Interference-Aware Broadcasting in Multi-Radio Multi-Channel Mesh Networks

Min Song, *Senior Member, IEEE,* Jun Wang, *Student Member, IEEE,* Kai Xing, *Student Member, IEEE,* and E. K. Park, *Member, IEEE*

***Abstract*—A vast number of broadcasting protocols have been developed for wireless networks. To the best of our knowledge, however, most of these protocols assume a single-radio single-channel network model and/or a generalized physical model, which does not take into account the impact of interference. In this paper, we present a Distributed Interference-aware Broadcasting (DIB) protocol for multi-radio multi-channel mesh networks. The protocol has two phases. In the first phase, each node constructs a local structure by removing bad links and channels. In the second phase, a high-performance broadcasting tree is built by using message passing procedures. Our research distinguishes itself in a number of ways. First, a multi-radio multi-channel mesh network model is used. Second, comprehensive link and channel quality metrics are defined to fully take into account interferences. Third, four design principles have been identified in the tree building process to combat inter-node and intra-node interferences. Finally, a comprehensive performance metric, called power, is defined which includes reliability, receiving redundancy, latency, and goodput. Analytical and simulation studies verify that the DIB protocol is able to achieve 100% reliability, less broadcasting redundancy, low broadcasting latency, and high goodput.**

***Index Terms*—Broadcasting, protocols, mesh networks, multiple radios, multiple channels.**

I. INTRODUCTION

MESH networks are viewed as a promising broadband access infrastructure in both urban and rural environments. In mesh networks there are two types of nodes: mesh routers and mesh clients [1]. A small set of routers also function as gateways connecting to the wired network. Typical deployments of mesh networks utilize mesh routers equipped with only one IEEE 802.11 radio. Research has indicated that single-radio single-channel mesh networks suffer from serious capacity degradation [2]. A promising approach to improve the capacity of mesh networks is to provide each node with multiple-radio multi-channel capabilities and permit MAC protocols to adjust the transmission rate [3].

Broadcasting in wireless networks is fundamentally different to the way in which wired networks function due to the well-known *Wireless Broadcast Advantage* [4]. A vast number of broadcasting protocols have been developed for wireless ad hoc networks with different focuses. In [5]–[9], the focus is to ensure 100% reliability, i.e., every node in the network is guaranteed to receive the broadcast message. In [10]–[13], the focus is to achieve a minimum broadcast latency, i.e., the time the last node in the network receives the broadcast message is minimized. In [11], [14]–[17], the focus is to alleviate the *Broadcast Storm Problem* [18] by reducing the redundant transmissions. Unfortunately, all of the aforementioned protocols assume a single-radio single-channel model and/or a generalized physical model, which does not take into account the impact of interferences.

Our contributions are as follows. First, we developed a Distributed Interference-aware Broadcasting (DIB) protocol to build a high-performance broadcasting tree to achieve 100% reliability, low broadcasting latency, less broadcasting redundancy, and high goodput. To combat inter-node and intra-node interferences, four design principles have been identified in guiding the tree construction. Second, a multi-radio multi-channel mesh network model is used. The presence of multi-radio allows a mesh node to send and receive at the same time; the availability of multi-channel allows channels to be reused across the network, which expands the available spectrum and reduces interference. Third, comprehensive link and channel quality metrics are defined to fully take into account interferences. The link and channel quality information are also made available to the DIB protocol. Fourth, to increase the scalability of the DIB protocol, only 1-hop or 2-hop local information is used in constructing the broadcasting tree. Finally, to facilitate the performance analysis, a comprehensive performance metric, called power, is defined. The power includes reliability, receiving redundancy, latency, and goodput.

The rest of the paper is organized as follows. Section II introduces the network model and problem formulation. Section III briefly surveys the related work. The new link and channel quality metrics and the DIB protocol are presented in Section IV. Section V provides the simulation results and analysis. The conclusions and future work are given in Section VI.

II. NETWORK MODEL AND PROBLEM STATEMENT

Computer networks are typically modeled by an undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is the set of vertices representing nodes and $\mathbf{E}$ is the set of edges representing the communication links. This model, however, may not represent the multi-radio multi-channel mesh networks in which multiple links

Manuscript received December 20, 2007; revised May 3, 2008; accepted August 6, 2008. The associate editor coordinating the review of this paper and approving it for publication was X. Zhang.

M. Song and J. Wang are with the Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA 23529 USA (e-mail: {msong, jwang012}@odu.edu).

K. Xing is with the Computer Science Department, George Washington University, Washington, DC 20052 USA (e-mail: gwukai@gmail.com).

E. K. Park is with the Computer Science and Electrical Engineering Department, University of Missouri at Kansas City, Kansas City, MO 64110 USA (e-mail: ekpark@umkc.edu).

Digital Object Identifier 10.1109/T-WC.2008.071433

may exist between two nodes and one link may connect to multiple nodes. As a result, the link quality is unidirectional. In this paper, we use a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E}_c)$ to model the multi-radio multi-channel mesh networks. Here $\mathbf{E}_c$ is the set of colored edges representing the directed links. We assume the multi-radio multi-channel mesh network is strongly connected, i.e., $\mathbf{E}_c$ is a strongly connected. A directed link (i, j, c), which corresponds to the link from node i to node j with channel c, is in set $\mathbf{E}_c$ if and only if the following two conditions hold,

- The Euclidean distance between nodes i and j is no greater than the communication range
- Node i is tuned to channel c for transmission and node j is tuned to c for receiving.

Two types of interference are considered. They are the inter-node interference, which occurs when adjacent nodes are using the same channel, and the intra-node interference, which happens when multiple channels are used by the same node. In multi-radio multi-channel mesh networks, the impact of these interferences dramatically increases without a proper channel assignment policy.

Given the network model defined above, the problem is to develop a broadcasting protocol to ensure that all nodes in the network quickly receive the broadcasting messages. This problem can be addressed by constructing a broadcasting tree, $\mathbf{T} = (\mathbf{N}^B, \mathbf{E}^B)$, where $\mathbf{N}^B \subset \mathbf{V}$ and $\mathbf{E}^B \subset \mathbf{E}_c$ represent the set of nodes and the set of links that participate in the broadcasting, respectively. Given the fact that the problem of minimum latency broadcasting in wireless networks is NP-hard, the objective of this paper is to construct a quasi-optimal tree to achieve 100% reliability, less broadcasting redundancy, low broadcasting latency, and high goodput. Not surprisingly, these performance metrics are often contradictory. Fig. 1 shows an 18-node mesh network, in which only the numbered nodes participating in broadcasting (node 1 is the source) and the unfilled nodes receive at least one redundant message. For clarity purpose, we assume each node has only one channel. If the primary goal is efficiency, the broadcasting protocol would result in 4 transmissions and 8 receiving redundancies (Fig. 1a). The price paid, however, is 94% reliability (one node is not covered). If the primary goal is reliability, the broadcasting protocol would result in 5 transmissions and 11 receiving redundancies (Fig. 1b). Certainly, more redundancies bring more interference and thus increase the latency. So, one of the design challenges of broadcasting protocols is to find a solution that has a favorable tradeoff.

