Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 241208, 16 pages
http://dx.doi.org/10.1155/2015/241208

Research Article

Exploiting Multiple Channels for Low Latency and Semireliable Broadcasting in Cognitive Wireless Sensor Networks

Tae-Sung Kim,[1] Sok-Hyong Kim,[2] Bo-Kyum Kim,[3] and Young-Yong Kim[1]

[1] *Yonsei University, 705 3rd Engineering Building, 50 Yonsei-ro, Seodaemun-gu, Seoul 120-749, Republic of Korea*
[2] *Telecom Sys., Samsung Electronics, Suwon, Republic of Korea*
[3] *Mobile Communications, LG Electronics, Seoul, Republic of Korea*

Correspondence should be addressed to Tae-Sung Kim; xinia0214@naver.com

Received 17 April 2015; Revised 5 August 2015; Accepted 10 August 2015

Academic Editor: Shamik Sengupta

In Wireless Sensor Networks (WSNs), disaster management is a crucial issue that focuses on disaster relief and recovery. Mobile sensor nodes support disaster relief and recovery by means of real-time bidirectional communication. For its high data rate requirement, IEEE 802.11 specification can be used for the radio interface of sensor nodes, and the nodes can be equipped with multiple 802.11 radios to utilize multiple channels and link data rates. Channel assignment algorithms can be applied in cognitive radio enabled networks which performs dynamic channel configuration for utilizing multiple channels. For efficient and semireliable broadcast in cognitive radio WSNs, we focus on reducing broadcast latency and achieving 100% delivery percentage. To realize these goals, in this study, we present our design for a novel *Channel Assignment Algorithm for a Collision-Reduced Broadcast Tree* (CA-CBT). Fundamentally, CA-CBT builds a broadcast tree and then uses several heuristic procedures to allocate collision-free channels to links on the tree. If CA-CBT fails to allocate collision-free channels due to a limited number of available channels, it allocates non-collision-free channels with the smallest number of interfering nodes. Through extensive simulations, we demonstrated that CA-CBT supports lower broadcast latency and higher delivery percentages compared with existing broadcast algorithms.

1. Introduction

Disaster management has been widely studied in wireless sensor networks [1]. Conventional studies primarily focused on disaster detection and efficient reporting from sensing nodes to a sink node; however, recent studies have expanded to cover disaster relief and recovery. For example, mobile sensor nodes, such as those embedded on robots, may be deployed to dangerous environments, including areas afflicted by toxic gas leaks, radiation exposure, and earthquakes. These nodes download data from a sink node such as a gateway or command center and then communicate with nearby sensor nodes to support disaster relief and recovery by means of real-time bidirectional communication. In wireless sensor networks (WSNs), each mobile sensor node can be equipped with multiple radio interfaces to deliver large or small sensing data [2–4].

Designing channel assignment algorithms to exploit multiple channels is similar to the concept of cognitive radio in that multiple available channels are utilized. Hence, our research considers cognitive radio WSNs with multiple channels for transporting large data from a sink node to mobile sensor nodes.

In multichannel WSNs, nodes can utilize multiple channels available within 802.11 specifications via multiple interfaces to deliver traffic in parallel. When the number of nodes increases, nodes require more channels to enable more concurrent transmissions over different channels; however, 802.11 supports only a limited number of channels. For example, 802.11b/g and 802.11a support three and 12 nonoverlapping channels, respectively. Such limitations force nodes to share the common channel, which increases network interference and significantly degrades networks performance. To solve this problem, channel assignment algorithms have been extensively explored [5–11], all of which primarily focus on unicast transmissions.

In addition to multiple channels, off-the-shelf 802.11 devices also provide a multirate function. Depending on the channel condition, a node can transmit its data frame over a wireless link by using multiple data rates. In multirate networks, there is a tradeoff between the transmission range and the data rate (or latency) of links. For example, when the data rate of a link decreases, the transmission range of the link increases. In wireless networks, a packet that is broadcast by a sender can be received by all neighbor nodes within transmission range of the sender (assuming omnidirectional antennas), a phenomenon known as *wireless broadcast advantage* (WBA) [12]. WBA stems from the broadcast nature of the wireless channel; hence, broadcasting in multirate networks can exploit WBA by controlling the tradeoff between the transmission range and the data rate of links. Although this tradeoff is primarily utilized for unicast transmissions, we can also make use of the tradeoff for broadcast transmissions as [13–19].

To satisfy the requirements, both *efficient* and low latency broadcasting are critical. To address these requirements, several solutions have been proposed [13–19], all of which assume a proper channel assignment of links. As such, these studies focus on constructing a broadcast tree rather than assigning available channels to links on the broadcast tree.

Therefore, our key research objective was to design a channel assignment algorithm for efficient and semireliable broadcasting in WSNs. Our proposed algorithm assigns channels to links on the broadcast tree by fully utilizing all available channels supported by 802.11 devices. For efficient and semireliable broadcast, we use *broadcast latency* and *delivery percentage* as metrics, as defined in [13]. Broadcast latency is defined as the elapsed time between the broadcast source sending a packet and all nodes in the network receiving it. More specifically, the minimum latency broadcast problem has been proven to be NP-hard due to the complexity in multichannel multiradio multirate networks [20].

Delivery percentage is defined as the number of nodes that receive a packet (sent by the broadcast source) divided by the total number of nodes in the network. Improving the delivery percentage is also a challenging problem since the delivery percentage metric reflects the collision degree of broadcast packets on the channels. In particular, 802.11 does not use the retransmission scheme for broadcast packets when the broadcast packets experience collisions caused by the hidden terminal problem, whereas unicast packets take advantage of this retransmission scheme. When transmitting a stream of broadcast packets, the delivery percentage decreases more significantly. Hence, our interest here lies in designing a channel assignment algorithm that makes full use of available channels to reduce broadcast latency and achieve up to 100% delivery percentage.

Given the above, in this paper, we propose a centralized *channel assignment algorithm for a Collision-Reduced Broadcast Tree* (CA-CBT) in WSNs. CA-CBT builds a shortest path tree by using the path cost metric and Dijkstra's algorithm. Next, CA-CBT traverses each node for the channel assignment of a parent node and its child nodes. CA-CBT attempts to assign channels that guarantee collision-free broadcasting by using three heuristic procedures. When the number of available channels is small, the number of nodes assigned with non-collision-free channels increases on the tree and those nodes can be influenced by interfering nodes. Thus, if CA-CBT fails to guarantee collision-free broadcasting, it then tries to assign channels that do not guarantee collision-free broadcast but have the smallest number of interfering nodes. Further, to improve the delivery percentage, CA-CBT compensates those nodes that could experience collisions by allowing them to receive redundant broadcast packets from one more parent node. In contrast, when the number of available channels is not small, the possibility of assigning non-collision-free channels is small and thus the delivery percentage can be increased. Since CA-CBT fully utilizes all available channels with a limited number of NICs, it is beneficial when the number of channels is larger than the number of NICs per node.

Through extensive ns-2 simulations, we showed that CA-CBT was able to achieve lower broadcast latency and higher delivery percentages as compared with existing broadcast algorithms.

In addition to this introductory section, the remainder of this paper is structured as follows. In Section 2, we survey related work. In Section 3, we define our research objective and provide a system model and several terms used in CA-CBT. Next, we propose our detailed CA-CBT in Section 4. In Section 5, we evaluate the performance of CA-CBT and compare it with existing broadcast algorithms. Finally, we conclude our paper and suggest directions for future work in Section 6.

2. Related Work

Recently, several algorithms have been suggested for broadcasting in WSNs. These algorithms can be classified into centralized algorithms [13–16] and distributed algorithms [17, 18]. In centralized algorithms, we assume a centralized entity, such as a gateway node, collects global WSN information from all nodes. For instance, in [13], Qadir et al. developed a locally parallelized multiradio multichannel weighted connected dominating set (WCDS) tree (LMT) and a parallelized approximate-shortest multiradio, multichannel WCDS tree (PAMT). Both LMT and PAMT build broadcast trees by specifying four tuples consisting of the transmitting node, the receiving nodes, the latency, and the channel. In LMT, the algorithm attempts to decrease the number of contending transmissions on the common channel. In PAMT, the algorithm tries to increase parallelization (i.e., by activating more interfaces).

In [16], Li et al. jointly formulated the broadcast routing and channel assignment problem using a mixed-integer linear programming approach. They then proposed a heuristics-based broadcast algorithm. Conversely, distributed algorithms require nodes to exchange control messages with neighboring nodes to utilize local information regarding the WSN. For example, the multiradio distributed tree (MRDT) approach [17] uses RREQ and RREP messages to calculate a low latency broadcast tree by considering interface, channel, and rate diversities.

In the literature, several algorithms have been proposed that make use of breath-first search (BFS) for broadcasting in multihop wireless networks [20–22]. In the BFS-based broadcast algorithms, receiving nodes may experience packet collisions when a number of nodes are transmitting broadcast packets in a single-channel single-rate wireless network. To combat this problem, a time slot is defined in the algorithms as the time duration between the time when the source node transmits a broadcast packet and the time when the receiver nodes receive the broadcast packet. The algorithms then schedule time slots to minimize the number of such slots required for transmitting broadcast packets.

Our main interest lies in designing a channel assignment algorithm for broadcasting in WSNs. In most of the aforementioned algorithms, the focus has been on the construction of a broadcast tree rather than the channel assignment algorithm itself. That is, after assuming channels have been assigned to available interfaces, they construct the broadcast tree by using available links. Regarding channel assignments, in [13], Qadir et al. concluded that the common channel approach (CCA) is a good strategy for broadcasting among existing channel assignment schemes, which are designed for unicast flows, such as CCA [9], the varying channel approach (VCA) [10], and interference survivable topology control (INSTC) [11].

In CCA, the number of equipped interfaces per node is equal to the number of available channels, and each interface is assigned a distinct channel. Thus, each node has full connectivity to neighboring nodes within its transmission range. For VCA, nodes use one NIC on a common channel to ensure network connectivity, making use of the other remaining NICs on random channels. INSTC performs channel assignments to reduce network interference. According to [13], CCA works well for broadcasting since it is simple and adjacent nodes can utilize a common set of channels; however, when the number of NICs per node is smaller than the number of available channels, the channel diversity of CCA is restricted to the number of equipped NICs, which may lead to low broadcast performance in WSNs.

