodes

Goal-2: Assigning channels and timeslots to convergecast tree nodes to avoid as much as possible interference among intrusion notifications

We propose an algorithm, called the *Voronoi Diagram Convergecast Tree Algorithm (VDCTA)* to achieve Goal-1 by combining the Voronoi diagram algorithm and the modified Dijkstra shortest path algorithm [9] to construct shortest path trees as convergecast trees. We further propose another algorithm, called the *Channel And Timeslot Scheduling Algorithm (CATSA)*, to achieve Goal-2 by using the greedy strategy to perform multi-channel and multi-timeslot scheduling for convergecast tree nodes to send notifications. The studies in [1, 3, 7] addressed either the multi-channel or the multi-timeslot scheduling for WSNs. However, the studies are not related to the barrier coverage problem and none of them investigated the scheduling of both multiple channels and multiple timeslots. In addition to the algorithm design, we consider algorithm implementation details and simulate the proposed algorithms and ONSA [5] on the basis of the IEEE 802.15.4 protocol [2] with OPNET and MATLAB for the sake of performance comparisons. By the simulation results, the proposed algorithms significantly outperform ONSA in terms of the notification success rate and the notification delay.

The rest of the paper is organized as follows. In Section 2, we present the network model and the interference model. In Section 3, we describe the proposed algorithms. Simulation results are reported in Section 4, and Section 5 concludes the paper.

Modelling

Network Model

Consider a WSN consisting of many sensor nodes and few sink nodes. Sensor nodes are randomly deployed throughout a specific monitored belt-shaped region B (e.g., a section of a boundary river between two countries), while sink nodes may be deployed at arbitrary positions or specific positions inside or outside the region B . It is assumed that each sensor node can find out its position and report it to the backend system after deployment.

The ONSA algorithm [5] can select a minimum set of detecting nodes to form the optimal k -barrier coverage for the region. As shown in Fig. 1, the detecting nodes can form a virtual barrier so that any intruder penetrating the region by passing through the region outer side and then passing through the region inner side can be detected by at least k ($k=3$ in Fig. 1) sensor nodes.

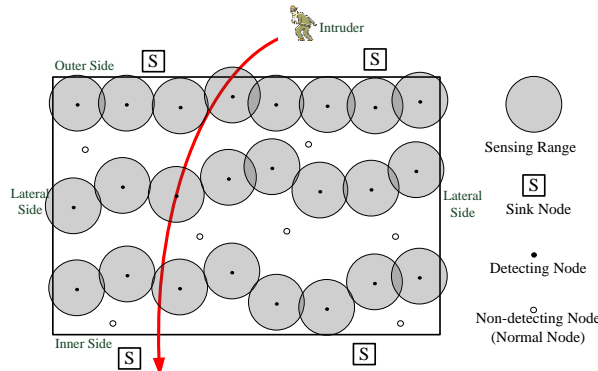


Fig. 1: An example of 3-barrier coverage

We below follow the terminologies used in ONSA [5]. We call the nodes forming the virtual barrier the *detecting nodes*. The sensor nodes that are not detecting nodes are called *non-detecting nodes* or *normal nodes*. The ONSA algorithm can further construct for every detecting node a routing path going through other detecting nodes or normal nodes toward one of the sink nodes. The normal nodes included in routing paths are called *forwarding nodes*. Below, we use V_s , V_k , V_d , V_n , and V_f to denote the sets of sensor nodes, sink nodes, detecting nodes, normal (non-detecting) nodes, and forwarding nodes, respectively.

We assumed that each sensor node has a fixed sensing range R_s , and a fixed communication range R_c . Note that we assume $R_c=2R_s$ in the following context. This is a practical assumption, since most sensor nodes have the communication range much larger than twice of the sensing range, and we can adjust the communication range R_c of every sensor node to be $2R_s$.

The directed communication graph $G_c=(V_a, E_c)$ is used to represent the communication relationship between nodes, where V_a is the set of all nodes (i.e., $V_a=V_s\cup V_k$) and E_c is the set of directed edges representing communication links. There is an edge (v_i, v_j) in E_c if $|v_i-v_j|\leq R_c$, where $|v_i-v_j|$ stands for the distance between v_i and v_j . Based on G_c , we can run routing algorithms to derive convergecast trees such that every detecting node has a path going through tree nodes to reach one of the sink nodes.