III. Related Work

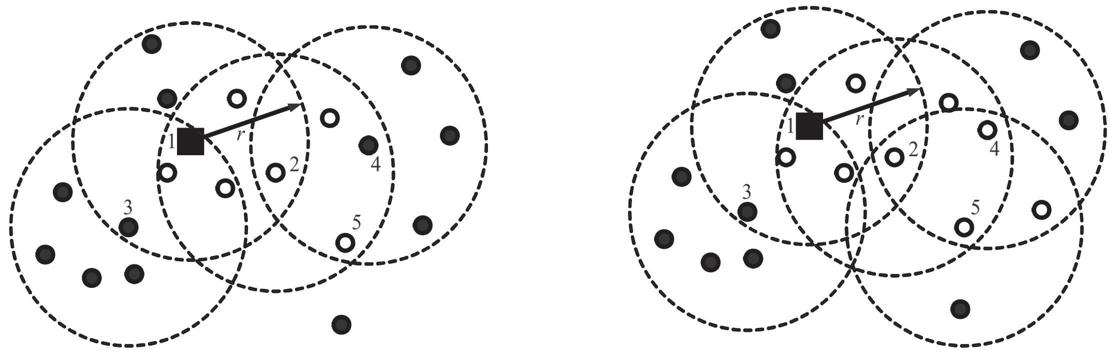
A. Channel Quality Assessment

Very often interference-aware routing protocols [3], [19] and interference-aware MAC layer protocols [19]–[21] assume that either a *priori* information about the interference is known, or a 0-1 function is applied to the link, i.e., a link either works (1) or does not work (0). Few studies have contributed to defining the measurement of interference. The first exceptional study was made in [22] to estimate the link interference in a static single-radio single-channel experimental wireless network. The way this study calculates the interference, however, is not practical in real-world mesh networks. Therefore, finding a practical wireless interference-aware metric is critical. Another notable study is presented in [23], in which a metric termed as *Expected Transmission Count* (*ETX*) is defined to find a high throughput path. The *ETX* of a link is calculated using the forward and reverse delivery rates of the link. The *ETX* of a path is then the sum of the *ETX* for each link in the path. Although *ETX* does very well in homogeneous single-radio environments, it does not perform well in environments with multiple radios as indicated in [3]. We further argue that *ETX* does not accurately represent the quality of the entire path in the context of broadcasting. Recall that there is no acknowledgment in broadcasting. Thus only the forward delivery rate of the link should be considered.

B. Broadcasting in Wireless Networks

Two widely used broadcasting methods are the probabilistic and tree-based approaches. In the probabilistic broadcasting approach (also called gossip-based approach) [11], [17], [24]–[26], when a node first receives a broadcasting message it broadcasts the message to its neighbors with a probability of p and discards the message with a probability of $1 - p$. Factors, including the node degree and network degree, may contribute to the determination of gossiping probability. Effectively, the nodes participating in the broadcasting build a tree. The probabilistic approach demonstrates several desirable features, such as scalability and fault-tolerance. The challenges for this approach are how to find the appropriate gossiping parameters and how to guarantee 100% reliability. In the tree-based approach [10]–[12], [15], [27], [28], a broadcasting tree is constructed first before the broadcasting messages are actually transmitted. By using local topological information or the entire network topological information, a sub-optimal tree can be constructed to reduce redundant transmissions. The tree-based method can achieve a deterministic performance. However, a nontrivial overhead is involved to construct the tree regardless of whether the tree is constructed in a centralized or a distributed way.

As we mentioned earlier, most of the broadcasting protocols have been developed primarily with one focus: reliability, broadcast latency, or redundant transmissions. These performance metrics are often contradictory goals. In an effort to minimize latency and the number of retransmissions, a broadcast schedule is developed for collision free broadcasting [11]. While the results are promising, the assumption of a single-radio single-channel and single-rate model limits its usage in multi-radio multi-channel networks. One notable work has been recently presented in [29], in which a set of algorithms are designed to achieve low broadcasting latency in multi-radio multi-channel and multi-rate mesh networks. The broadcasting tree is constructed using a set of centralized algorithms with a goal of minimizing broadcasting latency. However, the centralized approach results in a nontrivial overhead to construct and maintain the tree. In addition, these algorithms are evaluated in a 10-node mesh network, thus making it less clear about the scalability of the proposed algorithms.



(a) (b)

Fig. 1. Illustration of the performance tradeoff.

IV. New Metrics and the DIB Protocol

A. Notation

$\mathbf{N}(i)$	Set of nodes within the communication range of node i
$\mathbf{N}_c(i)$	Set of nodes that are tuned to channel c for receiving, $\mathbf{N}_c(i) \subset \mathbf{N}(i)$
$\mathbf{E}(i)$	Set of links connected to node i
$\mathbf{C}(i)$	Set of channels node i has
$\mathbf{Children}_i$	Set of nodes that receive the broadcasting messages from node i, initially empty
Father_i	The node that transmits the broadcasting messages to node i, initially empty
$i \xrightarrow{c} j$	The transmission link from node i to node j with channel c

B. New Link and Channel Metrics

In this paper, we will use a single comprehensive parameter to quantify the quality of each link. For the link from node i to node j with channel c, we define the link metric as

$$w_{ij,c} = R_c \times DR_{ij,c}, j \in \mathbf{N}_c(i) \tag{1}$$

where R_c is the transmission rate of channel c, and $DR_{ij,c}$ is the packet delivery rate from node i to node j with channel c. The packet delivery rate can be approximated using the techniques described in [3], [23].

To measure the quality of a channel, the qualities of all links that use the channel must be taken into account. Additionally, to increase the channel usage, a channel that has been tuned for receiving by a large number of neighbors should be granted a higher weight. Thus, we define the channel metric as

$$w_{i,c} = R_c \frac{\sum\limits_{j \in \mathbf{N}_c(i)} DR_{ij,c}}{|\mathbf{N}_c(i)|} \frac{|\mathbf{N}_c(i)|}{|\mathbf{N}(i)|} = R_c \frac{\sum\limits_{j \in \mathbf{N}_c(i)} DR_{ij,c}}{|\mathbf{N}(i)|} \tag{2}$$

Note that only the good links and channels that have a weight greater than or equal to the link threshold, noted as $\overline{w_l}$, and channel threshold, noted as $\overline{w_c}$, are eligible to participate in broadcasting.