3. Research Objective, System Model, and Definitions

3.1. Research Objective and System Model. Our research objective was to design a channel assignment algorithm to support efficient and semireliable broadcasting in WSNs. We consider both efficient broadcasting by reducing broadcast latency and reliable broadcasting by achieving up to 100% delivery. To achieve our goals, we developed a channel assignment algorithm that assigns channels to links in WSNs and fully utilizes all channels with a small number of NICs for broadcasting in WSNs.

We assumed a multichannel multiradio multirate WSN in which nodes are equipped with multiple (at least two) IEEE 802.11 NICs. Although our proposed algorithm can be used in WSNs in which the number of available channels is larger than the number of NICs per node, we considered only IEEE 802.11a-based WSNs that support 12 noninterfering channels and data rates from 6 Mbps to 54 Mbps.

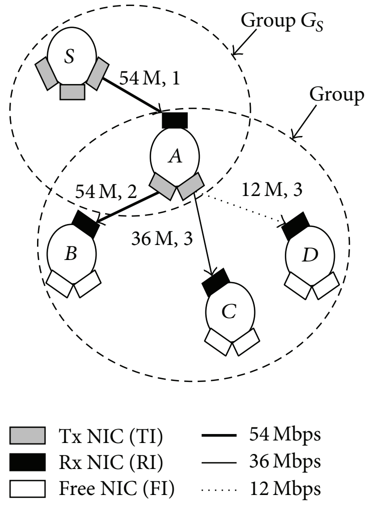


Figure 1: An example network setup depicting definitions used in CA-CBT.

We also assumed the mutual interference model for our network interference model; that is, two transmitting nodes A and B interfere with one another if the following two conditions are met: (1) both nodes A and B transmit at the same time on the common channel and (2) the receiving nodes of node A's transmission are in the interference range of node B and vice versa.

3.2. Definitions. In this subsection, we define several terms used in CA-CBT. Fundamentally, CA-CBT constructs a spanning tree rooted at the gateway node that serves as the broadcast source of the tree. Nodes use available interfaces in three distinct ways, namely, a *receiving NIC* (RI), a *transmitting NIC* (TI), and a *free NIC* (FI). The RI is used to receive broadcast packets from one or two parent nodes. The TI is used to transmit the received packet to its child nodes. All remaining NICs except for RIs and TIs become FIs. Further, each node can use one or more TIs and RIs, with TIs and RIs on the same node able to be assigned to different channels. Depending on the channel assignment algorithm, a node can transmit broadcast packets to its child nodes in parallel by using multiple TIs and link data rates. The node can also receive broadcast packets from one or two parent nodes through multiple RIs.

Figure 1 shows an example that consists of a broadcast source (node S), a parent node (node A), and three child nodes (nodes B, C, and D) of node A. Depending on link data rates, links are depicted differently; that is, each link is labeled by the broadcast rate and channel number. For example, links A-B, A-C, and A-D can use 54 Mbps, 36 Mbps, and 12 Mbps, respectively. Each node is equipped with three NICs, and parent node A uses two TIs and one RI. One TI transmits broadcast packets on channel two using 54 Mbps, while the other TI transmits broadcast packets on channel three using

12 Mbps. Here child nodes C and D are associated with one TI on parent node A via link A-C (36 Mbps) and link A-D (12 Mbps), respectively. Fundamentally, the transmission range of low data rates is larger than that of high data rates. Therefore, parent node A uses the broadcast rate for broadcasting to its child nodes, which is the lowest data rate among the data rates of links A-C and A-D.

For channel assignments, we define *group* G_X to include one parent node X (i.e., the transmitting node) and its child nodes (i.e., receiving nodes). The group can use multiple *group channels*. For example, in Figure 1, parent node A and its child nodes B, C, and D form group G_A on group channels two and three. In general, group G can be in either of two states, namely, *single-TX* or *multi-TX*, depending on the number of group channels used by group G. For example, in Figure 1, when group G uses multiple group channels as G_A, group G is in the multi-TX state. In contrast, when group G uses only one group channel as G_S, group G is in the single-TX state.

According to channel conditions, we define available channels as either *collision-free channels* (*CF-channels*) or *non-collision-free channels* (*CN-channels*). If there is no interfering node on a channel when the parent node transmits a broadcast packet to its child nodes in group G, the channel is a CF-channel for group G; otherwise, it is a CN-channel for group G. During channel assignments, the channel assignment algorithm maintains the following information for each group: a set of CF-channels (K_{CF}) and CN-channels (K_{CN}), as well as a set of its child nodes (N_C) and descendant nodes (N_D). For instance, in Figure 1, assume that there are four noninterfering channels; for parent node A in group G_A, $K_{\text{CF}}(A) = \{\text{chs } 2, 3, \text{and } 4\}$, $K_{\text{CN}}(A) = \{\text{ch } 1\}$, and $N_C(A) = N_D(A) = \{B, C, \text{and } D\}$.

4. *Channel Assignment Algorithm for a Collision-Reduced Broadcast Tree* (CA-CBT)

In this section, we describe the design of our centralized channel assignment algorithm, CA-CBT, in relation to WSNs. We start by describing the centralized operation to gather global network information from nodes in a WSN, which is required for the channel assignment algorithm of CA-CBT. After describing the tree construction algorithm, we then present the main portion of CA-CBT in detail.

4.1. Centralized Operation. Since CA-CBT uses a centralized algorithm, the gateway node gathers network information from nodes of a WSN; such information is required for the algorithm of CA-CBT. Initially, all nodes operate on the common channel by using one of their NICs. To obtain each neighbor node's information, nodes (say node X) exchange HELLO messages with their neighboring nodes (say node Y) within H-hop range. This neighbor's information of node X includes the data rate of link X-Y, the addresses of one-hop neighbor nodes around node X, and the address of interfering nodes within the interference range of node X. Here, H-hop is the ratio between the interference range and the transmission range. Since the H-hop value is small, nodes do not flood HELLO messages network-wide.

After obtaining a neighbor's information, nodes report such information to the gateway on the common channel by using a well-known routing protocol, such as AODV [23]. Once the network information is gathered, the gateway node performs the channel assignment algorithm of CA-CBT that determines the channel of links and routing paths on the broadcast tree. To notify all nodes of the algorithm's results, the gateway broadcasts *Advertisement* messages on the tree. Then, nodes that received the message are able to assign channels to their NICs and update routing tables accordingly.

4.2. Tree Construction. For broadcast in WSNs, Chou et al. [14] proved that transmitting broadcast packets at a higher data rate is better than at a lower data rate. Moreover, in [13], Qadir et al. insisted that a low data rate link increases broadcast latency over some links, that is, those links that can transmit faster without it. Therefore, we utilize high-rate links instead of low-rate links wherever possible during tree construction.

CA-CBT constructs a tree much like a shortest path tree by using a path cost metric. We adopted a similar approach as that of Dijkstra's algorithm in that each node maintains a measure from the broadcast source (i.e., the gateway node). Here, when link i uses data rate r_i, the approximated transmission time of link i can be expressed as $1/r_i$. Then, *path cost* is defined as the sum of the approximated transmission time of links along the path between the broadcast source and a node; that is, for node X on the tree, path cost $C(X)$ is given as

$$C(X) = \sum_{\text{link } i \in P_X} \frac{1}{r_i}, \tag{1}$$

where P_X is the path between the gateway and node X.

Algorithm 1 shows the pseudocode of the tree construction algorithm in CA-CBT. We assume that network information is collected by using a centralized operation, as noted above in Section 4.1. Initially, the status of all nodes is set to "not updated" and the path cost value of all nodes is set to infinity (lines 1-2). Next, the algorithm traverses each node from the broadcast source node (i.e., the gateway node) via lines 5–14. Assume that broadcast source node S has two neighbor nodes X and Y as its candidate child nodes. For current node S, if node S provides the minimum $C(X)$ for its neighbor nodes X and Y (line 7-8), it updates the path cost $C(X)$ of nodes X and Y and then sets itself as the parent of nodes X and Y. Next, the status of neighbor nodes X and Y is set to "updated" (line 9). After the algorithm has visited all neighbor nodes of current node S, it sets the status of node S to "finished" and visits the next node with a minimum $C(X)$ value among the remaining nodes with the "updated" status. This process is repeated until the $C(X)$ value of all nodes is calculated. Since each of the N nodes is iterated with N_B neighbor nodes on average, the computational complexity of the tree construction is $(N \cdot N_B)$ on average.

```
Tree_Construction (network information) {
  Set the status of all nodes to "not updated";
  Set the path cost value of all nodes to "MAX";
  curr_node = $S$;
  Set the path cost value of curr_node to zero;
  /* Update the path cost metric of all nodes with min $C(X)$ value */
  for each curr_node cn do {
      for each curr_neigh_node cnn of cn do {
          if (cn provides min $C(X)$ value for cnn) then {
              Update the path cost value and the parent node of cnn;
              Set the status of cnn to "updated";
          }
      }
      Set the status of cn to "finished";
      curr_node (cn) = get_min_node("updated");
  }
} /* End of Tree Construction */
```

Algorithm 1: Tree construction.

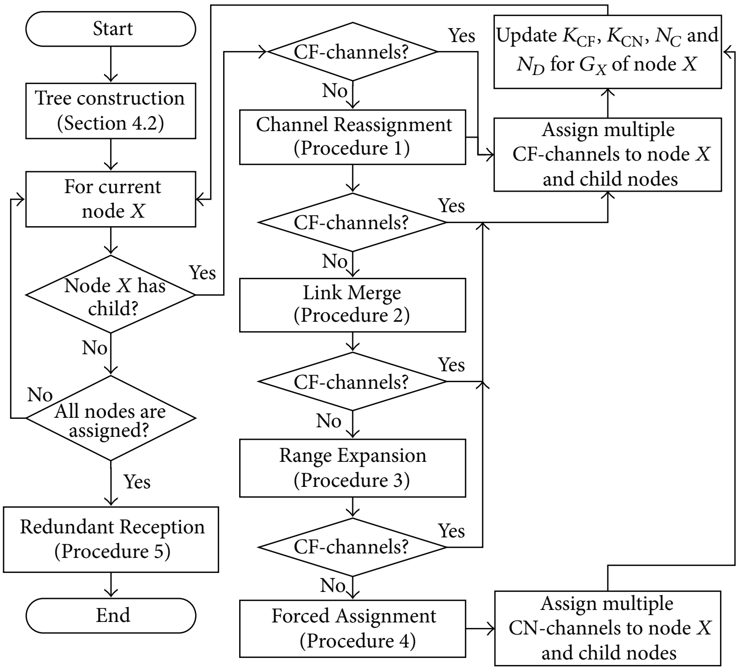

Start
Tree construction (Section 4.2)
For current node X
Node X has child?
Yes
No
All nodes are assigned?
No
Yes
Redundant Reception (Procedure 5)
End
CF-channels?
Yes
No
Channel Reassignment (Procedure 1)
CF-channels?
Yes
No
Link Merge (Procedure 2)
CF-channels?
Yes
No
Range Expansion (Procedure 3)
CF-channels?
Yes
No
Forced Assignment (Procedure 4)
Update K_{CF}, K_{CN}, N_C and N_D for G_X of node X
Assign multiple CF-channels to node X and child nodes
Assign multiple CN-channels to node X and child nodes

Figure 2: The entire procedure of the CA-CBT algorithm.