1.1 Interference Model

The performance of wireless communications is significantly affected by interference caused by simultaneous transmission. There are two widely used models to characterize interference between wireless network nodes: the *physical model*, a.k.a. *signal-to-interference-noise-ratio (SINR) model*, and the *protocol model*, a.k.a. *unified disk graph model* [10]. Under the physical model, a transmission is assumed to be successful if signal SINR at the intended receiver exceeds a threshold. Under the protocol model, a transmission is assumed to be successful if a node falls inside the transmission range of its intended transmitter and falls outside the interference ranges of other non-intended transmitters. Generally, the interference range of a node is 2 to 4 times of the communication range. The physical model better reflects the practical situation than the protocol model; however, the former has much higher computation complexity. Therefore, some studies adopt the protocol model to characterize interference [12].

In this paper, we adopt the protocol interference model and assume the interference range R_i of every node is twice of the communication range R_c (i.e., $R_i=2R_c$). We also apply the *receiver-based conflict graph* [10] for channel assignment (scheduling) on convergecast trees according to the protocol interference model. In a receiver-based conflict graph $G_f=(V_f, E_f)$ for a convergecast tree T , V_f is the set of all receivers in T , and an edge (v_i, v_j) exists in E_f (i.e., v_i conflicts with v_j) if a certain transmission destined at node v_i will interfere a transmission destined at v_j . Note that since we assume $R_i=2R_c$, two receivers conflict with each other and the conflict graph is symmetric; we therefore use undirected graph to represent conflict graph in this paper.

Propose Algorithms

The Voronoi Diagram Convergecast Tree Algorithm (VDCTA)

The Voronoi Diagram Convergecast Tree Algorithm (VDCTA) combines the Voronoi diagram algorithm and the modified Dijkstra shortest path algorithm [9] to form convergecast trees such that every detecting node has a path to send intrusion notifications to one of the sink nodes. VDCTA first partitions the belt-shaped region B of interest into m Voronoi regions, where m is the number of sink nodes. It then uses the modified Dijkstra algorithm, which is an all-source single-destination shortest path algorithm, to derive a shortest path (*minimum-hop path*) from every detecting node to the sink node for each partition of B by setting the weight of every edge to be 1. The pseudo code of VDCTA is shown in Fig. 2.

Fig. 3 shows an example of VDCTA's execution for a WSN that contains 4 sink nodes, 38 sensor nodes, and 24 detecting nodes to form 3-barrier coverage. VDCTA first partitions the region monitored by the WSN into 4 Voronoi regions. It then runs the modified Dijkstra shortest path algorithm to construct 4 convergecast trees such that every detecting node has a minimum-hop path to reach a sink node.

Voronoi Diagram Convergecast Tree Algorithm (VDCTA)

Input: B, V_s, V_d, V_k

Output: Convergecast Trees

1. Obtain the Voronoi diagram of sink nodes in V_k to partition belt-shaped region B into m Voronoi regions B^1, \dots, B^m , where $m=|V_k|$
2. Derive the associated communication graph $G_c^i = (V^i, E_c^i)$ for each Voronoi region $B^i, 1 \leq i \leq m$. Note that $V^i = V_s^i \cup \{sk^i\}$, where V_s^i is the set of sensor nodes in B^i and sk^i is the sink node in B^i .
3. Run the modified Dijkstra algorithm for each communication graph $G_c^i = (V^i, E_c^i), 1 \leq i \leq m$ to derive the shortest path from every detecting node in V_s^i to the sink node sk^i by setting the weight of each edge in E_c^i to be 1. Obtain a convergecast tree T^i associated with G_c^i by merging all shortest paths derived.
4. Return $T^i, 1 \leq i \leq m$.