C. DIB Protocol

To combat inter-node and intra-node interferences, the following principles are used in building the broadcasting tree:

1) A node should avoid using the same channel for both transmitting and receiving;
2) A node should avoid using the same channel for both transmitting and receiving;
3) When a node chooses a channel for transmission, a channel with a higher weight from its own perspective and a lower weight from its children's perspective is preferred;
4) Adjacent nodes should avoid using the same channel for transmission.

It should be noticed that not all of these principles can be followed in some extreme cases. For instance, principles 2 and 4 can not be applied if there are not enough channel resources. For this reason, a MAC-layer scheduler is assumed to avoid channel conflict. For principle 2, if one node has to broadcast and it has only one available transmission channel which is the same as its receiving channel, the receiving and transmission must be scheduled to avoid intra-node interference. For principle 4, if two adjacent broadcasting nodes, i and j, choose the same transmission channel, c, the broadcasting of node i and j must be scheduled to avoid inter-node interference. Next, we describe the DIB protocol. The protocol has two phases. In the first phase, each node builds a local structure by removing *bad* channels and links. In the second phase, a high-performance broadcasting tree is built by using message passing procedures. We assume all nodes initially share a common channel for exchanging all the control messages.

1) Phase 1: Construct local structures: In phase 1, node i uses its local information $\langle \mathbf{N}(i), \mathbf{E}(i), \mathbf{C}(i) \rangle$ to construct a local structure $\langle \{\mathbf{N}_i^T, \mathbf{N}_i^R\}, \{\mathbf{E}_i^T, \mathbf{E}_i^R\}, \{\mathbf{C}_i^T, \mathbf{C}_i^R\} \rangle$ as follows:

- The good channels for transmission are the subset $\mathbf{C}_i^T = \{c | w_{i,c} \geq \overline{w_c}, c \in \mathbf{C}(i)\}$; the good channels for receiving are the subset $\mathbf{C}_i^R = \{c | w_{ni,c} \geq \overline{w_l}, n \in \mathbf{N}(i), c \in \mathbf{C}(i)\}$.
- The good links for transmission are the subset $\mathbf{E}_i^T = \{i \xrightarrow{c} j | w_{ij,c} \geq \overline{w_l}, j \in \mathbf{N}(i), c \in \mathbf{C}_i^T\}$; the good links for receiving are the subset $\mathbf{E}_i^R = \{n \xrightarrow{c} i | w_{ni,c} \geq \overline{w_l}, n \in \mathbf{N}(i), c \in \mathbf{C}_i^R\}$.
- The outgoing neighbors of node i (neighbors that are going to receive the broadcasting messages from node i)

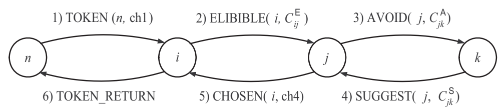


Fig. 2. Illustration of message passing procedures.

are the subset $\mathbf{N}_i^T = \{j | i \xrightarrow{c} j \in \mathbf{E}_i^T, j \in \mathbf{N}(i), c \in \mathbf{C}_i^T\}$; and the incoming neighbors of node i are the subset $\mathbf{N}_i^R = \{n | n \xrightarrow{c} i \in \mathbf{E}_i^R, n \in \mathbf{N}(i), c \in \mathbf{C}_i^R\}$.

In summary, phase 1 removes all bad channels and links whose weights are below the thresholds. Once the local structure is built, node i can easily figure out the good transmission channels and links from node i to node j, which are $\mathbf{C}_{ij} = \{c | i \xrightarrow{c} j \in \mathbf{E}_i^T, c \in \mathbf{C}_i^T\}$ and $\mathbf{E}_{ij} = \{i \xrightarrow{c} j | c \in \mathbf{C}_{ij}\}$, respectively.

2) Phase 2: Build the broadcasting tree using message passing procedures: We first use Fig. 2 as an example to illustrate the main idea of phase 2. Assume node n has already chosen ch1 for broadcasting, $\mathbf{C}_{ij} = \{\mathrm{ch4}, \mathrm{ch3}, \mathrm{ch1}\}$, $\mathbf{C}_{jk} = \{\mathrm{ch3}, \mathrm{ch4}\}$, and $\mathbf{C}_k^T = \{\mathrm{ch1}, \mathrm{ch4}\}$. Notice that the order of channels indicates the quality from high to low. Assuming node i needs to participate in broadcasting, it needs to decide which channel should be used.

Initially, node n generates a TOKEN message that contains its ID and broadcasting channel (ch1). Once node i receives the TOKEN message, it sends out an ELIGIBLE message to node j containing a list of eligible channels that node i may use for broadcasting, $\mathbf{C}_{ij}^E = \mathbf{C}_{ij} - \{\mathrm{ch1}\} = \{\mathrm{ch4}, \mathrm{ch3}\}$. Observing that $\mathbf{C}_{jk}$ and $\mathbf{C}_{ij}^E$ consist of two common channels, node j sends out an AVOID message to node k. The AVOID message includes a set of channels, $\mathbf{C}_{jk}^A = \mathbf{C}_{ij}^E = \{\mathrm{ch4}, \mathrm{ch3}\}$, that may cause interference should they be chosen by node k as its receiving channels. Notice that $\mathbf{C}_{jk}^A$ can also be interpreted as the potential channels for node i as its transmission channels. Node k responds to node j by generating a SUGGEST message including a set of channels, $\mathbf{C}_{jk}^S$, that node j should avoid using for transmission and thus node i may use for transmission. In this example, $\mathbf{C}_k^T$ has no impact on node i since it has one channel (ch1) which is not included in $\mathbf{C}_{jk}^A$. Therefore, $\mathbf{C}_{jk}^S = \mathbf{C}_{jk}^A = \{\mathrm{ch4}, \mathrm{ch3}\}$. Node j chooses the best channel (ch4) from $\mathbf{C}_{jk}^S$ that should be used as its receiving channel and then responds to node i with a CHOSEN message. The CHOSEN message includes the particular channel (ch4) that will be used by node i for broadcasting.

After choosing its broadcasting channel, node i generates a TOKEN message to node j, and the above procedures are repeated until node j selects its broadcasting channel. Node j then sends the TOKEN_RETURN message to node i, and node i finally passes the TOKEN_RETURN message to node n. This concludes the entire process.