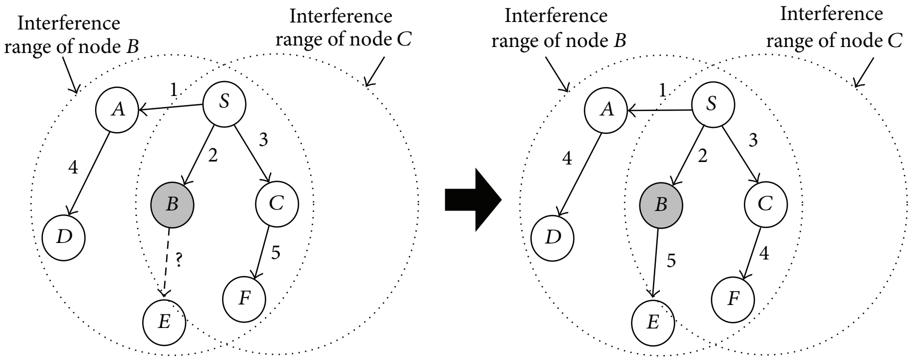
4.3. Channel Assignment Algorithm. In this subsection, we propose the main algorithm of CA-CBT. Figure 2 illustrates the entire procedure of CA-CBT. After a tree is constructed (i.e., Section 4.2), CA-CBT performs channel assignments for links and determines routing paths on the tree by using the five procedures labeled from P1 to P5. The algorithm uses BFS to traverse nodes on the tree from the broadcast source node to the leaf nodes. The path cost (C) value of nodes determines the traversal order, and at each iteration, the node with a minimum C becomes the first node of the BFS.

For each traversal, the algorithm visits the current visited node (say node X), allocating one or more group channels to the group that includes current node X and its child nodes. Since the number of available channels is limited in IEEE 802.11, CF-channels may or may not be available. Depending on the availability of CF-channels, channel assignments fall into one of the following three cases.

Case 1. When node X can use multiple TIs and there are enough CF-channels for each TI, node X can use multiple CF-channels via TIs for collision-free broadcasting. As a result, parent node X of group G_X is in the multi-TX state. Since group G_X may include multiple child nodes, TIs and child nodes must be bound together. To handle this, we specified a *Rate-TI binding rule* as follows. Suppose node X has M TIs, M CF-channels, and N_L links for N_L child nodes. At first,

```
(0) Channel_Reassignment (current node $B$) {
(1)   for each interfering group $G_i$ around current node $B$ do {
(2)      Group $G_i$ changes its CF-channels to other CF-channels temporarily;
(3)      if (current node $B$ can assign CF-channels) then {
(4)         Group $G_i$ changes its CF-channels to other CF-channels permanently;
(5)         Add the CF-channels to $K_{CF}(B)$;
(6)      }
(7)   }
(8)   Return $K_{CF}(B)$;
(9) } /* End of Channel Reassignment */
```

Algorithm 2: Channel Reassignment.



Interference range of node B
Interference range of node C
A
S
1
2
3
4
B
C
D
?
5
E
F
Interference range of node B
Interference range of node C
A
S
1
2
3
4
B
C
D
5
4
E
F

(a) Node B has no CF-channel for its child node E

(b) Nodes C and F switch to channel 4

Figure 3: An example illustrating Channel Reassignment (Procedure 1).

M TIs are assigned to M CF-channels. After the algorithm sorts N_L links in decreasing order of their link data rates, it calculates average rate S_{avg}, which is the data rate sum of N_L links divided by M TIs. Then, each link in the sorted links is assigned to the current TI_m until the data rate sum of links assigned to TI_m is larger than S_{avg}.

For example, in Figure 1, parent node A has two TIs and three links, that is, A-B (54 Mbps), A-C (36 Mbps), and A-D (12 Mbps). TI_1 and TI_2 on node A are assigned to channels two and three, respectively. Then, the sorted links are A-B, A-C, and A-D, and $S_{\text{avg}} = (54 + 36 + 12)/2 = 51$ Mbps. Hence, link A-B is assigned to TI_1 because 54 Mbps > $S_{\text{avg}} = 51$ Mbps, while links A-C and A-D are assigned to TI_2. Since the Rate-TI binding rule iterates N_L links for M TIs assigned to M CF-channels, it takes $O(M \cdot N_L)$.

Case 2. When node X can use multiple TIs but there is no CF-channel, CA-CBT obtains at least one CF-channel by using up to three iterations of *Channel Reassignment* (*P1*), *Link Merge* (*P2*), and *Range Expansion* (*P3*), each of which is discussed in this subsection. If CF-channels are obtained through these procedures, CA-CBT assigns the CF-channels to group G_X. In this case, CA-CBT guarantees collision-free broadcasting.

Case 3. If CA-CBT cannot obtain CF-channels by using the aforementioned procedures, it assigns CN-channels to group G_X by using *Forced Assignment* (*P4*). In this case, CA-CBT cannot guarantee collision-free broadcasting but improves delivery percentage performance through *Redundant Reception* (*P5*).

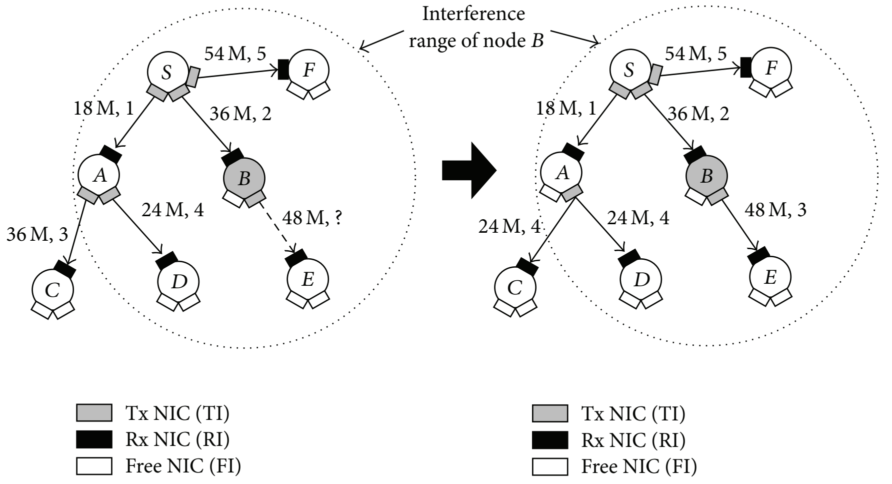
We describe the five aforementioned procedures below.

Procedure 1 (Channel Reassignment). Suppose that CA-CBT visits the current node (e.g., node X) and there is no CF-channel for group G_X. Then, CA-CBT performs *Channel Reassignment* (see Figure 2). The design goal of Channel Reassignment is to make interfering groups (e.g., group G_Y) around node X change their assigned group channels to other channels in the set of CF-channels (i.e., $K_{\text{CF}}(Y)$). This may return one or more CF-channels for group G_X. Here, the interfering group of node X is the group that interferes with the reception of nodes in group G_X.

The pseudocode of Channel Reassignment is shown as Algorithm 2. We explain this procedure by using the example shown in Figure 3. In this figure, solid lines are the links on the tree and dotted lines are the links not on the tree. Five channels are available, the current visited node is node B, and the dotted circles around nodes B and C represent the interference ranges of nodes B and C, respectively. In Figure 3(a), suppose that node B has no CF-channel for its child node E. Then, Channel Reassignment *temporarily* changes the CF-channels of the interfering group G_i around node B to the other CF-channels in $K_{\text{CF}}(i)$ (line 2). If node B can use at least one CF-channel, Channel Reassignment

```
(0)  Link_Merge (current node $B$) {
(1)    for each interfering group $G_i$ around node $B$ in the multi-TX state do {
(2)      Select two TIs with the minimum broadcast rate;
(3)      Calculate $C_D$ for the selected TIs on interfering group $G_i$;
(4)      Perform Link Merge by selecting one of the TIs temporarily;
(5)      if (current node $B$ can use CF-channels) then {
(6)        Set the status of the interfering group $G_i$ to "available";
(7)      }
(8)    }
(9)    Get the interfering group with minimum $C_D$ and "available" status;
(10)   Perform Link Merge for the interfering group permanently;
(11)   Set the state of the interfering group to single-TX;
(12)   Add the obtained CF-channels to $K_{CF}(B)$;
(13)   Return $K_{CF}(B)$;
(14) } /* End of Link Merge */
```

Algorithm 3: Link Merge.



(a) Node B has no CF-channel for its child node E

(b) Node A merges links A-C and A-D on the same TI

Figure 4: An example illustrating Link Merge (Procedure 2).

permanently changes the channels of interfering group G_i to the new CF-channels (line 4). For example, when group G_C switches from channel five to channel four, node B can use CF-channel five, as shown in Figure 3(b).

Note that Channel Reassignment does not affect interference on group G_C since interfering group G_C selects the other CF-channels in $K_{CF}(C)$. In this way, if Channel Reassignment returns one or more CF-channels (line 8), current node B can use the CF-channels for its child node E. Otherwise, CA-CBT performs the second procedure, called *Link Merge*, which guarantees collision-free broadcasting.

Assuming that N_I is the average number of interfering groups around the current node, Channel Reassignment iterates N_I interfering nodes for M TIs and K channels. Hence, its complexity is $O(M \cdot N_I + K \cdot N_I)$ for the current node.