Fig.2: The Voronoi Diagram Convergecast Tree Algorithm (VDCTA)

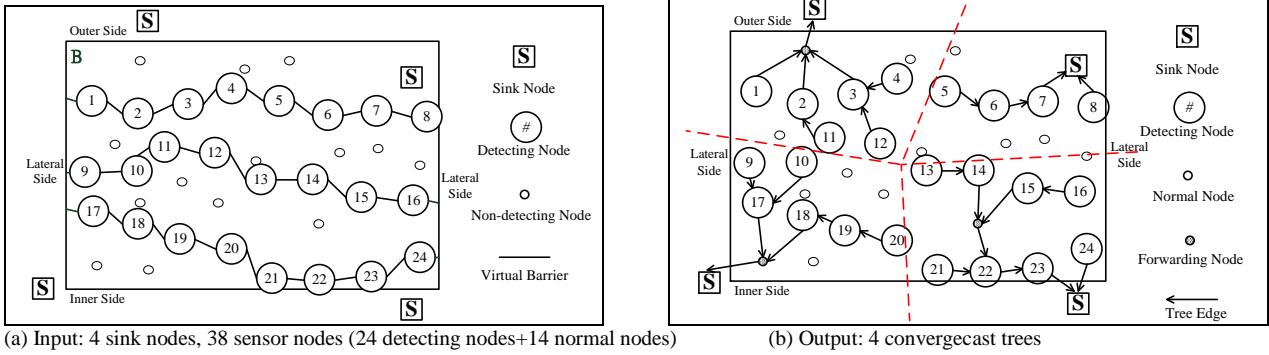


Fig.3: An example of the execution of VDCTA

The Channel And Timeslot Scheduling Algorithm (CATSA)

Channel And Timeslot Scheduling Algorithm (CATSA)

Input: Convergecast trees $T^i, 1 \leq i \leq m$

Output: Convergecast tree T^i with channel and timeslot assignment, $1 \leq i \leq m$

1. Derive the associated conflict graph $G_f^i = (V_f^i, E_f^i)$ for each convergecast tree $T^i, 1 \leq i \leq m$, where $V_f^i = sk^i \cup \{v | v \text{ is a receiver node in } T^i\}$.
2. For each conflict graph $G_f^i = (V_f^i, E_f^i)$, put every node into priority queue Q^i according to the criterion that a larger degree implies a higher priority.
3. For each priority queue $Q^i, 1 \leq i \leq m$, do channel assignment as follows: repeat to remove the top-priority element h from Q^i and assign channel c to h , where c is the smallest number of unused channels among h 's neighbors in G_f^i , until Q^i is empty. If no such unused channel exists, just assign h the channel used by either h 's parent node or child node in T^i .
4. For each convergecast tree $T^i, 1 \leq i \leq m$, derive the breadth first traversal (BFT) sequence S^i .
5. For each sequence $S^i, 1 \leq i \leq m$, do timeslot assignment (scheduling) as follows: repeat to (a) obtain the channel c used by the first element v of S^i , (b) remove from S^i all elements (nodes) that are receivers using channel c , (c) find all nodes in tree T^i that are the transmitters of the removed receiver nodes, (d) form a forest for all the transmitters and receivers, and (e) for each tree r in the forest, assign the smallest timeslots of 1, 2, ... to the transmitters according to r 's BFT sequence under the condition that a node should not be assigned the timeslot already assigned to its parent, until all nodes except for the sink node in tree T^i has been assigned a timeslot.
6. Return T^i with channel and timeslot assignment, $1 \leq i \leq m$.

Fig.4: Channel And Timeslot Scheduling Algorithm (CATSA)

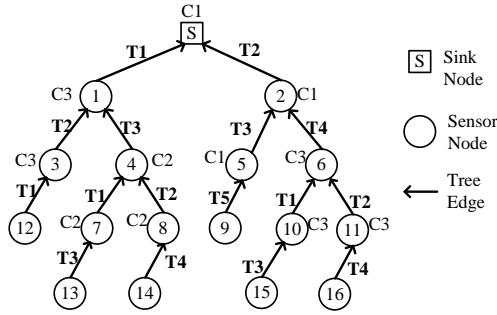


Fig. 5: An example of the execution of CATSA on a convergecast tree, in which C# stands for the channel assigned to receivers and T# stands for the timeslot assigned (scheduled) to transmitters

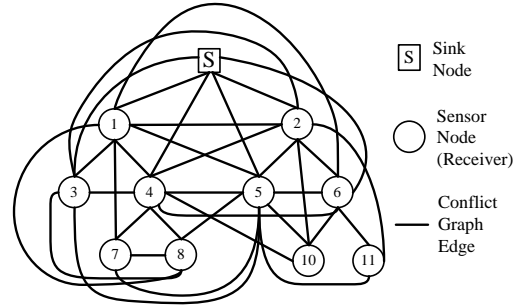


Fig. 6: The receiver-based conflict graph of the convergecast tree given in Fig. 5.