In this example, node i has to use 2-hop information to decide its broadcasting channel. In other cases, 1-hop information is enough. For example, if $\mathbf{C}_{jk} = \{\mathrm{ch3}, \mathrm{ch2}\}$, node i can immediately identify ch4 as its transmission channel without issuing an ELIGIBLE message. We now proceed to present the main procedures in phase 2.

- TOKEN procedure

When receiving a TOKEN message from node n, node i decides whether or not to participate in broadcasting and chooses its transmission channel if it participates.

On arrival of TOKEN(n, ch_{ni}) at node i, do the following,
  // ch_{ni} is the chosen broadcasting channel from n to i
  for all j such that $j \in \mathbf{N}_i^T - \{n\}$ **do**
    $\mathbf{C}_{ij}^E = \mathbf{C}_{ij} - \{\mathrm{ch}_{ni}\}$ // $\mathbf{C}_{ij}^E$ is the set of eligible channels that i may use for broadcasting
    $\forall c \in \mathbf{C}_{ij}^E$, sort $\mathbf{C}_{ij}^E$ by descent order of $w_{i,c} - w_{j,c}$
    Send ELIGIBLE$(i, \mathbf{C}_{ij}^E)$ to node j
    Wait CHOSEN(i, ch_{ij}) from node j
      // ch_{ij} is the chosen transmission channel of i
    if $\mathrm{ch}_{ij} \neq$ NULL **then**
      Add j to $\mathbf{Children}_i$ with channel ch_{ij}
    end if
    Remove links $\{i \xrightarrow{c} j | c \in \mathbf{C}_{ij}, c \neq \mathrm{ch}_{ij}\}$ from $\mathbf{E}_i^T$ and $\mathbf{E}_{ij}$ // Lemma 2 refers to this as RO1
  end for
  for all m such that $m \in \mathbf{N}_i^R - \{n\}$ **do**
    Send NOTIFY$(i, \bigcup_{j \in \mathbf{N}_i^T - \{n\}} \{\mathrm{ch}_{ij}\})$ to node m
  end for
  for all j such that $j \in \mathbf{Children}_i$ **do**
    Send TOKEN(i, ch_{ij}) to node j
    Wait TOKEN_RETURN from node j
  end for
  Send TOKEN_RETURN to node n
End

- ELIGIBLE procedure

When receiving an ELIGIBLE message from node i, node j makes a decision to either accept node i as its father (and thus has a broadcast link from node i) or reject nodes i as its father.

On arrival of ELIGIBLE (i, $\mathbf{C}_{ij}^E$) at node j, do the following,
  if $\mathrm{Father}_j \neq$ NULL **then**
    $\mathrm{ch}_{ij} =$ NULL // i does not need to transmit to j
  else if $\mathbf{C}_{ij}^E = \emptyset$ **then**
      // the only good transmission channel from i to j is same as i's receiving channel
    if $|\mathbf{E}_j^R| = 1$ **then**
      $\mathrm{ch}_{ij} = c$, s.t. $n \xrightarrow{c} j \in \mathbf{E}_j^R$
        // j chooses i as its father with channel ch_{ij}
    else
      $\mathrm{ch}_{ij} =$ NULL // j receives from other neighbors
    end if
  else
    for all k such that $k \in \mathbf{N}_j^T - \{i\}$ **do**
      if $|\mathbf{C}_{jk}| = 1$ **then**
        if $\mathbf{C}_{ij}^E = \mathbf{C}_{jk}$ **then**
          $\mathbf{C}_{jk}^S = \mathbf{C}_{ij}^E$ // $\mathbf{C}_{jk}^S$ includes the channel that j will avoid using for transmission
        else
          $\mathbf{C}_{jk}^S = \mathbf{C}_{ij}^E - \mathbf{C}_{jk}$
        end if
      else if $|\mathbf{C}_{jk}| = 2$ **then**
        if $\mathbf{C}_{ij}^E = \mathbf{C}_{jk}$ **then**

Send AVOID$(j, \mathbf{C}_{ij}^{E})$ to node k
Wait SUGGEST$(j, \mathbf{C}_{jk}^{S})$ from node k
else if $\mathbf{C}_{jk} \cap \mathbf{C}_{ij}^{E} \neq \emptyset$, $\mathbf{C}_{jk} \cap \mathbf{C}_{ij}^{E} \neq \mathbf{C}_{jk}$, $\mathbf{C}_{jk} \cap \mathbf{C}_{ij}^{E} \neq \mathbf{C}_{ij}^{E}$ **then**
$\mathbf{C}_{jk}^{S} = \mathbf{C}_{ij}^{E} - \mathbf{C}_{jk}$
else
$\mathbf{C}_{jk}^{S} = \mathbf{C}_{ij}^{E}$ // k has no impact on the decision
end if
else
$\mathbf{C}_{jk}^{S} = \mathbf{C}_{ij}^{E}$
end if
end for
if $\bigcap_{k \in \mathbf{N}_j^T - \{i\}} \mathbf{C}_{jk}^{S} \neq \emptyset$ **then**
Choose ch_{ij} from $\bigcap_{k \in \mathbf{N}_j^T - \{i\}} \mathbf{C}_{jk}^{S}$ with highest weight
else
Choose ch_{ij} from $\bigcup_{k \in \mathbf{N}_j^T - \{i\}} \mathbf{C}_{jk}^{S}$ with maximal counts
end if
end if
Remove links $\{i \xrightarrow{c} j | c \in \mathbf{C}_j^R, c \neq \mathrm{ch}_{ij}\}$ from $\mathbf{E}_i^R$
// Lemma 2 refer to this removing as RO2
for all m such that $m \in \mathbf{N}_j^R - \{i\}$ **do**
Send NOTIFY$(j, \{\mathrm{ch}_{ij}\})$ to node m
end for
if $\mathrm{ch}_{ij} \neq$ NULL **then**
$Father_j = i$ // j chooses i as its father
end if
Send CHOSEN(i, ch_{ij}) to node i
End

- AVOID procedure

When receiving an AVOID message from node j, node k uses its own transmission channel(s) information to help node j choose its receiving channel.

On arrival of AVOID (j, $\mathbf{C}_{jk}^{A}$) at node k, do the following,
if $\mathrm{Father}_k \neq$ NULL **then**
// k already has a father. Note that j can't be k's father.
Remove links $\{j \xrightarrow{c} k | c \in \mathbf{C}_k^R\}$ from $\mathbf{E}_k^R$
// Lemma 2 refer to this removing as RO3
Send SUGGEST(j, $\mathbf{C}_{jk}^{A}$) to node j
else if $|\mathbf{C}_k^T| = 1$, and $\mathbf{C}_k^T \subset \mathbf{C}_{jk}^{A}$ **then**
Send SUGGEST(j, $\mathbf{C}_k^T$) to node j
else
Send SUGGEST(j, $\mathbf{C}_{jk}^{A}$) to node j
end if
End

- NOTIFY procedure

Once node i chooses its broadcasting channels, it sends out a NOTIFY message to its neighbors to let them lower the priority of the chosen channels in their transmission channels sets. The NOTIFY message also effectively lessens the hidden terminal problem and exposed terminal problem.

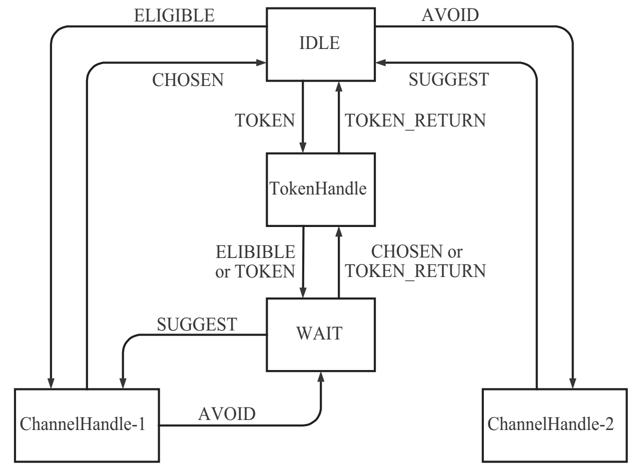


Fig. 3. Finite state machine of the message passing procedures.

On arrival of NOTIFY (i, $\mathbf{C}_{ij}^{N}$) at node j, do the following,
for all c such that $c \in \mathbf{C}_{ij}^{N}$ **do**
Lower the priority of channel c in $\mathbf{C}_j^T$
end for
End

To summarize phase 2, node i uses its local structure and the ones from its neighbors to build a local broadcasting branch, $\mathbf{B}_i = \{j, i \xrightarrow{\mathrm{ch}_{ij}} j | \mathrm{ch}_{ij} \neq \text{NULL}, j \in \mathbf{N}_i^T, \mathrm{ch}_{ij} \in \mathbf{C}_i^T\}$. Eventually, a broadcasting tree is constructed, $\mathbf{T} = \bigcup_{\forall i \in \mathbf{N}} \mathbf{B}_i$. As can be seen, our protocols have good scalability since at maximum 2-hop information is needed.

D. Finite state machine of DIB phase 2

Fig. 3 shows the finite state machine for the general case of phase 2. Each node is in one of five states, as follows:

- IDLE: Either no message is received or messages have been handled.
- TokenHandle: Upon receiving a TOKEN message, a node sends an ELIGIBLE message to each of its outgoing neighbors telling them the eligible channels and then turns into the WAIT state. After receiving all responded CHOSEN messages, the node keeps all the transmission channels and removes the other channels. After that, the node sends a TOKEN message to each of its children and moves into the WAIT state waiting for TOKEN_RETURN. Finally, after receiving all responded TOKEN_RETURN messages from its children, the node sends back a TOKEN_RETURN message to its father and moves back to the IDLE state.
- ChannelHandle-1: Upon receiving an ELIGIBLE message, a node sends out an AVOID message to its outgoing neighbors and then moves into the WAIT state. After receiving all responded SUGGEST messages from these neighbors, the node chooses one channel as its receiving channel and removes other unnecessary links to its neighbors. Finally, the node sends back a CHOSEN message to its father and moves to the IDLE state.

- ChannelHandle-2: Upon receiving an AVOID message, a node computes a set of channels that may not be used as receiving channel and then sends its upstream node a SUGGEST message including the channel set. Finally, the node goes back to the IDLE state.
- WAIT: In this state, a node waits for the response of an ELIGIBLE or TOKEN message from its neighbors and moves to the TokenHandle state once it receives one of them. The node may also wait for the response of AVOID message and moves to ChannelHandle-1 state once it receives it.

E. Reliability Analysis

Recall that we assume the original mesh network is strongly connected. Therefore, the proof of 100% reliability is to prove the broadcasting tree obtained from the DIB protocol is still strongly connected.

Definition 1. *A strongly connected path (SCP) is a directed path in which only good links are included.*

Definition 2. *A directed graph or network is strongly connected if there is at least one SCP between any pair of vertices/nodes.*

Definition 3. *A directed broadcasting tree is strongly connected if there is at least one SCP from the source node to any other node.*

Lemma 1. *After phase 1 is completed, the union of all local structures is still strongly connected.*

Proof: The union of all local structures is the same as the initial strongly connected graph removing the bad links. For the initial graph, the removing of bad links does not cause the connectivity loss of the graph based on Definition 2. ■

Lemma 2. *All removing operations in phase 2 do not cause the connectivity loss of any node in the graph.*

Proof: Recall that both RO1 and RO2 remove the links between node i and its outgoing neighbor j except the ones with channel ch_{ij}. This removal does not cause the connectivity loss of nodes i and j, because node i already has a father and node j has at least one link to node i with channel ch_{ij}. RO3 removes all the links connected to node k except the one to its father. Node k maintains the connectivity because it gets the connection through its father node. ■

Lemma 3. *After phase 2 is completed, if there exists a SCP from source node s to an arbitrary node i, there also exists a SCP from s to node j, where $j \in \mathbf{N}_i^T$.*

Proof: After phase 2 is completed, there are two cases for the connection between nodes i and j. First, there exists a direct link between nodes i and j. According to the definition of SCP, a SCP that adds one good link at one end is still a SCP. Let $P(s\aleph ij)$ denote one SCP from s to i, where $\aleph$ represents a list of intermediate nodes along the path. Thus, $P(s\aleph ij)$ is also a SCP. Second, there is no direct link between nodes i and j due to the fact that all direct links between i and j are removed. From Lemma 2, the removing operations in phase 2 do not cause the connectivity loss of any node involved. Node j must have another node instead of i as its father node. The connectivity of node j is maintained through j's father, and thus there exists a SCP from node s to node j. ■

Theorem 1. *The broadcasting tree obtained from the DIB protocol is strongly connected.*

Proof: After the DIB protocol is completed, a node's connections consist of links that participate in broadcasting. The union of every node's connections is the broadcasting tree. From Lemma 3, any node in the broadcasting tree has a SCP from the source node. Thus the broadcasting tree is strongly connected. ■

Theorem 2. *The depth of the broadcasting tree obtained from the DIB protocol is bounded.*

Proof: During the execution of the DIB protocol, nodes in the network can be classified into three sets: $\mathbf{N}^B$, the set containing the nodes that have already been added to the current broadcasting tree, $\mathbf{N}^C$, the set containing the nodes that have a connection to the current broadcasting tree, and $\mathbf{N}^O$, the set containing all the other nodes in the network. Let $\overline{\mathbf{N}^O}$ denote the set of the nodes in $\mathbf{N}^O$ that have connections to some nodes in $\mathbf{N}^C$. As the process moves on, a node in $\mathbf{N}^C$ will receive a TOKEN message from a node in $\mathbf{N}^B$ and is triggered to start the message passing procedures. Upon receiving the TOKEN_RETURN message, the node either joins $\mathbf{N}^B$ or stays in $\mathbf{N}^C$. In either case, the protocol ensures that nodes in $\overline{\mathbf{N}^O}$ will join $\mathbf{N}^C$. Apparently, the size of $\mathbf{N}^O$ keeps decreasing as the TOKEN moves forward. Once $\mathbf{N}^O$ becomes empty, the construction of the broadcast tree is finished. Since the size of $\mathbf{N}^O$ is a bounded number and keeps decreasing until $\mathbf{N}^O$ is empty, the broadcast tree is built in finite steps. Therefore, the broadcasting tree has a bounded depth. In the worst case, the depth of the constructed broadcasting tree is at most N. Thus, the depth of the broadcasting tree obtained from the DIB protocol is bounded by $O(N)$. ■

F. Control Messages Overhead

Theorem 3. *The number of control messages does not exceed $4|E_{\overline{C}}|$, where $|E_{\overline{C}}|$ is the number of directed links. Notice that multiple directed links between a pair of nodes with different channels are counted once.*

Proof: We count the number of control messages node i needs to send. First, the number of TOKEN and TOKEN_RETURN messages does not exceed the number of its neighbors since node i only needs to send one TOKEN message to each child and one TOKEN_RETURN message to its father. Second, node i sends one ELIGIBLE message to each outgoing neighbor (excluding its father) and one CHOSEN message to each incoming neighbor. Third, node i sends no more than one AVOID message to each outgoing neighbor and no more than one SUGGEST message to each incoming neighbor. Fourth, node i sends no more than one NOTIFY message to each incoming neighbor. Notice that each type of message needs to be transmitted at most once between any pair of nodes since all the channel information is included in the message. In summary, no more than four control messages will traverse each directed link, and the total

TABLE I
SIMULATION CONFIGURATIONS

Size of the topography	2500 × 2500 m
Communication range	250 m
Propagation model	Two-ray ground
MAC protocol	802.11 CSMA based
Bandwidth of links	1 Mbps
Packet length (L)	250 Bytes
Traffic rate (r)	50 packets/s
Total traffic	1000 packets

number of control message is bounded by $4|E_{\overline{C}}|$. Equivalently, the message complexity of Phase 2 is $O(N^2)$, where N is the number of nodes in the network. ■

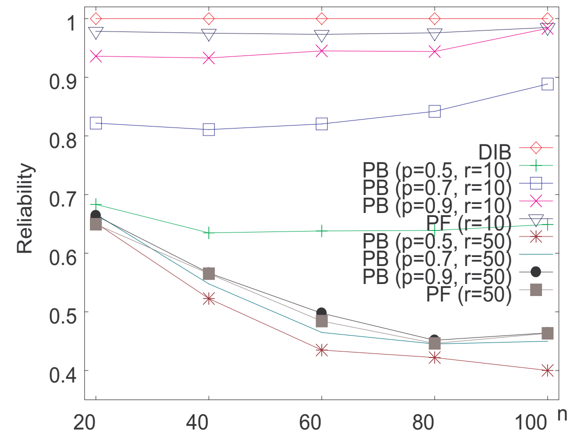


Fig. 4. Reliability versus number of nodes.

V. Simulations and Analysis

To evaluate the performance of the Distributed Interference-aware Broadcasting (DIB) protocol, we have conducted extensive simulations using ns-2. For comparison purpose the performance of Probabilistic Broadcasting (PB) and Pure Flooding (PF) are also simulated, in which a channel is randomly chosen for broadcasting. For the PB protocol, three probabilities (0.5, 0.7, and 0.9) are used to study different scenarios. Table I specifies the configurations of simulations. When deploying the network, nodes are randomly placed with a constraint of connectivity. Four performance metrics are measured: reliability, redundancy, latency, and goodput.

The reliability is defined as $Rel = \frac{\sum_{i=1}^{N} M_i}{NM}$, where M is the number of packets that the source node sends out, and M_i is the number of packets (excluding duplicates) that node i received. The average receiving redundancy is defined as $Red = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} X_{i,j}}{\sum_{i=1}^{N} M_i} - 1$, where $X_{i,j}$ is the total number of the j-th packet (including duplicates) received by node i. The transmission redundancy is indicated by the percentage of the number of nodes participating in the broadcasting. $Red_T = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} B_{i,j}}{NM}$, where $B_{i,j}$ is a 0-1 function that indicates whether node i broadcasts the j-th packet (1) or not (0). The average latency is defined as $Lat = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(t_{i,j}-t_{j,start})}{\sum_{i=1}^{N} M_i}$, where $t_{i,j}$ is the time node i receives the j-th packet, and $t_{j,start}$ is the time the source node sends out the j-th packet. The goodput of the system is defined as $Gdp = \frac{L}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\frac{1}{t_{i,j}-t_{j,start}}$, where L is the packet length. To ease the performance comparison, we define a comprehensive metric called *power*, defined as follows,

$$P = \frac{Rel \times Gdp}{Lat \times Red}$$

The power is defined in this way because a mesh network is expected to provide high reliability and goodput with small latency and redundancy. Notice that the transmission redundancy is an unclear factor of the system performance, and thus it is not used in the definition of power.

We first study the reliability of the three protocols. As can be seen from Fig. 4, the proposed DIB protocol consistently achieves 100% reliability. The PB and PF protocols, however, can not achieve 100% reliability due to serious contentions and interferences. To resolve the heavy contention problem, a longer backoff time is needed. Thus, some broadcast messages are dropped. To make the situation worse, the significant interference among adjacent nodes causes continuous collisions. That is why even the PF protocol cannot achieve 100% reliability. To further study how the traffic load impacts the reliability, two traffic rates are used in Fig. 4.

When the traffic rate $r = 50$ packets/s, both PB and PF protocols have to handle the new broadcasting messages while the previous messages are still buffered in the transmission queue. Thus, the broadcast messages keep accumulating at each node. The timing of broadcasting for the new messages is highly correlated with that for the accumulated messages. Therefore, collisions occur not only in the same broadcast message, but also among the consecutive messages. That is why the reliabilities of PB and PF protocols are decreasing while the number of nodes is increasing. When the traffic rate $r = 10$ packets/s, the contention of consecutive messages is much less and is not the dominant factor. Thus, the reliability is much higher and keeps increasing while the number of nodes is increasing. In the rest of the simulations, we compare the three protocols under a heavy traffic load ($r = 50$ packets/s).

Fig. 5 shows the average number of redundancies each node receives under different network sizes. Our DIB protocol significantly reduces the receiving redundancy because only the nodes included in the broadcast tree relay the broadcast messages and only the nodes that tune to the same channel as the transmitting nodes receive the broadcast messages. Naturally, the PF protocol performs the worst. The PB protocol reduces the receiving redundancy a little compared to the PF protocol; however, its redundancy linearly increases as the number of nodes increases. This is because the denser the network, the greater the number of neighboring nodes.

Fig. 6 shows the average transmission redundancy. Obviously the PF protocol has the highest transmission redundancy since every node is participating in broadcasting. The transmission redundancy of the PB protocol heavily replies on the chosen probability; the bigger the probability, the higher the redundancy. The transmission redundancy of our DIB protocol is only dependent on the broadcasting tree and is not related to the node degree. Thus, there is no

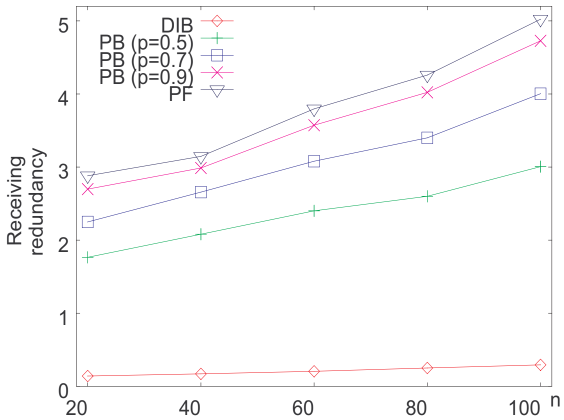


Fig. 5. Receiving redundancy versus number of nodes.

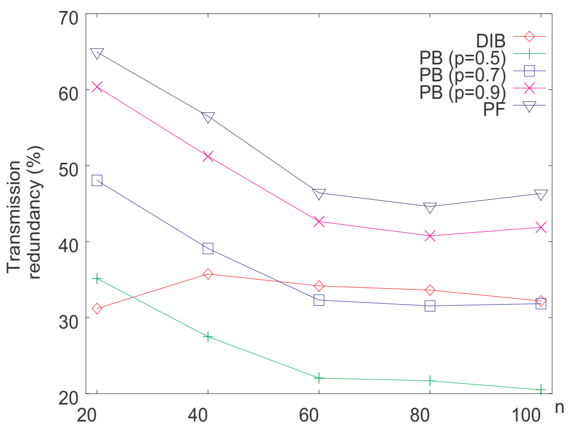


Fig. 6. Transmission redundancy versus number of nodes.

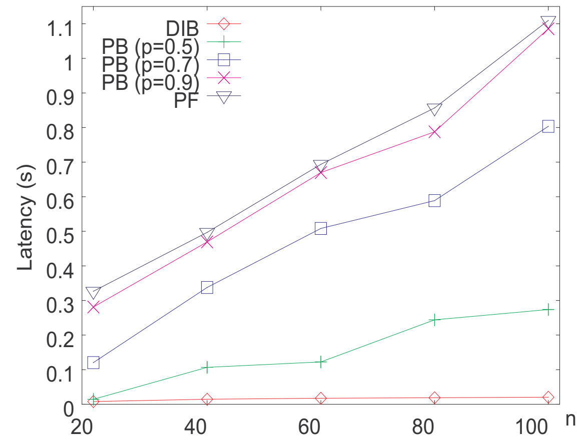


Fig. 7. Latency versus number of nodes.

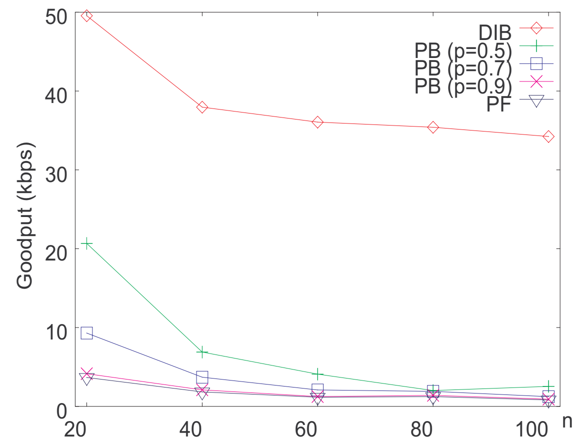


Fig. 8. Goodput versus number of nodes.

notable increment of redundancy while the number of nodes is increased. The redundancy is within the range of 30-40%. It is interesting to notice that the transmission redundancy of the PB and PF protocols is decreasing while the number of the nodes is increasing. This is due to the fact that their reliability is decreasing, and thus fewer nodes participate in the broadcasting.

The latency performance is illustrated in Fig. 7. It can be seen that both PB and PF protocols have large latencies that increase with the network size. The DIB protocol, however, consistently achieves very small latency, as explained below. In the PB and PF protocols, the large numbers of transmission and receiving redundancies results in serious collisions and thus causes longer backoff time. As shown in Fig. 5, the increase in network size further aggravates the situation. In addition, nodes that are farther from the source have larger backoff times; consequently, it takes a longer time for these nodes to receive the messages. On the other hand, our DIB protocol significantly reduces the receiving redundancy. The probability of collision is negligible; therefore most of the transmissions are successful at the first attempt. While the number of nodes increases, the broadcasting latency of the DIB protocol is only increased slightly since the ratio of the longer path nodes to the shorter path nodes is increased slightly.

Fig. 8 clearly demonstrates that the goodput of the DIB protocol significantly outperforms the PB and PF protocols. One important observation is that the goodput of all three protocols decreases as the number of nodes increases. According to the definition of goodput, each non-redundant received message contributes to the goodput. Also the goodput varies inversely with the latency. In general, a node far away from the source node has a higher probability of having a long path, and thus a larger latency, than one closer to the source node. Therefore, with the latency being inversely proportional to goodput, a node with a longer path has less goodput than the one with a shorter path. As the total number of nodes is increasing, the proportion of nodes with longer distances increased accordingly; therefore, the goodput of all three protocols is decreased. We speculate that the goodput will become saturated at some point as deploying more nodes has little impact on the proportion of path length. In Fig. 9, as can be seen, the DIB protocol significantly outperforms the other two protocols in power performance.

VI. Conclusion

We have developed two metrics to assess the link and channel qualities and a distributed interference-aware broadcasting (DIB) protocol to build a high-performance broadcasting tree for multi-radio multi-channel mesh networks. Both intra-node and inter-node interferences were taken into account

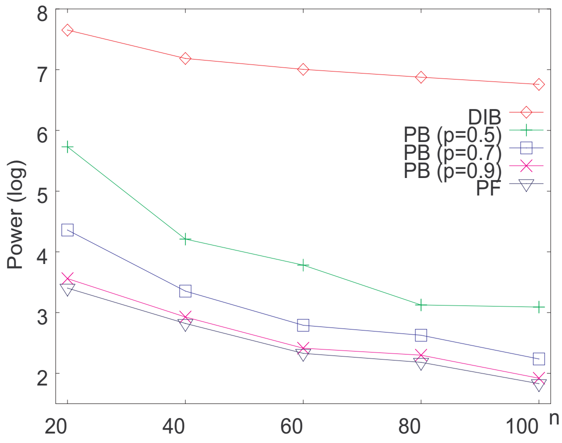


Fig. 9. Power versus number of nodes.

in the development process. Our protocol has demonstrated good scalability since the information of only 2 hops is needed. A simulator to simulate multi-radio multi-channel mesh networks has been developed to evaluate the proposed DIB protocol. Simulation results have suggested that the DIB protocol is able to achieve 100% reliability, less broadcasting redundancies, low broadcasting latency, and high goodput. To better justify the performance, a comprehensive network performance metric, called power, has been defined. For the future work, we will investigate the extent to which a local optimized tree can build a global optimum tree. The broadcast scheduling scheme will also be investigated.

Acknowledgment

The research of Min Song is supported by NSF CAREER Award CNS-0644247.

References

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, pp. 445–487, 2005.

[2] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inform. Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[3] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *MobiCom*, 2004, pp. 114–128.

[4] B. Raman and K. Chebrolu, "Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks," in *MobiCom*, 2005, pp. 156–169.

[5] G. Wang, D. Lu, W. Jia, and J. Ca, "Reliable gossip-based broadcast protocol in mobile ad hoc networks," in *MSN*, 2005, pp. 207–218.

[6] E. Pagani and G. P. Rossi, "Reliable broadcast in mobile multihop packet networks," in *MobiCom*, 1997, pp. 34–42.

[7] W. Lou and J. Wu, "A reliable broadcast algorithm with selected acknowledgements in mobile ad hoc networks," in *Proc. IEEE GLOBECOM*, 2003, pp. 3536–3541.

[8] S. Gupta and P. Srimani, "An adaptive protocol for reliable multicast in mobile multi-hop radio networks," in *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 111–122.

[9] I. Chlamtac and S. Kutten, "Tree-based broadcasting in multi-hop radio networks," *IEEE Trans. Computers*, vol. 36, no. 10, pp. 1209–1223, 1987.

[10] C. T. Chou and A. Misra, "Low latency multimedia broadcast in multirate wireless meshes," in *Proc. First IEEE Workshop on Wireless Mesh Networks (WiMesh)*, 2005.

[11] R. Gandhi, S. Parthasarathy, and A. Mishra, "Minimizing broadcast latency and redundancy in ad hoc networks," in *Proc. MobiHoc*, 2003, pp. 222–232.

[12] C. T. Chou, A. Misra, and J. Qadir, "Low latency broadcast in multi-rate wireless mesh networks," *IEEE J. Select. Areas Commun.*, special issue on wireless mesh networks, vol. 24, no. 11, pp. 2081–2091, 2006.

[13] B. Das and V. Bharghavan, "Routing in ad-hoc networks using minimum connected dominating sets," in *Proc. IEEE ICC*, vol. 1, 1997, pp. 376–380.

[14] W. Lou and J. Wu, "On reducing broadcast redundancy in ad hoc wireless networks," *IEEE Trans. Mobile Computing*, vol. 1, no. 2, pp. 111–122, 2002.

[15] H. Lim and C. Kim, "Flooding in wireless ad hoc networks," *Computer Commun.*, vol. 24, no. 3-4, pp. 353–363, 2001.

[16] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Proc. IEEE WCNC*, vol. 2, 2003, pp. 1124–1130.

[17] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proc. IEEE INFOCOM*, 2002, pp. 1707–1716.

[18] Y. Tseng, S. Ni, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless Network*, vol. 8, no. 2/3, pp. 153–167, 2002.

[19] P. Kyasanur and N. H. Vaidya, "Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks," *SIGMOBILE Mobile Computing and Communications Review*, vol. 10, no. 1, pp. 31–43, 2006.

[20] R. L. Cruz and A. Santhanam, "Optimal routing, link scheduling and power control in multi-hop wireless networks," in *Proc. IEEE INFOCOM*, vol. 1, 2003, pp. 702–711.

[21] C. Barrett, M. Drozda, A. Marathe, and M. V. Marathe, "Characterizing the interaction between routing and mac protocols in ad-hoc networks," in *MobiHoc*, 2002.

[22] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multi-hop wireless networks," in *IMC '05: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, 2005, pp. 28–28.

[23] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.

[24] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Gossip algorithms: Design, analysis and applications," in *Proc. IEEE INFOCOM*, vol. 3, 2005, pp. 1653–1664.

[25] X. Y. Li, K. Moaveninejad, and O. Frieder, "Regional gossip routing for wireless ad hoc networks," in *Proc. IEEE International Conference on Local Computer Networks*, 2003, pp. 274–275.

[26] D. Gavidia, S. Voulgaris, and M. van Steen, "A gossip-based distributed news service for wireless mesh networks," in *Proc. 3rd IEEE Conference on Wireless On demand Network Systems and Services(WONS)*, 2006.

[27] N. Li, J. C. Hou, and L. Sha, "Design and analysis of an mst-based topology control algorithm," in *Proc. IEEE INFOCOM*, 2003.

[28] X. Y. Li, Y. Wang, P. J. Wan, W. Z. Song, and O. Frieder, "Localized low-weight graph and its applications in wireless ad hoc networks," in *Proc. IEEE INFOCOM*, vol. 1, 2004, pp. 431–442.

[29] J. Qadir, C. T. Chou, and A. Misra, "Minimum latency broadcasting in multi-radio multi-channel multi-rate wireless mesh networks," in *Proc. IEEE SECON*, 2006, pp. 80–89.