Procedure 2 (Link Merge). CA-CBT uses Link Merge when there is no CF-channel for current node X after Channel Reassignment (see Figure 2). The design goal of this procedure is to change the state of interfering groups from multi-TX to single-TX. In other words, CA-CBT requests only the interfering groups in a multi-TX state to decrease their number of assigned group channels in the groups. As a result of Link Merge, some groups in the multi-TX state may change to a single-TX state, and thus the current node X can obtain CF-channels.

Algorithm 3 shows pseudocode for Link Merge, while Figure 4 shows an illustrative example of this procedure in which all nodes on the broadcast tree are depicted by solid links. Dotted links are the links not on the tree. Each link is labeled with the corresponding broadcast rate and channel number. Suppose that the current node is node B and comprises five channels, and the dotted circle is the interference range of node B. In Figure 4(a), node B cannot use any CF-channels for link B-E after Channel Reassignment (P1).

For Link Merge, CA-CBT requests the parent nodes of all interfering groups in a multi-TX state around node B to decrease the number of their group channels. First, the parent node of all interfering groups selects two TIs a and b with the two lowest broadcast rates (line 2). As shown in Figure 4(a), node S uses three TIs assigned to broadcast rates 54, 36, and 18 Mbps, respectively. It temporarily selects two TIs assigned to the two lowest broadcast rates of 36 and 18 Mbps. Second, CA-CBT calculates the *delayed cost* (C_D) metric for the current interfering group (line 3); for interfering group G_i, the delayed cost $C_D(i)$ is defined as

$$C_D(i) = \left| \frac{L}{R_{\mathrm{TI},a}} - \frac{L}{R_{\mathrm{TI},b}} \right| \cdot \left| N_D^I \right|, \quad (2)$$

where L is the packet size and $|N_D^I|$ is the number of descendant nodes that may experience increased broadcast latency due to Link Merge under the parent node of group G_i. Also, $R_{\mathrm{TI},a}$ and $R_{\mathrm{TI},b}$ are the broadcast rates of TI a and TI b, respectively.

In (2), the delayed cost metric indicates the increased broadcast latency that can be experienced by the descendant nodes of the parent node of group G_i. That is, a larger C_D leads to increased broadcast latency. For example, the interfering groups around node B are G_S, G_A, G_D, and G_F, as shown in Figure 4(a). Among them, G_S and G_A are in a multi-TX state. For group G_S, the set of influenced descendant nodes are nodes B and E. Hence, $C_D(S) = |L/36M - L/18M| \cdot 2 = 0.44$ ms by assuming $L = 1000$ bytes. Regarding interfering group G_A, $C_D(A) = |L/36M - L/24M| \cdot 1 = 0.11$ ms. For interfering groups G_D and G_F, the C_D values are set to zero since both nodes D and F have no child nodes.

After calculating the delayed cost (C_D) values, CA-CBT performs Link Merge for each interfering group *temporarily* (line 4). For example, in Figure 4(a), node A uses two TIs on channels three and four. To perform Link Merge, node A selects one of the two TIs with the minimum broadcast rate, which makes node A use only the TI with broadcast rate 24 Mbps on channel four. If current node B can use channel three as its CF-channel, CA-CBT sets the status of group G_A to "available" (line 6).

Once all interfering groups in the multi-TX state perform Link Merge temporarily, CA-CBT then selects the interfering group with the minimum C_D value among the interfering groups with an "available" status (line 9). For example, CA-CBT selects interfering group G_A with $C_D(A) = 0.11$ ms. Then, selected group G_A performs Link Merge and sets its state to single-TX *permanently* (lines 10-11). Finally, node B assigns CF-channel 3 as its child node E. If current node B cannot obtain any CF-channels, CA-CBT proceeds to *Range Expansion*, which ensures collision-free broadcasting.

Link Merge iterates over the interfering groups of the current node for M TIs and K channels. When the average number of interfering nodes is N_I around current node X, Link Merge takes $O(M \cdot N_I + K \cdot N_I)$ per the current node.

Procedure 3 (Range Expansion). CA-CBT performs Range Expansion when current node X has no CF-channels after the Channel Reassignment and Link Merge procedures (see Figure 2). The design goal of the Range Expansion procedure is to decrease the broadcast rate of TIs on the parent node of current node X so that the parent node expands its transmission range to cover more child nodes of node X on the tree. In this procedure, the parent node of current node X assigns existing CF-channels to the child nodes of node X, where the covered child nodes should not be interfered with by other nodes on the assigned CF-channels.

We explain Range Expansion by using the pseudocode of Algorithm 4 and the example shown in Figure 5. In the figure, nodes S, A, and B are on the tree depicted by solid lines, while dotted lines represent links not on the tree. Each link is labeled by the broadcast rate and channel number. Further, the dotted circle represents the interference range of node B.

In Figure 5(a), suppose that the current node is node B and that it has no CF-channels for links B-C and B-D after the Channel Reassignment (P1) and Link Merge (P2) procedures have been completed. In Range Expansion, current node B makes its parent node (e.g., node S) decrease the broadcast rate of TIs on parent node S. If parent node S uses multiple TIs, it selects the TI with the lowest broadcast rate among TIs that guarantee collision-free transmission for the child nodes of node B (lines 1-2). Then, parent node S covers all child nodes of current node B by decreasing the broadcast rate of the selected TI (lines 3–7).

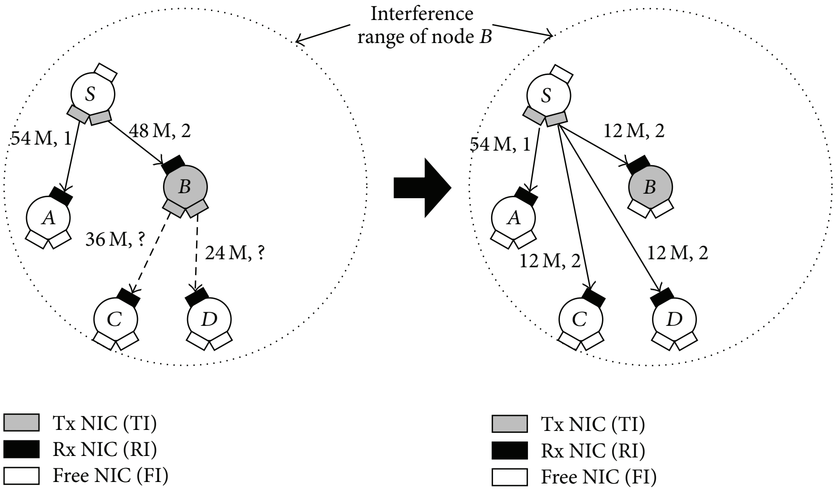
For instance, in Figure 5(a), parent node S of current node B selects the TI assigned to channel two at 48 Mbps, since the broadcast rate of the TI is lower than that of the other TI assigned to channel one at 54 Mbps (line 3). Next, node S decreases the selected TI's broadcast rate to cover all child nodes of node B (e.g., nodes C and D) as follows (lines 4–7). For simplicity, assume that the data rates of links S-C and S-D are 18 Mbps and 12 Mbps, respectively, and channels one and two are CF-channels. Then, the broadcast rate of the selected TI on parent node S is temporarily set to the lowest data rate, that is, 12 Mbps, as shown in Figure 5(b) (line 4). If parent node S can cover all child nodes of node B, it then sets the broadcast rate of the TI to the lowest data rate permanently (lines 5–7). In this way, CA-CBT can assign CF-channel two to child nodes C and D of current node B. If parent node S of node B cannot cover all child nodes of node B, CA-CBT proceeds to the fourth procedure, *Forced Assignment*, which does not guarantee collision-free broadcasting.

Range Expansion is only performed by the parent node of current node X for M TIs, K channels, and N_L links between the parent node and child nodes of the current node. Therefore, the complexity of Range Expansion is $O(M + K + N_L)$.

Procedure 4 (Forced Assignment). CA-CBT executes Forced Assignment when current node X cannot assign CF-channels for its child nodes after Channel Reassignment, Link Merge, and Range Expansion (see Figure 2). In Forced Assignment, CA-CBT assigns CN-channels with the smallest number of interfering nodes to the current group that comprises current node X and its child nodes. For CN-channel k, we define the total number of nodes in all interfering groups around current node X as $N_k^I(X)$. Then, CA-CBT selects CN-channel

```
***Range_Expansion*** (current node $B$) {
  Select a set of TIs on the parent of node $B$, which ensures collision-free;
  Select a TI with the minimum broadcast rate among the selected TIs;
  Get the min data rate of links between node $B$ and its child nodes;
  Set the min data rate to the broadcast rate of the selected TI temporarily;
  **if** (all child nodes of current node $B$ are covered by the parent) **then** {
      Set the min data rate to the broadcast rate of the TI permanently;
  }
} /* End of Range Expansion */
```

Algorithm 4: Range Expansion.



(a) Node B has no CF-channel for its child nodes C and D

(b) Node S expands TX range for the child nodes C and D of node B

Figure 5: An example illustrating Range Expansion (Procedure 3).