The Channel and Timeslot Scheduling Algorithm (CATSA) performs receiver-based channel assignment and transmitter-based timeslot assignment (scheduling). It first assigns channels to convergecast tree nodes based on receiver-based conflict graph. For a convergecast tree T^i , CATSA derives its associated receiver-based conflict tree by adopting the protocol interference model. It then assigns channels according to receivers of higher degrees with higher priorities. Afterwards, CATSA derives the breadth first traversal (BFT) sequence of the convergecast tree. According to the sequence, all transmitters using the same channel and associated receivers are grouped together to form a forest. Finally, each transmitter of every tree is sequentially assigned a timeslot numbered by 1, 2, ... that is not used by its parent node in the convergecast tree according to the BFT sequence. Fig. 4 shows the pseudo code of CATSA.

Fig. 5 shows an example of CATSA's execution for a convergecast tree that is rooted at a sink node and contains 16 sensor nodes. The number of channels assigned to receivers is limited to be 3 (a pre-specified constraint) and the number of timeslots assigned to transmitters is 5. Fig. 6 is the receiver-based conflict graph associated with the convergecast tree given Fig. 5. Note that to list the nodes in the conflict graph according to the highest-degree to the lowest-degree order, we have 5, 4, 1, 2, 3, 6, S, 11, 7, 8, 10.

Simulation

On the basis the IEEE 802.15.4 protocol [2], the proposed algorithms are simulated with OPNET and MATLAB tools and compared with the ONSA algorithm [5]. IEEE 802.15.4 supports three kinds of networks, namely *star*, *tree*, and *mesh* networks, among which we adopt the *tree network* to form a WSN. A tree network has only one *coordinator* and multiple *routers*. The coordinator is responsible for initializing and controlling the network; we assume that a WSN sink node plays the role of a network coordinator. A router can communicate with another router or coordinator in a multihop manner; we assume WSN sensor nodes play the role of routers.

The coordinator defines the *superframe* structure consisting of two portions: *active portion* and *inactive portion*. The active portion consists of 16 slots of the same length, which can be further partitioned into a *contention access period (CAP)* and a *contention free period (CFP)*. The coordinator can assign *guarantee time slots (GTSs)* in the contention free period.

This paper assumes a superframe has neither inactive portion nor CAP for performing simulation. It also assumes the CFP occupies 15 slots and the CAP occupies 1 slot which is reserved for a router to announce a *beacon* to start a superframe to be followed by the router's child router. It also adopts the revised version of IEEE 802.15.4 released in 2006 [2], in which a router can select one active portion as its *outgoing superframe*, and rely on the beacon sent by the parent router to start the *incoming superframe*. The superframe structure specified by the coordinator (the sink node) should accommodate t outgoing superframes and t incoming superframes, where t is the number of timeslots assigned to transmitters by CATSA. In the simulation, we restrict t to be 15.

Under the above-mentioned settings, we evaluate the performance of the proposed algorithms and the ONSA algorithm for the sake of comparison with OPNET and MATLAB tools. The overall simulation settings are listed in Table I. There are 204 sensor nodes which can form 7-barrier coverage over the network region. The simulation period is 300 seconds, during which 50 intruders choose a random start time to cross the network region by following a linear trajectory. For each intruder, 14 detecting nodes will detect it and each sends out intrusion notifications repeatedly for 10 seconds.

Table 1: Simulation settings

| | | | |
|---------------------------------------|--------------|-----------------------------|---------|
| Network Dimension | 100 m x 40 m | Simulation Period | 300 (s) |
| Number of Deployed Sensors | 204 | Sensing Range | 10 m |
| Number of Available Channels | 16 | Transmission Range | 20 m |
| Number of Available Time Slots | 15 | Sensor Distribution | Uniform |
| The Degree of Barrier Coverage | 7 | Number of Sink Nodes | 4 |