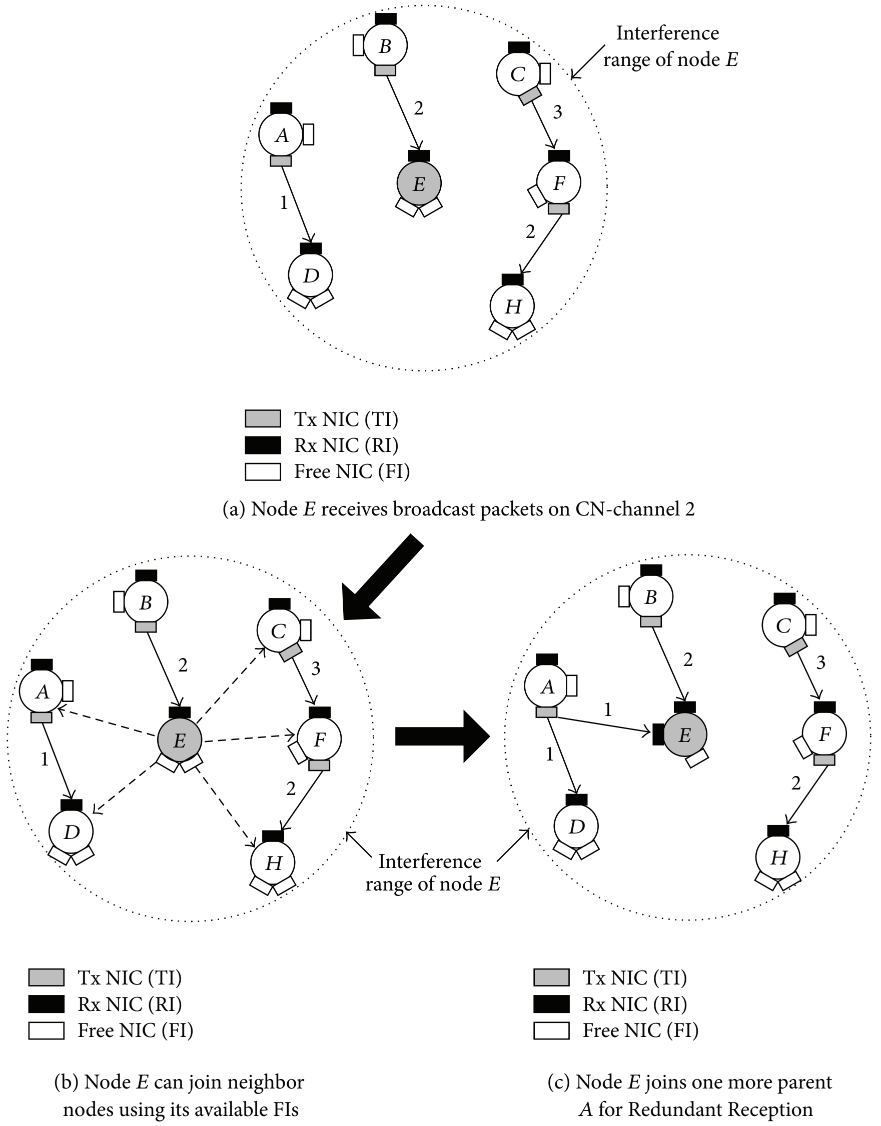
k with the smallest N_k^I value to evenly reduce the interference on CN-channels. Since Forced Assignment assigns CN-channels, the broadcast reception of the child nodes can be interfered with by broadcasting from other adjacent groups on the same CN-channels. In other words, Forced Assignment cannot guarantee collision-free broadcasting, and broadcasting on CN-channels may introduce packet loss due to collisions. To handle this issue, CA-CBT next performs the final procedure, *Redundant Reception*.

Since Forced Assignment calculates the N_k^I value for K CN-channels by iterating over the average number of N_I interfering nodes around current node X, the complexity here is $O(K \cdot N_I)$.

Procedure 5 (Redundant Reception). CA-CBT performs Redundant Reception after all nodes of the tree are assigned CF-channels or CN-channels (see Figure 2). The design goal of Redundant Reception is to increase the possibility of broadcast receptions on CN-channels. Without Redundant Reception, CA-CBT ensures that each node receives broadcast packets from only one (primary) parent node; however, with Redundant Reception, the current node (say, node X) assigned CN-channels joins an additional (secondary) parent node if it has at least one RI for the secondary parent. This causes these nodes on CN-channels to receive, with high probability, broadcast packets from both the primary and secondary parent nodes. To join with this additional parent node, current node X selects the secondary parent node that provides smallest path cost $C(X)$, as explained in Section 4.2, excepting its own primary parent node.

Figure 6 shows an example of Forced Assignment and Redundant Reception in which all nodes are on the tree. Here, each link is labeled by the channel number. Suppose that the current node is node E and the dotted circle is the interference range of node E. In Figure 6(a), node E receives broadcast packets from its primary parent node B on CN-channel two. To select a secondary parent node, node E considers neighbor nodes A, C, D, F, and H on the tree, as shown in Figure 6(b), excepting primary parent node B.

To select a secondary parent, the algorithm calculates path cost $C(E)$ from node E to the broadcast source via each neighbor node. For simplicity, assume that path cost $C(E)$ is the smallest when node E joins node A. Here, $C(E) = C(A) + 1/R_{\text{bc}}(A)$, where $R_{\text{bc}}(A)$ is the broadcast rate of node A for child node E. Then, node E uses one of its available FIs as an RI and assigns the RI with channel one to join the secondary parent node A. Figure 6(c) shows the result of


Interference range of node E
A
B
C
D
E
F
H
1
2
3
2
▨ Tx NIC (TI)
■ Rx NIC (RI)
□ Free NIC (FI)
Interference range of node E
Interference range of node E

(a) Node E receives broadcast packets on CN-channel 2

(b) Node E can join neighbor nodes using its available FIs

(c) Node E joins one more parent A for Redundant Reception

Figure 6: An example illustrating Forced Assignment and Redundant Reception (Procedures 4 and 5).

this Redundant Reception procedure. In this way, Redundant Reception increases the possibility of broadcast receptions for nodes on CN-channels. Since this procedure iterates over N_B neighbor nodes to select a secondary parent node, its complexity is $O(N_B)$.

4.4. Discussion. In summary, CA-CBT traverses each node of the tree by checking whether current node X has CF-channels or not. If there is no CF-channel for group G_X, CA-CBT performs Channel Reassignment, Link Merge, and Range Expansion in sequence until it obtains CF-channels. If CA-CBT cannot obtain any CF-channels from these three procedures, it then performs Forced Assignment and Redundant Reception to assign CN-channels.

We summarize the features of the above five procedures as follows. First, Channel Reassignment does not degrade the broadcast performance of CA-CBT since it does not affect any broadcast rates of interfering groups around current node X. Second, Link Merge may degrade broadcast performance of all influenced descendant nodes of current node X, because it decreases the broadcast rate of interfering groups around node X; that is, it changes the state of those interfering groups from multi-TX to single-TX. Third, Range Expansion also degrades the broadcast performance of CA-CBT since the parent node of current node X decreases the broadcast rate of its TI to cover all child nodes of node X.

In short, Link Merge and Range Expansion may degrade the broadcast performance of CA-CBT; however, CA-CBT can guarantee collision-free broadcasting in a WSN through Channel Reassignment, Link Merge, and Range Expansion. Fourth, Forced Assignment cannot guarantee collision-free broadcasting, since it assigns CN-channels that include at least one interfering group; however, CA-CBT reduces interference by assigning CN-channels with the smallest number of interfering groups around current node X. Moreover, Redundant Reception compensates for the nodes assigned to CN-channels by allowing them to receive redundant broadcast packets from both primary and secondary parent nodes, thus increasing the possibility of broadcast receptions on CN-channels.

Complexity of CA-CBT. The tree construction process of Algorithm 1 visits N nodes by iterating over their one-hop neighbor nodes. This requires $O(N \cdot N_B)$, where N_B is the average number of neighbor nodes near node X. For current node X, Channel Reassignment (Algorithm 2) and Link Merge (Algorithm 3) iterate over the interfering groups of node X for M TIs and K channels. Thus, both Channel Reassignment and Link Merge together require $O(M \cdot N_I + K \cdot N_I)$, where N_I is the average number of interfering nodes around node X.

Range Expansion (Algorithm 4) is only performed by the parent node of node X for M TIs, K channels, and N_L links between the parent and child nodes of node X, thus requiring $O(M + K + N_L)$. Since Forced Assignment calculates the N_k^I value for K CN-channels by iterating over N_I interfering nodes around node X, it requires $O(K \cdot N_I)$. Redundant Reception iterates over N_B neighbor nodes to select a secondary parent node for node X, which results in $O(N_B)$.

Based on our system model, we assume that the number of NICs per node (M) is smaller than that of available channels (K) in a WSN. Since CA-CBT uses these five procedures for each set of N nodes, we conclude that the overall complexity of CA-CBT is $O(N \cdot N_B + (M+K) \cdot N_I \cdot N) = O(N \cdot N_B + K \cdot N_I \cdot N) = O(K \cdot N^2)$.

5. Performance Evaluation

In this section, we evaluate the broadcast performance of CA-CBT by using ns-2 simulations [24]. For performance comparison, we considered three recently proposed algorithms for broadcasting in WSNs, that is, LMT [13], PAMT [13], and MRDT [17]. These three existing algorithms assign channels via CCA, which is known to be good for broadcasting in WMNs [13, 17]. For each simulation result, we generated 10 different scenarios, each scenario including a single broadcast source node (i.e., the gateway node) and from 40 to 100.

Broadcast destination nodes were randomly positioned in a network field of 1000 m by 1000 m [13–18]. For the physical and MAC layers, we used default parameters for IEEE 802.11a NICs, and each node was equipped with three IEEE 802.11a NICs. Based on IEEE 802.11a, nodes could use eight link data rates from 6 Mbps to 54 Mbps and 12 nonoverlapping channels. Further, radio propagation was modeled using the two-ray ground propagation model, and we configured distance ranges for different link data rates based on the *Cisco Aironet* IEEE 802.11a card data sheet [25].

We set the interference range to the transmission range of the basic data rate (i.e., 6 Mbps). For broadcast traffic, we used a single broadcast packet and a stream of broadcast packets. The payload size of a broadcast packet was set to 1000 bytes. For a stream of broadcast packets, the interpacket delay between two successive packets was set to 20 ms. Table 1 summarizes all of the simulation parameters for our evaluation study.

Table 1: Simulation parameters for performance evaluation.

Parameters	Values
Network size	1000 m by 1000 m
Number of nodes	A single source and from 40 to 100 nodes
Number of NICs	Three NICs per node
PHY & MAC spec.	IEEE 802.11a NIC
Data rates	6, 9, 12, 18, 24, 36, 48, and 54 Mbps
Number of channels	12 nonoverlapping channels
Broadcast traffic	A single broadcast packet or a stream of broadcast packets with the payload size of 1000 bytes
Interpacket delay	20 ms

For a single broadcast packet, we used the following performance metrics: (i) *broadcast latency*, the elapsed time between the time when the broadcast source node transmits a broadcast packet and the time when the broadcast packet is successfully received by all nodes, and (ii) *delivery percentage*, the number of nodes that received a broadcast packet divided by the total number of nodes. For a stream of broadcast packets, we used the additional set of metrics from [13]: (iii) *total broadcast latency*, the elapsed time between the transmission time of the first broadcast packet and the reception time of the last broadcast packet at all nodes, and (iv) *broadcast delivery percentage*, the number of broadcast packets successfully received at all nodes divided by the number of packets to be sent at all nodes. We averaged each performance metric value over 10 different scenarios for each simulation result.

5.1. Performance with a Single Broadcast Packet. In this subsection, we evaluate the performance of CA-CBT with a single broadcast packet. Figure 7(a) shows broadcast latency with the number of nodes varied in a WSN, with results being averaged from 10 different scenarios. In each scenario, the broadcast source node (i.e., the gateway) transmitted one broadcast packet on the tree. Each node was equipped with three NICs and used 12 noninterfering channels. From the figure, when the number of available channels is not small, we observe that CA-CBT showed lower broadcast latency as compared to LMT, PAMT, and MRDT. For instance, when there were 80 nodes in the WSN, the broadcast latency of CA-CBT was approximately 30.26%, 35.19%, and 33.11% lower than LMT, PAMT, and MRDT, respectively.

Figure 7(b) shows broadcast latency with a varying number of channels, with each result averaged over 10 distinct scenarios. In each scenario, the gateway node transmitted one broadcast packet. Further, the number of nodes was 100, and each node was equipped with three NICs. From Figure 7(b), when the number of channels was less than six, we observe that CA-CBT showed higher broadcast latency than LMT, PAMT, and MRDT. This occurred because CA-CBT may decrease the broadcast rate of the groups on the tree by using

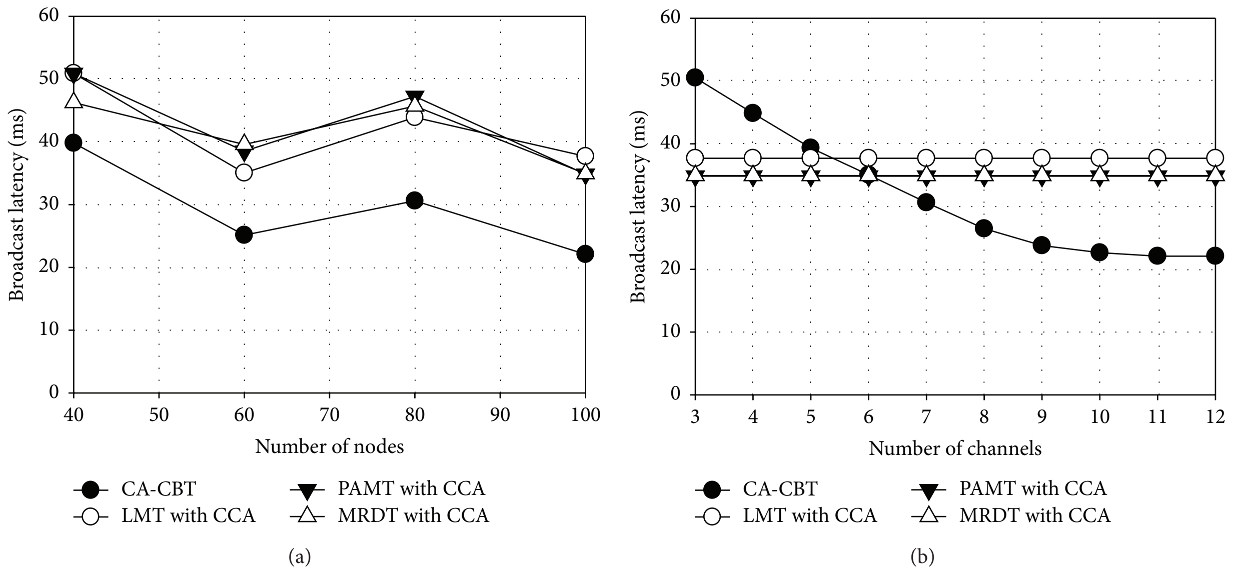


Figure 7: Broadcast latency performance with a single broadcast packet for (a) three NICs, 12 channels and (b) three NICs, 100 nodes.

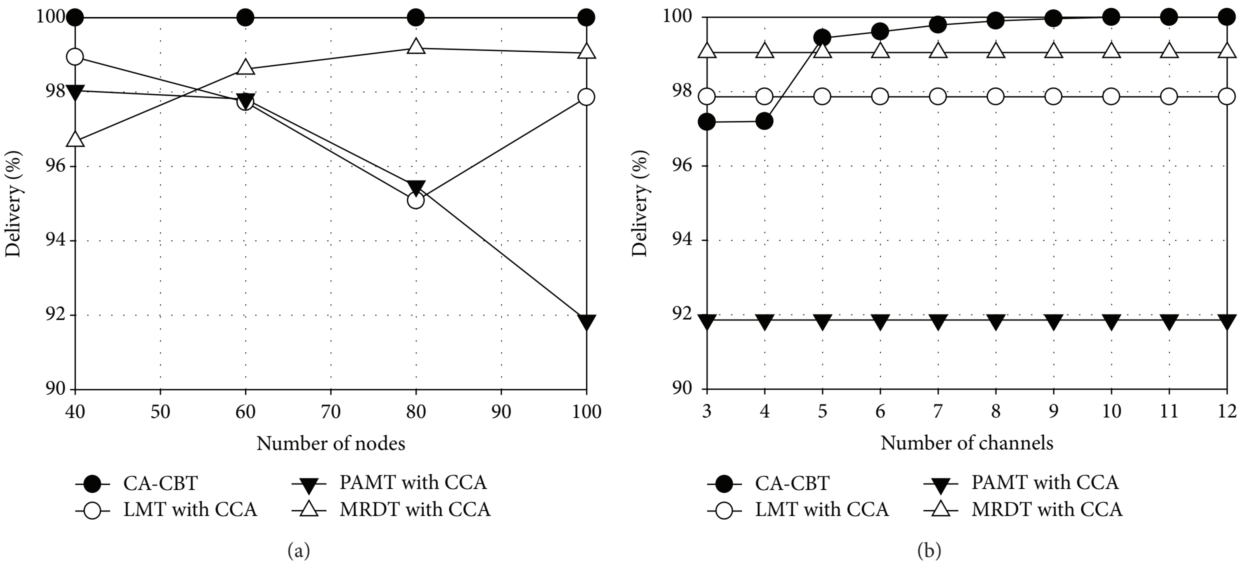


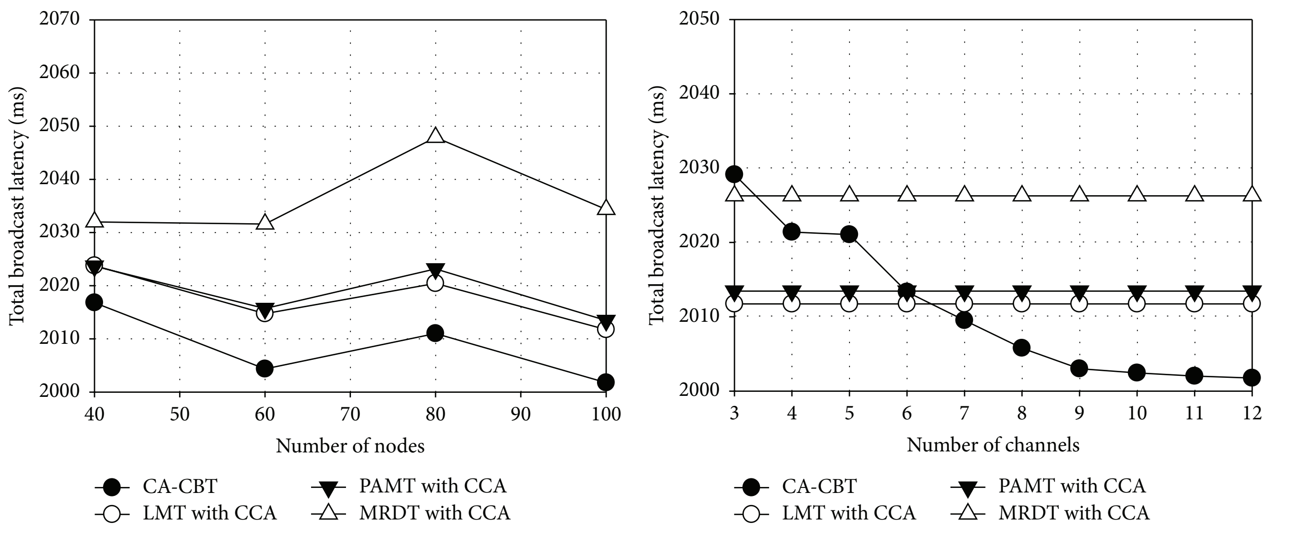
Figure 8: Delivery percentage performance with a single broadcast packet for (a) 3 NICs, 12 channels and (b) 3 NICs, 100 nodes.

Link Merge and Range Expansion when a small number of channels are available; however, when there are more than six channels, we observe that CA-CBT showed lower broadcast latency than the other algorithms.

Based on the results of Figure 7, we make a number of observations. To reduce broadcast latency, the number of concurrent broadcasts, which is determined by the channel diversity of a broadcast tree, should be increased. In LMT, PAMT, and MRDT, channel diversity is restricted to the number of NICs per node since CCA only utilizes three channels through three NICs. We observe this phenomenon in Figure 7(b), where LMT, PAMT, and MRDT showed the same broadcast latency regardless of the number of available channels. Conversely, CA-CBT exploited all 12 of the nonoverlapping channels by using only three NICs, which definitely increased the number of concurrent broadcast transmissions on the tree, thus increasing the channel diversity of CA-CBT's broadcast tree.

Figure 8(a) shows delivery percentage as the number of nodes increases. Results here are averaged over 10 different scenarios. In each scenario, a gateway node transmitted one broadcast packet. Each node used three NICs and 12 noninterfering channels. From the figure, we observe that CA-CBT showed 100% delivery percentage, whereas LMT, PAMT, and MRDT experienced packet loss due to broadcast packet collisions.

Figure 8(b) shows the delivery percentage as the number of available channels was increased from three to 12. As per



(a)

(b)

Figure 9: Total broadcast latency performance with a stream of 100 broadcast packets. (a) 3 NICs, 12 channels. (b) 3 NICs, 100 nodes.

usual, we averaged results over 10 distinct scenarios. Each scenario included 100 nodes and a single broadcast source node that transmitted one broadcast packet. Each of the 100 nodes was equipped with three NICs. As observed in the figure, when the number of channels was small (e.g., less than five), CA-CBT showed lower delivery percentage as compared to LMT and MRDT. The reason here is that CA-CBT may use Forced Assignment for some groups on the tree, which cannot guarantee collision-free broadcasting; however, when the number of channels exceeded four channels, CA-CBT showed a higher delivery percentage than the other algorithms. In particular, when there were more than nine channels, CA-CBT achieved 100% delivery percentage with no packet collisions.