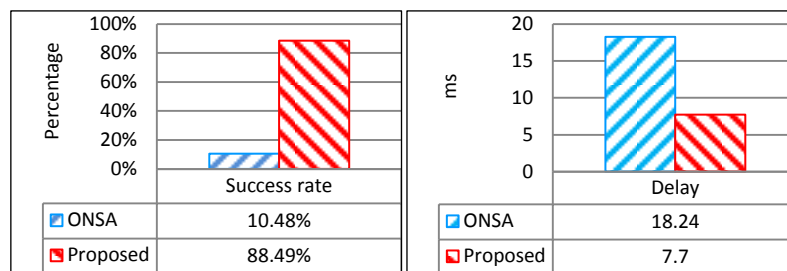


Fig.7: Simulation Results (left: notification success rate; right: notification delay)

As shown in Fig. 7, the proposed algorithms have the much better success rate and delay than ONSA, where the success rate is the ratio of the number of transmitted intrusion notifications over the number of notifications received by the sink node and the notification delay is the time for the notification to travel from the detecting node to the sink node. This is because the proposed algorithms use multiple channels and multiple timeslots to reduce interference among notification transmissions, while ONSA uses only the CSMA/CA mechanism to transmit notifications and suffers from interference.

Conclusion

In this paper, we have proposed two algorithms, namely VDCTA and CATSA, for a WSN with detecting nodes forming barrier coverage to achieve two goals: (1) forming convergecast trees to ensure every detecting node has a routing path for sending intrusion notifications to one of the sink nodes, and (2) assigning channels and timeslots to convergecast tree nodes to avoid notification interference as much as possible. Moreover, we have simulated the proposed algorithms and ONSA [5] based on the IEEE 802.15.4 protocol with OPNET and MATLAB for the sake of performance comparisons. By the simulation results, we can observe the proposed algorithms significantly outperform ONSA in terms of the notification success rate and the notification delay. In the future, we plan to investigate the implementation issues of the proposed algorithms and to apply genetic algorithms for achieving the same two goals.

References

- [1] Ghosh, A., Incel, O. D., Kumar, V. S. A. & Krishnamachari, B. Multi-channel scheduling algorithms for fast aggregated convergecast in sensor networks. *Proc. of the 6th IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp.363-372, 2009.
- [2] *IEEE standard for information technology-- local and metropolitan area networks-- specific requirements-- part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (WPANs)*, 2006.

- [3] Koubâ, A., Cunha, A., Alves, M. & Tovar, E. A time division beacon scheduling mechanism for ZigBee cluster-tree wireless sensor networks. *Proc. of the 19th Euromicro Conference on Real-Time Systems*, pp.125-135, 2007.
- [4] Kumar, S., Lai, T. H. & Arora, A. Barrier coverage with wireless sensors. *Proc. of the 11th annual international conference on Mobile computing and networking*, pp.284-298, 2005.
- [5] Lai, Y. L., & Jiang, J. R. Sink-connected barrier coverage optimization for wireless sensor networks. *Proc. of the 7th International Conference on Wireless and Mobile Communications*, pp.198-203, 2011.
- [6] Lai, Y. L., & Jiang, J. R. Barrier Coverage with Optimized Quality for Wireless Sensor Networks. *Proc. of the 15th International Symposium on Wireless Personal Multimedia Communications Symposium*, pp.192-196, 2012.
- [7] Pan, M. S. & Tseng, Y. C. Quick convergecast in ZigBee beacon-enabled tree-based wireless sensor networks. *Comput. Commun.* 31(5), pp.999-1011, 2008.
- [8] Rawat, P., Singh, K. D., Chaouchi, H. & Bonnin, J. M. Wireless sensor networks: a survey on recent developments and potential synergies. *The Journal of Supercomputing*, 68(1), pp.1-48, 2014.
- [9] Ru, A. B., Dobrota, V., Vedinas, A., Boanea, G. & Barabas, M. Modified Dijkstra's algorithm with cross-layer QoS. *ACTA TECHNICA NAPOCENSIS, Electronics and Telecommunications*, 51(3), pp. 75-80, 2010.
- [10] Saifullah, A., Xu, Y., Lu, C. & Chen, Y. Distributed channel allocation protocols for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(9), pp.2264-2274, 2013.
- [11] Saipulla, A., Westphal, C., Liu, B. & Wang, J. Barrier coverage of line-based deployed wireless sensor networks. *Proc. of INFOCOM 2009*, pp.127-135, 2009.
- [12] Shi, Y, Hou, Y. T., Liu, J. & Kompella, S. How to correctly use the protocol interference model for multi-hop wireless networks. *Proc. of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '09)*, pp.239-248, 2009.