From the results of Figure 8, we explain the improved delivery percentage performance of CA-CBT as follows. In general, as the number of nodes increased, the number of interfering nodes increased on the common channels. Clearly, this also increased the probability of broadcast packet loss, primarily caused by collisions. To reduce packet collisions, LMT, PAMT, and MRDT attempt to cover many nodes with one broadcast transmission positioned in the transmission range of a transmitting node; however, as observed in Figure 8, they still suffer from collisions among broadcast packets since CCA utilizes only three channels via three NICs. On the contrary, CA-CBT fully utilized all available channels to avoid collisions through CF-channels. More specifically, this was achieved by using the Channel Reassignment, Link Merge, and Range Expansion procedures, which decreased the possibility of utilizing CN-channels that may introduce collision-based packet loss.

5.2. Performance with a Stream of Broadcast Packets. Next, we evaluated the performance of CA-CBT by using a stream of broadcast packets. Figure 9 shows the total broadcast latency for a stream of 100 broadcast packets, averaged over 10 different scenarios. The interpacket delay of the broadcast stream was 20 ms. In Figure 9(a), each scenario included a single gateway node and from 40 to 100 nodes. Each node was equipped with three NICs and used 12 noninterfering channels. In Figure 9(b), each scenario consisted of a single broadcast source node and 100 nodes, with each node using three NICs.

From Figure 9(a), we observe that CA-CBT showed a lower total broadcast latency than that of LMT, PAMT, and MRDT, even with a stream of broadcast packets. For example, when there were 80 nodes in the WSN, the total broadcast latency values of CA-CBT, LMT, PAMT, and MRDT were 2011.0 ms, 2020.4 ms, 2023.2 ms, and 2047.9 ms, respectively. In Figure 9(b), when a small number of channels were available (e.g., less than four channels), CA-CBT showed a higher total broadcast latency than that of LMT, PAMT, and MRDT; however, CA-CBT supported lower total broadcast latency than the other algorithms when more than six channels were available.

Figure 10 shows the broadcast delivery percentage for a stream of 100 broadcast packets. Each result here was averaged over 10 different scenarios. We set the interpacket delay to 20 ms. Further, in Figure 10(a), each scenario consisted of a single broadcast source node and a varying number of nodes. The number of NICs per node was three, and there were 12 noninterfering channels. In Figure 10(b), each scenario made up a single gateway node and 100 nodes equipped with three NICs.

From Figure 10(a), we note that CA-CBT achieved 100% broadcast delivery percentage, whereas LMT, PAMT, and MRDT experienced some packet loss due to collisions. In the figure, CA-CBT showed a lower broadcast delivery percentage than that of the other algorithms when the number of channels was small (e.g., three channels); however, as the number of channels increased, CA-CBT achieved almost 100% broadcast delivery by fully utilizing every available channel in the WSN.

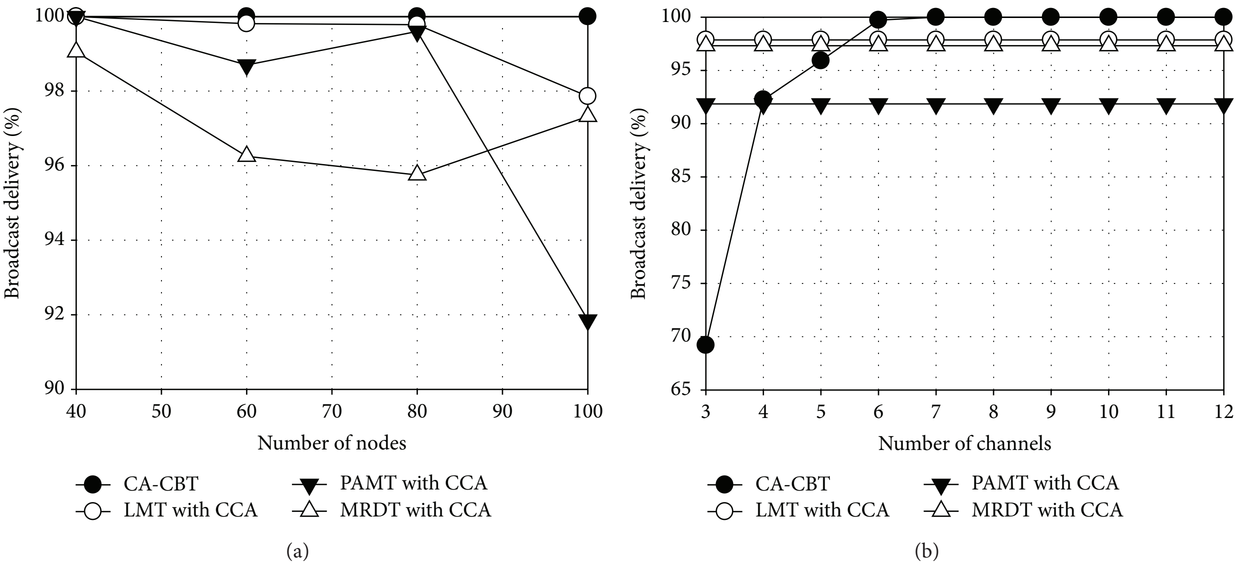


Figure 10: Broadcast delivery percentage performance with a stream of 100 broadcast packets for (a) three NICs, 12 channels and (b) three NICs, 100 nodes.

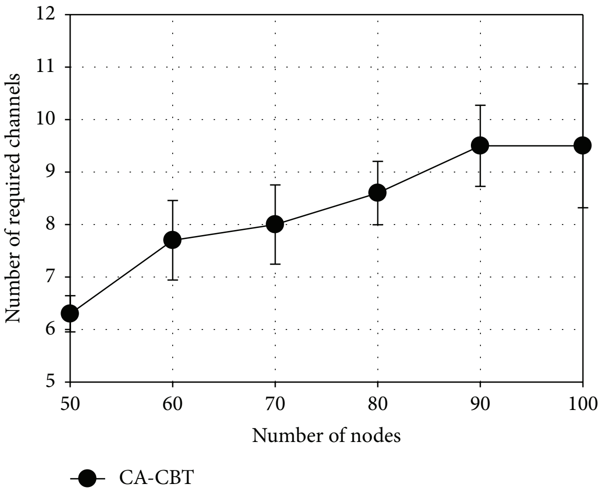


Figure 11: The number of required channels to guarantee collision-free broadcasting via CA-CBT (with three NICs per node).

5.3. The Number of Required Channels for Collision-Free Broadcast. When the number of available channels is unlimited, CA-CBT can guarantee collision-free broadcast by assigning CF-channels to all groups on the broadcast tree; however, IEEE 802.11 standards specify a limited number of channels. Thus, we studied the number of channels required for guaranteeing collision-free broadcasting by CA-CBT as the number of nodes increases in a WSN. Specifically, this result indicates the number of required channels for which we would not perform the Forced Assignment and Redundant Reception procedures of CA-CBT. Figure 11 shows results averaged over 10 different scenarios with a 95% confidence interval. Each scenario included a single broadcast source node and from 50 to 100 broadcast receiver nodes randomly positioned in the network field. Each node used three IEEE 802.11a NICs and 12 noninterfering channels.

In general, the number of interfering groups (or nodes) on the same channels increased as the number of nodes increased. Hence, as expected, we observe in Figure 11 that the number of required channels increased as more nodes were deployed; however, even when there were 100 nodes in the WSN covering a 1000 m by 1000 m field, which can be considered dense mesh router density in WSNs [26], the number of required channels was less than 12 on average. Therefore, if we assume IEEE 802.11a, which supports 12 noninterfering channels, CA-CBT can assign CF-channels to all groups on the tree without performing Forced Assignment and Redundant Reception to assign CN-channels.

5.4. The Number of Groups in the Multi-TX State. As the number of available channels increases, more groups can send broadcast packets in parallel by using multiple TIs. To observe this, we measured the number of groups in the multi-TX state by varying the number of channels. Figure 12 shows results with a 95% confidence interval that were averaged over 10 distinct scenarios. Each scenario consisted of a broadcast source and 50 nodes. The number of NICs per node was three.

From the figure, we observe that the number of multi-TX groups increased as the number of channels increased; that is, because CA-CBT can utilize more CF-channels and does not perform Link Merge by using additional channels, CA-CBT fully utilized additional channels for more multi-TX groups such that it reduced broadcast latency and improved delivery percentage.

5.5. Channel Diversity of CA-CBT and CCA. We investigated the channel diversity of CA-CBT and CCA used by LMT, PAMT, and MRDT. Here, we defined a metric called *channel used ratio* as the total number of channels utilized by each

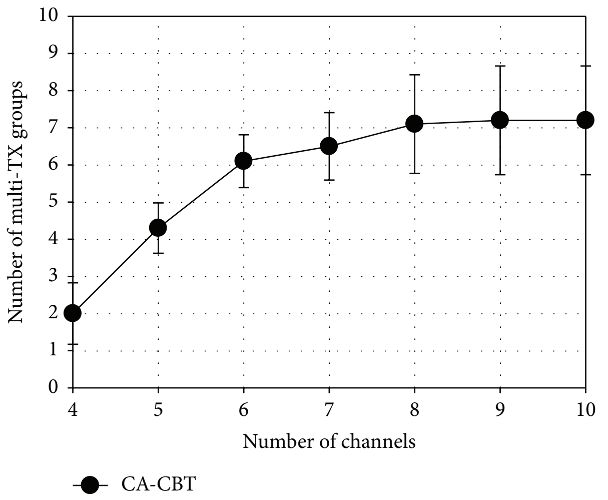


Figure 12: The number of groups in the multi-TX state (50 nodes, three NICs).

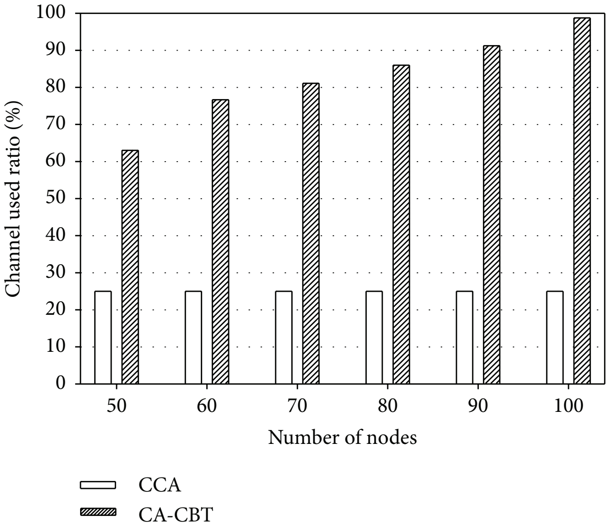


Figure 13: Channel use ratio as the number of nodes increases for three NICs, 12 channels.

algorithm divided by the total number of channels available in the given scenario. As an algorithm utilized more channels, the channel used ratio increased. Figure 13 shows the channel used ratio of CA-CBT and CCA, averaged over 10 different scenarios. In each scenario, there was a single broadcast source node and from 50 to 100 nodes, each node consisting of three NICs and 12 noninterfering channels.

As observed in Figure 13, CCA showed the same channel used ratio value irrespective of the number of nodes. The reason here is that CCA utilized only three channels through three NICs; that is, the channel diversity of CCA was proportional to the number of NICs per node. Conversely, the channel used ratio of CA-CBT increased as the number of nodes increased in the WSN, because CA-CBT utilized all available channels with only three NICs, indicating that the channel diversity of CA-CBT is not restricted to the number of equipped NICs.

6. Conclusion

In this paper, we presented our work on designing a centralized channel assignment algorithm, that is, CA-CBT, for efficient and semireliable broadcasting in WSNs. For efficient and semireliable broadcasting, we focused our efforts on reducing broadcast latency and achieving up to 100% delivery percentage. To realize these goals, our proposed CA-CBT algorithm fully exploited all available channels by using a small number of NICs. First, CA-CBT assigns CF-channels to links on the tree to support collision-free broadcasting. When CA-CBT cannot assign CF-channels, it assigns CN-channels by considering the number of interfering groups (or nodes). Next, the nodes assigned to CN-channels are compensated for by joining both primary and secondary parent nodes to redundantly receive broadcast packets.

We performed extensive ns-2 simulations with a single broadcast packet and a stream of broadcast packets. From our evaluation results, CA-CBT showed lower broadcast latency and higher delivery percentage as compared with existing broadcast algorithms developed for multichannel multiradio multirate WSNs. For our future work, we plan to extend our current CA-CBT algorithm to operate in a distributed way for more practical broadcasting in WSNs.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] R. I. Da Silva, V. Del Duca, A. M. Almeida, and J. M. S. Nogueira, "Wireless sensor network for disaster management," in *Proceedings of the IEEE Network Operations and Management Symposium (NOMS '10)*, pp. 870–873, IEEE, Osaka, Japan, April 2010.

[2] J. Ansari, X. Zhang, and P. Mlähönen, "Multi-radio medium access control protocol for wireless sensor networks," *International Journal of Sensor Networks*, vol. 8, no. 1, pp. 47–61, 2010.

[3] A. M. Canthadai, S. Radhakrishnan, and V. Sarangan, "Multi-radio wireless sensor networks: Energy efficient solutions for radio activation," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '10)*, pp. 1–5, IEEE, Miami, Fla, USA, December 2010.

[4] S. Ji, Y. Li, and X. Jia, "Capacity of dual-radio multi-channel wireless sensor networks for continuous data collection," in *Proceedings of the IEEE INFOCOM*, pp. 1062–1070, IEEE, Shanghai, China, April 2011.

[5] A. Raniwala and T.-C. Chiueh, "Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network," in *Proceedings of the IEEE INFOCOM*, pp. 2223–2234, March 2005.

[6] A. Dhananjay, H. Zhang, J. Li, and L. Subramanian, "Practical, distributed channel assignment and routing in dual-radio mesh networks," in *Proceedings of the ACM SIGCOMM Conference on Data Communication (SIGCOMM '09)*, pp. 99–110, August 2009.

[7] S. Avallone, I. F. Akyildiz, and G. Ventre, "A channel and rate assignment algorithm and a layer-2.5 forwarding paradigm for

multi-radio wireless mesh networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 1, pp. 267–280, 2009.

[8] R. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot, "Interference-aware channel assignment in multi-radio wireless mesh networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, pp. 1–12, Barcelona, Spain, April 2006.

[9] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, 2004.

[10] A. Raniwala, K. Gopalan, and T. Chiueh, "Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 2, pp. 50–65, 2004.

[11] J. Tang, G. Xue, and W. Zhang, "Interference-aware topology control and QoS routing in multi-channel wireless mesh networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC '05)*, pp. 68–77, ACM, May 2005.

[12] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-efficient broadcast and multicast trees in wireless networks," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 481–492, 2002.

[13] J. Qadir, C. T. Chou, A. Misra, and J. G. Lim, "Minimum latency broadcasting in multiradio, multichannel, multirate wireless meshes," *IEEE Transactions on Mobile Computing*, vol. 8, no. 11, pp. 1510–1523, 2009.

[14] C. T. Chou, A. Misra, and J. Qadir, "Low-latency broadcast in multirate wireless mesh networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 11, pp. 2081–2091, 2006.

[15] C. T. Chou and A. Misra, "Low latency multimedia broadcast in multi-rate wireless meshes," in *Proceedings of the 1st IEEE Workshop on Wireless Mesh Networks (WiMesh '05), in Conjunction with IEEE Sensor and Ad Hoc Communications and Networks (SECON '05)*, 2005.

[16] L. Li, B. Qin, and C. Zhang, "Interference-aware broadcast routing and channel assignment in multi-radio wireless mesh networks," in *Proceedings of the 70th IEEE Vehicular Technology Conference Fall (VTC '09)*, pp. 1–5, IEEE, Anchorage, Alaska, USA, September 2009.

[17] J. Qadir, C. T. Chou, A. Misra, and J. G. Lim, "Localized minimum-latency broadcasting in multi-radio multi-rate wireless mesh networks," in *Proceedings of the 9th IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM '08)*, pp. 1–12, IEEE, Newport Beach, Calif, USA, June 2008.

[18] J. Qadir, C. T. Chou, A. Misra, and J. G. Lim, "Localized minimum-latency broadcasting in multi-radio multi-rate wireless mesh networks," in *Proceedings of the International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM '08)*, pp. 1–12, Newport Beach, Calif, USA, June 2008.

[19] C. T. Chou, J. Qadir, and J. G. Lim, "Advances and challenges with data broadcasting in wireless mesh networks," *IEEE Communications Magazine*, vol. 45, no. 11, pp. 78–85, 2007.

[20] R. Gandhi, S. Parthasarathy, and A. Mishra, "Minimizing broadcast latency and redundancy in ad hoc networks," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, ACM, San Diego, Calif, USA, September 2003.

[21] Z. Chen, C. Qiao, J. Xu, and T. Lee, "A constant approximation algorithm for interference aware broadcast in wireless networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 740–748, IEEE, Anchorage, Alaska, USA, May 2007.

[22] R. Mahjourian, F. Chen, and R. Tiwari, "An approximation algorithm for conflict-aware broadcast scheduling in wireless ad hoc networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, May 2008.

[23] C. E. Perkins, E. M. Belding-Royer, and I. D. Chakeres, "Ad Hoc On Demand Distance Vector (AODV) Routing," IETF Internet draft, 2003.

[24] NS Official Website, http://www.isi.edu/nsnam/ns/.

[25] Cisco Aironet 1240 AG Series 802.11 a/b/g, http://www.cisco.com/.

[26] J. Robinson and E. W. Knightly, "A performance study of deployment factors in wireless mesh networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (IEEE INFOCOM '07)*, pp. 2054–2062, May 2007.

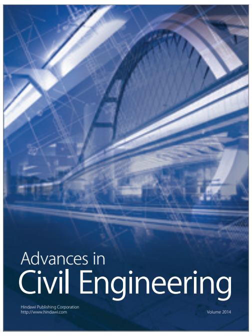
Journal of
Engineering

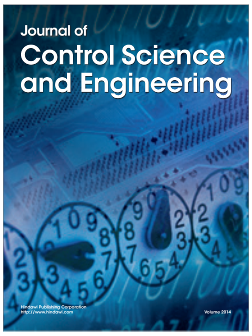
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

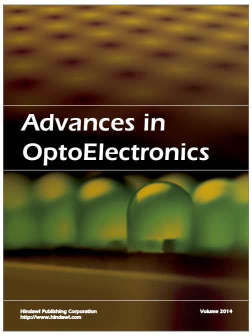
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

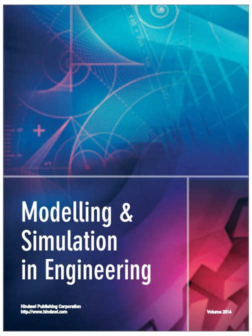
Journal of
Robotics

Journal of
Electrical and Computer Engineering

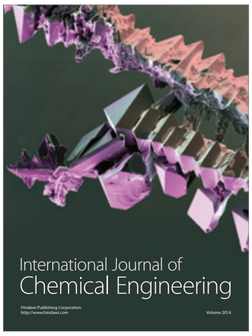

Advances in
OptoElectronics

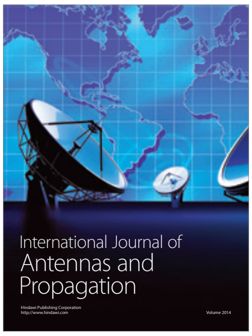
VLSI Design

International Journal of
Navigation and Observation

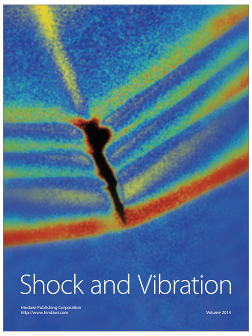

Modelling & Simulation in Engineering

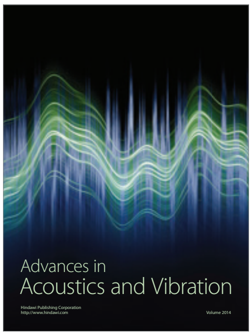
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration