

# Interference-Aware, Fully-Distributed Virtual Backbone Construction and its Application in Multi-Hop Wireless Networks

Scott C.-H. Huang, Min-Te Sun, *Member, IEEE*, Qilian Liang, Peng-Jun Wan, and Xiaohua Jia

**Abstract**—In multi-hop wireless networks, the use of virtual backbone can greatly simplify routing, broadcasting, as well as energy/bandwidth saving. However, constructing a virtual backbone is costly and time-consuming because of the inevitable transmission interference during the process of the construction. In the literature, most of virtual backbone construction algorithms did not take the interference issue into consideration. To the best of our knowledge, our proposed algorithm is the first fully-distributed, interference-aware virtual backbone construction algorithm that has a proven bound on the construction latency. Besides, our proposed algorithm can be applied to the leader election problem, and such application results in a fully-distributed and interference-aware leader election algorithm of time complexity  $O(n \log n)$  (where  $n$  is the number of nodes). This new leader election algorithm is practical in wireless networks because interference has already been dealt with; is also results in the fastest interference-aware leader election algorithm to the best of our knowledge.

**Index Terms**—Multi-hop wireless networks, interference, virtual backbone, distributed algorithm.

## I. INTRODUCTION

A MULTI-HOP wireless ad hoc network is a collection of mobile devices with RF transceivers. This type of network can be quickly deployed for many applications such as automated battlefield operations, search and rescue, and disaster relief. If two hosts are located closely together within the wireless transmission range of each other, then no real routing protocol or decision is necessary. However, if two hosts are outside their wireless transmission ranges, they could communicate only if other hosts between them are willing to relay packets for them. The advantage of such networks is its tremendous flexibility, but the disadvantage is its lack of infrastructure. This disadvantage causes many applications to be difficult ([27], [26]). For example, broadcast can easily cause excessive transmission redundancy; such problem is known

as the broadcast storm problem [17]. Beside broadcast, many other operations such as query, routing, or data collection can also generate excessive redundant messages. To remedy this type of problem, virtual backbones arise since they greatly reduce redundant messages and improve performance. In addition, other uses of virtual backbones include topology control, point/area coverage, as well as routing protocol design. In the literature, *Connected Dominating Sets* (CDS) are quite often constructed to play the role of virtual backbones as they exhibit many geometrical properties ([11], [7]). Although non-CDS virtual backbones do exist, they generally lack the geometrical properties that can be used in the above-mentioned applications. Therefore we only consider CDS type of virtual backbones in this paper. However, constructing a CDS is costly and time-consuming because the construction process will cause lots of collision and interference. In the literature, although there are already many CDS construction algorithms, they either do not guarantee the construction latency (such as in [1] or [6]), or they do not take interference into consideration (such as in [2], [3], [4], [13], [14]). These drawbacks motivated us to design a practical CDS construction algorithm that has the following properties: (1) *It is a fully-distributed algorithm.* (2) *It is interference-aware.* (3) *It has an explicit bound on its construction latency.*

In this paper, we design a CDS construction algorithm achieving these three goals. In addition, our algorithm has another important application to the *leader election problem*. We could apply our proposed algorithm to the leader election problem such that the combined algorithm is a fully-distributed, is interference-aware, and has an explicit bound on its time complexity. We attempt to achieve these goals because the current state-of-the-art leader election algorithm [9] did not take interference into consideration and is therefore not practical in wireless networks. We show that our algorithm, combined with [9], can achieve the same time complexity  $O(n \log n)$  as [9], which turned out to be the fastest interference-aware leader election algorithm. In other words, as an important application, our proposed algorithm solved the interference problem of [9] while preserving its overall time complexity. We believe that our proposed algorithm can still be applied to many other classical distributed algorithms besides the leader election problem to deal with interference without compromising their efficiency.

The rest of this paper is organized as follows. We put the preliminaries in Section II. We then introduce our virtual

Paper approved by B. Sikdar, the Editor for Wireless Packet Access and Cross-Layer Design of the IEEE Communications Society. Manuscript received March 10, 2009; revised February 17, 2010.

S. C.-H. Huang is with the Department of Electrical Engineering, National Tsing Hua University, Taiwan (e-mail: chhuang@ee.nthu.edu.tw).

M.-T. Sun is with the Department of Computer Science and Information Engineering, National Central University, Taiwan (e-mail: msun@csie.ncu.edu.tw).

Q. Liang is with the Department of Electrical Engineering, University of Texas at Arlington, TX, USA (e-mail: liang@uta.edu).

P.-J. Wan is with the Department of Computer Science, Illinois Institute of Technology, IL, USA (e-mail: wan@cs.iit.edu).

X. Jia is with the Department of Computer Science, City University of Hong Kong, Hong Kong SAR (e-mail: jia@cs.cityu.edu.hk).

Digital Object Identifier 10.1109/TCOMM.2010.101910.090144

backbone construction algorithm in Section III. We applied our proposed algorithm to the leader election problem in Section IV. We show our simulation results in Section V, related works in Section VI, and conclusion in Section VII. We put all proofs in the appendix.

## II. PRELIMINARIES

### A. Network Model and Problem Definition

An ad hoc network can be modeled as a unit disk graph  $G = (V, E)$ . The transmission range of each node is one. Two nodes  $u, v$  are neighbors if and only if their Euclidean distance is less than one. Time, assumed to be discrete, is represented as time slots throughout this paper. We assume each node in the network is equipped with a synchronized clock. Each node is able to read a variable, denoted by  $Time$ , representing its clock value. We also assume that each node  $x$  has a GPS device that can be used to retrieve its location. The problem formulation of this work can be described as follows. *Given a unit disk graph  $G = (V, E)$ . Construct an interference-aware virtual backbone in a distributed fashion.*

### B. Key Terms, Methods, and Concepts

Now we introduce some key terms and concepts that will be used throughout this work.

*Virtual backbone:* In a network  $G = (V, E)$ , a virtual backbone can be defined as a connected dominating set  $S \subset V$ , in which for any node  $v \in V$  we have the requirement that either  $v \in S$  or  $v$  is adjacent to some node in  $S$ . Details of virtual backbones and connected dominating sets can be found in [7] or [11].

*Maximal Independent Sets (MIS):* A subset  $S \subset V$  is an *independent set* of  $G$  if the nodes in  $S$  are pairwise non-adjacent, and an MIS  $S$  of  $G$  is an independent set of  $G$  while no proper superset containing  $S$  is an independent set of  $G$ . Any node ordering  $v_1, v_2, \dots, v_n$  of  $V$  induces an MIS  $S$  in the following first-fit manner. Initially,  $S = \{v_1\}$ . For  $i = 2$  up to  $i = n$ , add  $v_i$  to  $S$  if  $v_i$  is not adjacent to any node in  $S$ . Details of MIS can be found in [22].

*Bipartite and tripartite graphs:* A graph  $G = (V, E)$  is bipartite if its vertex set  $V$  is the disjoint union of two independent sets  $V_1, V_2$ .  $G$  is tripartite if  $V$  is the disjoint union of three independent sets  $V_1, V_2, V_3$ . We usually represent a bipartite graph as  $(X, Y, E)$  in which  $X, Y$  are the independent sets and  $E$  is the edge set. Similarly, we can use  $(X, Y, Z, E)$  to represent a tripartite graph.

*Local transmission schedule:* Since we only focus on fully-distributed algorithms in this work, we only discuss local transmission schedules with respect to a particular node. A local transmission schedule with respect to a node can be represented as a sequence of numbers  $\{a_1, a_2, \dots\}$ , in which  $a_i \in \mathbb{N} (\forall i)$  and  $a_i < a_j (\forall i < j)$ , meaning this node is scheduled to transmit in time slots  $a_1, a_2, \dots$

*Hexagonal tessellation and colorings:* A tessellation of the plane is a way of partitioning it into identical (or similar) pieces. A hexagonal tessellation is partitioning the entire plane into hexagons, as shown in Fig. 1 (a). Each hexagon is half open, half closed, without both the topmost and the

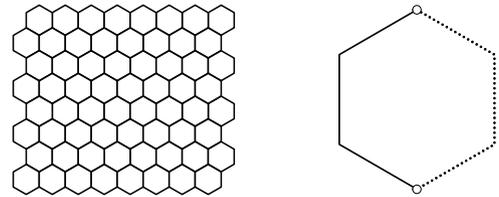


Fig. 1. (a) Hexagonal tessellation (b) One hexagon

bottommost points, as shown in Fig. 1 (b). We can color this tessellation in various ways. In this paper, we will be using two colorings: 12 and 37 coloring. The 12- and 37-colorings, denoted by  $C_{12}$  and  $C_{37}$ , are shown in Fig. 2(a)(b)(c), respectively. Note that a coloring essentially associates each hexagon with a color, represented as an integer. Since the plane is a disjoint union of all hexagons (as we define them as half-open, half-closed and no overlapping happens), we can associate each point on the plane with the color of the hexagon containing it. We can therefore view a coloring as a mapping from a point on the plane to a fixed subset  $\{1, \dots, k\}$  of integers (representing colors) as follows.  $C : \mathbb{R} \times \mathbb{R} \rightarrow \{1, \dots, k\} \subset \mathbb{N}$ . If we use  $k$  integers to represent the colors (i.e. the size of the subset is  $k$ ), we also call it a  $k$ -coloring. For each point  $x$  on the plane, we denote the corresponding  $C_{12}$  and  $C_{37}$  colors by  $C_{12}(x)$  and  $C_{37}(x)$ , respectively, as shown in Fig. 2(c). In this paper, we assume that each node  $x$  has the knowledge of its own coloring information  $C_{12}(x)$  and  $C_{37}(x)$ . Such assumption is reasonable if each node is equipped with a GPS device can have an estimate of its location. Once its own location is known, each node can simply compare its location to a predefined reference point and get its own coloring information in a purely distributed manner. Such comparison only takes constant time.

## III. INTERFERENCE-AWARE VIRTUAL BACKBONE CONSTRUCTION

We introduce our main algorithm in this section. The dependence relations of our algorithms are illustrated in Fig. 3. Note that, except Algs. 1 and 2, which are very straightforward, the input and output of every other algorithm are presented as its *precondition* and *postcondition*. We consider the precondition and postcondition since, in many cases, the input or output of an algorithm may not be concrete. For example, the output of an algorithm may be a *property* instead of a set. We define the precondition and postcondition as follows. A precondition is the requirement of running an algorithm correctly, while its corresponding postcondition is the result of running the algorithm provided that its precondition holds.

As illustrated in Fig. 3, Algs. 1–4 altogether form the virtual backbone construction algorithm, of which [16] is one of the preconditions. The virtual backbone's vertex set  $VB$  consists of two subsets: the set of *dominators* denoted by  $DOM$  and the set of *connectors* denoted by  $CON$ . If a node is not a dominator, then it is *dominatee*. Thus, connectors must be dominatees, but the reverse may not be true. Note that all algorithms (Algs. 1–4 and [16]) are run locally at each node in parallel in a fully-distributed fashion.  $DOM$  is simply constructed by running [16]. The construction method

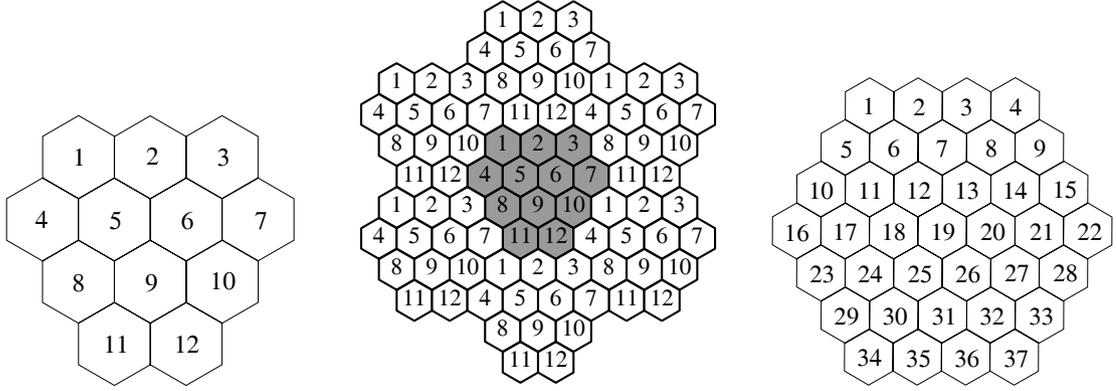


Fig. 2. (a) 12-coloring (b) 12-coloring filling up the plane (c) 37-coloring

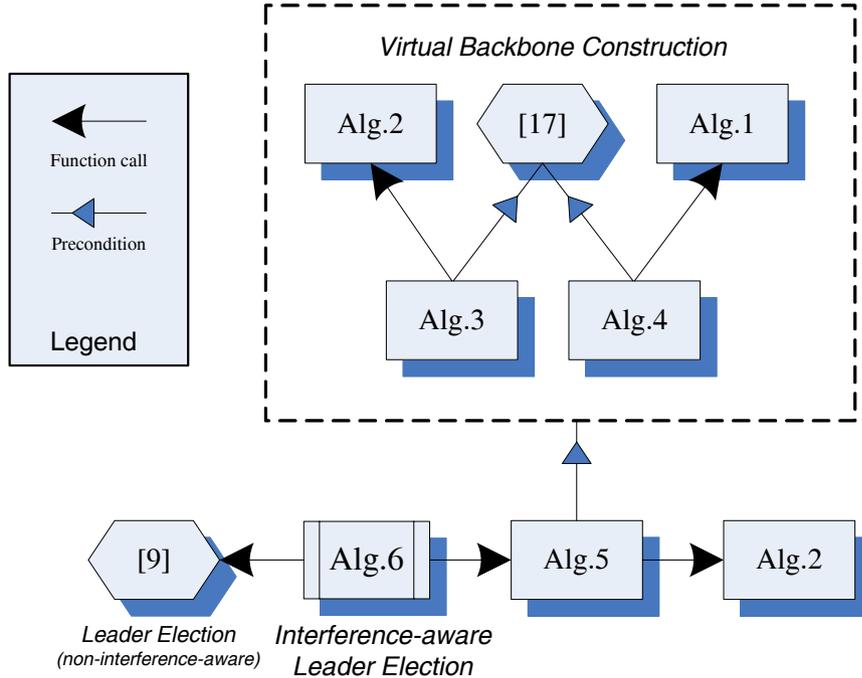


Fig. 3. Dependence relation of our algorithms. An algorithm  $\mathcal{A}$  depends on another algorithm  $\mathcal{B}$  if either (1)  $\mathcal{A}$  invokes  $\mathcal{B}$  as a subroutine, or (2)  $\mathcal{B}$  is a precondition of  $\mathcal{A}$ . The former and latter cases are categorized as *function call* and *precondition*, respectively.

of *CON* has two cases for the following reason. Since the construction method is a distributed algorithm that runs locally at a node  $x \in V$ , depending on whether or not  $x \in DOM$  we execute different tasks. Thus, the construction method is represented as two separate algorithms: Algs. 3 and 4, and they should be regarded as one single method.

In the process of virtual backbone construction, we need to invoke two subroutines repeatedly. Thus, we present these two subroutines as Algs. 1 and 2, in which Alg. 1 is called the *One-hop Broadcast Schedule (OBS)* and Alg. 2 is called *Dominators' Transmission Schedule (DTS)*.

Alg. 1 is essentially a *neighbor-to-neighbor handshaking* algorithm, which can be applied in a distributed way such that every node can transmit message  $\mathcal{M}$  to all of its neighbors with a very high probability. Note that it is a randomized algorithm of *Las Vegas* type, meaning that the postcondition is not guaranteed to happen (even if the precondition holds) but it happens with a very high probability. Alg. 2 is a subrou-

---

#### Algorithm 1 OBS (run at a node $x \in V$ )

---

**Input:** Starting time  $T_s$ , message  $\mathcal{M}$ , a constant  $C$ .

**Precondition:**  $x$  knows the bound  $\Delta$  on the maximum node degree in the network.  $C > 0$ .

**Postcondition:** All neighbors of  $x$  receive  $\mathcal{M}$  successfully from  $x$ .

- 1: Define  $T_{OBS} \stackrel{\text{def}}{=} \lceil e(\ln(n\Delta + 1) + C)\Delta / (1 - 1/\Delta) \rceil$
  - 2: **repeat** the following for each time slot
  - 3:     Schedule  $x$  to transmit  $\mathcal{M}$  with probability  $1/\Delta$
  - 4: **until**  $Time > T_s + T_{OBS}$
- 

---

#### Algorithm 2 DTS (run at a node $d \in DOM$ )

---

**Input:** Starting time  $T_s$  and message  $\mathcal{M}$

**Precondition:**  $d$  knows its color  $C_{12}(d)$ .

**Postcondition:** All neighbors of  $d$  receive  $\mathcal{M}$  successfully from  $d$ .

- 1: Schedule  $d$  to transmit  $\mathcal{M}$  at  $Time = T_s + C_{12}(d)$ .
-

---

**Algorithm 3** Virtual Backbone Construction (run at  $d \in \text{DOM}$  with starting time  $T_s$ )

---

**Pre-/Post-condition:** shown in Table I

- 1:  $d$  locally constructs a bipartite graph  $\mathcal{B} = (V_{\mathcal{D}}, V_{\mathcal{C}}, E_{\mathcal{B}})$ , a tripartite graph  $\mathcal{T} = (V'_{\mathcal{D}}, V''_{\mathcal{C}}, V'_C, E_{\mathcal{T}})$  and initializes both of them to  $\emptyset$ .
  - 2:  $d$  sends its own ID by applying Alg. 2.
  - 3:  $d$  keeps listening until  $\text{Time} > T_s + 12 + T_{OBS}$ . Upon receipt of each message  $\{(d_1, \dots, d_j); w\}$ ,  $d$  updates  $\mathcal{B}$  as follows.  $d$  adds the vertices  $d_1, \dots, d_j$  to  $V_{\mathcal{D}}$ , the vertex  $w$  to  $V_{\mathcal{C}}$ , and the edges  $(d_1, w), \dots, (d_j, w)$  to  $E_{\mathcal{B}}$ .
  - 4:  $d$  keeps listening until  $\text{Time} > T_s + 12 + 2T_{OBS}$ . Upon receipt of each message  $\{(d'_1, w'_1), \dots, (d'_k, w'_k); w'\}$ ,  $d$  updates  $\mathcal{T}$  as follows.  $d$  adds the vertices  $d'_1, \dots, d'_k$  to  $V'_{\mathcal{D}}$ ,  $w'_1, \dots, w'_k$  to  $V''_{\mathcal{C}}$ , and the vertex  $w'$  to  $V'_C$ .  $d$  also adds the edges  $(d'_1, w'_1), \dots, (d'_k, w'_k)$  and  $(w'_1, w'), \dots, (w'_k, w')$  to  $E_{\mathcal{T}}$ .
  - 5: Let  $V_{\mathcal{D}} = \{\tilde{d}_1, \dots, \tilde{d}_l\}$ , then for each  $\tilde{d}_i$ ,  $d$  finds the node  $\tilde{w}_i \in V_{\mathcal{C}}$  adjacent to  $\tilde{d}_i$  in  $\mathcal{B}$  with the largest ID as the sole designated connector for  $d$  and  $\tilde{d}_i$ .
  - 6: Let  $V'_{\mathcal{D}} = \{\tilde{d}'_1, \dots, \tilde{d}'_m\}$ , then for each  $\tilde{d}'_i$ ,  $d$  compares the IDs of itself and  $\tilde{d}'_i$ . If  $ID(d) > ID(\tilde{d}'_i)$ ,  $d$  looks at all 2-hop neighbors of  $\tilde{d}'_i$  in  $V'_C$  and picks  $c'_i$  with the largest ID as the first designated connector for  $d$  and  $\tilde{d}'_i$ .  $d$  then looks at the nodes in  $V''_{\mathcal{C}}$  adjacent to both  $c'_i$  and  $\tilde{d}'_i$ , and picks  $c''_i$  with the largest ID as the second designated connector for  $d$  and  $\tilde{d}'_i$ . If  $ID(d) < ID(\tilde{d}'_i)$ ,  $d$  looks at all neighbors of  $\tilde{d}'_i$  in  $V'_C$  and picks  $c'_i$  with the largest ID as the second designated connector for  $d$  and  $\tilde{d}'_i$ .  $d$  then looks at the nodes in  $V''_{\mathcal{C}}$  adjacent to  $c'_i$ , and picks  $c''_i$  with the largest ID as the first designated connector for  $d$  and  $\tilde{d}'_i$ .
  - 7:  $d$  sends  $\{d; (\tilde{w}_1, \tilde{d}_1), \dots, (\tilde{w}_k, \tilde{d}_k); (c'_1, c''_1, \tilde{d}'_1), \dots, (c'_m, c''_m, \tilde{d}'_m)\}$  by applying Alg. 2.
- 

time dedicated to dominators. It is a deterministic scheduling algorithm ensuring that all neighbors of each dominator get the messages within 12 time slots. Alg. 1 is an important subroutine in the virtual backbone construction method when each dominator needs to transmit a message to all of its neighbors. Similarly, Alg. 2 is an important subroutine when each connector needs to transmit a message to all of its neighbors. Algs. 3 and 4 are the main construction algorithm. We list the following theorems regarding the correctness of these algorithms. Their proofs are presented in the appendix.

*Theorem 1:* In Alg. 1, if the precondition is satisfied, then, after execution, the overall probability that the postcondition holds for all nodes in the network is at least  $e^{-e^{-C}}$ .

*Theorem 2:* In Alg. 2, if the precondition is satisfied, then the postcondition holds after execution.

*Theorem 3:* If the precondition holds, then after applying Alg. 3 on each dominator and applying Alg. 4 on each dominatee in parallel, the postcondition holds with at least probability  $e^{-2e^{-C}} \approx 1$ , where  $C$  is selected in Alg. 1.

*Theorem 4:* The execution time of both Alg. 3 and Alg. 4 is  $O(\Delta \log n)$ .

The precondition and postcondition of both Algs. 3 and 4

---

**Algorithm 4** Virtual Backbone Construction (run at  $w \notin \text{DOM}$  with starting time  $T_s$ )

---

**Pre-/Post-condition:** shown in Table I

- 1:  $w$  locally constructs a set  $\mathcal{D}$  and a bipartite graph  $\mathcal{B} = (V_{\mathcal{D}}, V_{\mathcal{C}}, E_{\mathcal{B}})$ . Both of them are initialized to  $\emptyset$ .
  - 2:  $w$  keeps listening until  $\text{Time} > T_s + 12$ . During this listening period, upon receipt of each message  $\{d\}$ ,  $w$  adds  $d$  to  $\mathcal{D}$ .
  - 3: When  $\text{Time} = T_s + 12$ ,  $w$  sends the message  $\{(d_1, \dots, d_k); w\}$  by applying Alg. 1, in which  $\{d_1, \dots, d_k\} = \mathcal{D}$  (Note that  $k \leq 5$  according to Lemma 2). In the meantime, when  $w$  is not sending,  $w$  keeps listening and does the followings upon receipt of each message  $\{(d'_1, \dots, d'_l); w'\}$ .  $w$  adds the vertices  $d'_1, \dots, d'_l$  to  $V_{\mathcal{D}}$ , the vertex  $w'$  to  $V_{\mathcal{C}}$ , and the edges  $(d'_1, w'), \dots, (d'_l, w')$  to  $E_{\mathcal{B}}$ . This step terminates at  $\text{Time} = T_s + 12 + T_{OBS}$ .
  - 4: Suppose, at this moment,  $V_{\mathcal{D}} = \{d''_1, \dots, d''_m\}$ . For each dominator  $d''_i \in V_{\mathcal{D}}$ ,  $w$  finds the node  $w''_i \in V_{\mathcal{C}}$ , adjacent to  $d''_i$  with the largest ID, and sends the message:  $\{(d''_1, w''_1), \dots, (d''_m, w''_m); w\}$  by applying Alg. 1. (Note that  $m \leq 21$  according to Lemma 3).
  - 5:  $w$  keeps listening until  $\text{Time} > T_s + 24 + 2T_{OBS}$ . Upon receipt of each message  $\{\tilde{d}; (\tilde{w}_1, \tilde{d}_1), \dots, (\tilde{w}_p, \tilde{d}_p); (c'_1, c''_1, \tilde{d}'_1), \dots, (c'_q, c''_q, \tilde{d}'_q)\}$ ,  $w$  checks each pair and triple in this message and does the followings. If  $w = \tilde{w}_i$  for some  $i$ , then  $w$  saves  $(\tilde{d}, \tilde{w}_i, \tilde{d}_i)$ . If  $w = c'_j$  or  $w = c''_j$  for some  $j$ , then  $w$  saves  $(\tilde{d}, c'_j, c''_j, \tilde{d}'_j)$ .
- 

are lengthy, so we present them in Table I instead of putting them in the algorithm description.

**Some Remarks about Algs. 1–4:** The expected result after executing Alg. 1 is the situation that all nodes in the network receives successfully from all of its neighbors. Here by receiving successfully is referred from a receiver's point of view with respect to a sender, which means the receiver will receive successfully at least once successfully from this sender. Note that the result of Theorem 1 is very strong. As long as there is one node that fails to receive successfully from one of its neighbors, the postcondition is considered to be failed. Moreover, this result happens with a high probability depending on the input  $C$ . As  $C$  grows, the probability  $e^{-e^{-C}}$  grows very quickly to 1. For example, If  $C = 10$ , then  $e^{-e^{-C}} \approx 0.99995$ . In Table II, we show how  $e^{-e^{-C}}$  grows with respect to  $C$ . In practice  $C = 9$  is more than enough to ensure Alg. 1 works. In what follows, we simply assume  $C$  is a predefined constant and we will not specify its exact value in the description of Alg. 3 and Alg. 4.

Since the ID of a node appears very often in Algs. 3–6, for simplicity, if  $x$  is a node, we use  $x$  itself to refer to its ID in our algorithm description. In other words, when we state *sending  $x$*  in our algorithm description, we actually refer to *sending the ID of  $x$* . Moreover, due to the nature of omnidirectional wireless transmission, in our algorithm we may omit the receivers to simplify the description: “*node  $x$  sends a message*” means “*node  $x$  sends a message to its neighbors*”.

TABLE I  
PRECONDITION AND POSTCONDITION OF ALGS 3 AND 4

PRECONDITION	
1)	Each node has a unique ID and there is a bound $\Delta$ on the maximum degree.
2)	Each node $x$ has the knowledge of its own ID, whether or not $x \in DOM$ , and $\Delta$ .
3)	Each dominator $d$ has the knowledge of $C_{12}(d)$ .
POSTCONDITION	
1)	For any two dominators $d_1, d_2$ within 3 hops, suppose $d_1$ choose $c_1, c_2$ as the first and second designated connectors and $d_2$ choose $c'_1, c'_2$ as the first and second designated connectors according to Alg. 3 and Alg. 4, then $c_1 = c'_2$ and $c_2 = c'_1$ . (In other words, $d_1, d_2$ must agree on their choice of connectors.)
2)	Each dominator has the knowledge of its all 2- and 3-hop neighboring dominators and the corresponding designated connectors.
3)	Each dominatee knows whether or not it is a sole, first, or second designated connector for all dominators within 2 hops. Each connector has the knowledge of all the dominator pairs within 2 hops for which it is selected as the sole, first, or second designated connector.
4)	Let $J \stackrel{\text{def}}{=} G _{DOM \cup CON}$ (the subgraph of $G$ restricted to the vertex set $DOM \cup CON$ ). Then $J$ is connected. In other words, $DOM \cup CON$ forms a connected dominating set.

TABLE II  
How  $e^{-e^{-C}}$  GROWS WITH  $C$

$C$	1	2	3	5	7	10	15	20
$e^{-e^{-C}}$	0.69	0.87	0.95	0.993	0.999	0.99995	0.9999997	0.99999998

Note that we only connect the dominators within three hops in Algs. 3 and 4. Why don't we connect the dominators within two hops or beyond three hops? Because connecting all dominators within three hops guarantees the connectivity of the resulted virtual backbone. One one hand, if we only connect the dominators within two hops, then resulted virtual backbone  $DOM \cup CON$  may not be connected. For more details, please see the postcondition (4) in Table I and its proof presented in Lemma 3 in the appendix. On the other hand, although connecting the dominators beyond three hops also guarantees the connectivity, doing so would increase the communication latency among the dominators.

Because our proposed algorithm is fully-distributed, no node has the whole picture of the virtual backbone and each node is only aware of its neighbors within three hops. If  $d_1, d_2$  are two dominators and they are 2-hop neighbors, after running our proposed algorithm, a connector  $c$ , called the *sole designated connector*, will be selected to connect  $d_1, d_2$ . Moreover, both  $d_1, d_2$ , and  $c$  will know about this selection. If  $d_1, d_2$  are 3-hop neighbors, after running our proposed algorithm, two nodes  $c_1, c_2$  will be selected as the designated connector and  $(d_1, c_1), (c_1, c_2), (c_2, d_2)$  must be edges in the network and both  $d_1, d_2, c_1, c_2$  will know about this selection. Here, from  $d_1$ 's point of view, we call  $c_1$  the *first designated connector* and  $c_2$  the *second designated connector*. Therefore, from  $d_2$ 's point of view,  $c_2$  is the first designated connector.

Note that [16] is part of the precondition of Algs. 3 and 4. Moreover, the execution time of [16] as well as Algs. 3 and 4 are  $O(\log^2 n)$  and  $O(\Delta \log n)$ , respectively. Therefore, the overall virtual backbone construction latency is  $O(\max(\Delta, \log n) \cdot \log n)$ , which is at most  $O(n \log n)$ . We did not analyze the worst case of our proposed algorithms for the following reasons. It is not possible to analyze the worst case of a randomized algorithm of Las Vegas type due to its probabilistic nature. The postcondition of this type of algorithm does not hold deterministically, so the worst case cannot be determined.

#### IV. APPLICATION ON THE LEADER ELECTION PROBLEM AND OTHER DISTRIBUTED ALGORITHMS

In this section, we are going to show how to apply our proposed algorithm in a particular way to many classical distributed algorithms, particularly the *leader election problem*. The leader election problem is defined as follows.

*Problem 1:* (Leader Election) Suppose each node can transmit or receive a message with its 1-hop neighbors at any time. The objective is to find a transmission schedule such that, after applying this schedule, each node can locally make a true-false decision about whether itself is a leader. Moreover, exactly one node in the network decides that itself is a leader and every other node decides that itself is a non-leader.

There is a classical algorithm [9] for the leader election problem. However, interference was not taken into consideration in designing this algorithm and therefore it is not applicable in wireless networks. Here we will show that, we can actually apply [9] in a particular way such that it becomes an interference-aware leader election algorithm. In order to do that, we have to introduce the *Unit Round Schedule*, presented in Alg. 5. Having this, we can ensure that after each execution, each dominator can exchange a message collision-free with each of its 2-hop or 3-hop neighboring dominators. In other words, each execution of Alg. 5 corresponds to one round of [9], which is not interference-aware. We can thus transform a non-interference aware leader election algorithm to an interference-aware one, as described in Alg. 6.

*Theorem 5:* In Alg. 5, if the precondition is satisfied, then the postcondition holds after execution.

*Theorem 6:* The execution time of Alg. 5 is 1972 time slots.

*Theorem 7:* In Alg. 6, if the precondition is satisfied, then the postcondition holds after execution.

*Theorem 8:* The execution time of Alg. 6 is  $O(n \log n)$ , where  $n$  is the number of nodes in the network.

Finally we want to make some justifications for the precondition of Alg. 5, particularly on first designated connectors'

**Algorithm 5** Unit Round Schedule (run at  $x \in CDS$ )**Input:** Starting time  $T_s$  and message  $\mathcal{M}$ **Precondition:** Each dominator maintains a sorted ID list of the dominators within 3 hops, sorted in decreasing order. Each dominatee knows  $C_{12}(d)$  and  $C_{37}(d)$  for each dominator  $d$  within 2 hops. Each first designated connector knows, for each neighboring dominator  $d'$  and each 2-hop neighboring dominator  $d''$ , the place of  $d''$  on the (sorted) ID list of  $d'$ , denoted by  $pl(d', d'')$ .▷ (Suppose the (sorted) ID list of  $d'$  is  $\{d''_1, d''_2, \dots, d''_k\}$ , then  $pl(d', d''_i) \stackrel{\text{def}}{=} i$ . Moreover,  $pl(d', d''_i) \leq 40$  according to Lemma 4)**Postcondition:** All dominators within 3 hops receive  $\mathcal{M}$  successfully through relaying.

- 1: **if**  $x \in DOM$  **then**
- 2:    $x$  sends  $\{\mathcal{M}; x; (d_1, \dots, d_k)\}$  by applying DTS with starting time  $T_s$ .
- 3: **end if**
- 4: **if**  $x \in CON$  **then**
- 5:    $x$  keeps listening until  $Time > T_s + 12$ . Upon receipt of each message  $\{\mathcal{M}; y; (d'_1, \dots, d'_i)\}$ ,  $x$  checks whether itself is the 1<sup>st</sup> or sole designated connector for  $y$  and each  $d'_j$ . If yes,  $x$  relays the message  $\{\mathcal{M}, y, d'_j\}$  at  $Time = T_s + 12 + (C_{37}(y) - 1) * 40 + pl(y, d'_j)$ . Otherwise  $x$  does nothing.
- 6:    $x$  keeps listening until  $Time > T_s + 1492$ . Upon receipt of a message  $\{\mathcal{M}, z, d^*\}$ ,  $x$  checks whether itself if the 2<sup>nd</sup> designated connector for  $z$  and  $d^*$ . If yes, it simply relays message  $\mathcal{M}$  at  $Time = T_s + 1492 + (C_{12}(d^*) - 1) * 40 + pl(d^*, z)$ . Otherwise it does nothing.
- 7: **end if**

knowledge of  $pl(d', d'')$ . Note that these conditions are reasonable assumptions as long as each dominator  $d$  knows its colors  $C_{12}(d)$  and  $C_{37}(d)$  for the following reason. When we run Alg. 3/Alg. 4, we can have each dominator attach its coloring information in the message, sent in line 2 of Alg. 3. This way we can guarantee that after running Alg. 3/Alg. 4 each dominator will be able to gather this coloring information from each dominator within 3 hops and each related connector will be able to gather corresponding pieces of information too. As for the knowledge of  $pl(d', d'')$ , we know that at the beginning of line 7 of Alg. 3, each dominator has already gathered all dominator information within 3 hops, and it can simply append this ID order information to the message and send to its first designated connector. Therefore, this condition can always be satisfied this way.

## V. SIMULATION RESULTS

In all of the experiments, nodes are uniformly deployed at random within a fixed region  $150m \times 150m$ . All nodes have the same transmission range. We made 10 sets of experiments, corresponding to 10 different network deployments. In each set, we ran 100 rounds for each configuration (the same number of nodes and the same transmission range). All data in the following figures were averaged over these sets and rounds. In all figures, for each value on the horizontal axis,

we show the following three corresponding values on the vertical axis: the maximum, the average, and the minimum of observed values. Throughout these experiments, the total time slots of OBS and virtual backbone construction are measured. In addition, the success probability of OBS is also estimated. Fig. 4 and Fig. 5 show the relation between latency and transmission range as well as the number of nodes, respectively.

In Fig. 4, we did the experiment with 200 nodes. We measured the number of time slots (i.e. latency) for OBS and virtual backbone construction with transmission ranges varying from  $10m$  to  $100m$ . Fig. 4(a) indicates that the chance of successful OBS achieves more than 95%, which matches with our theoretical estimation. It's interesting to note that initially the time for OBS increases steadily but then it gradually increases much more slowly when the transmission range reaches certain threshold ( $80m$  in this case). In addition, we observe that experimental values are considerably lower than our theoretical bound, indicating that our algorithm is pessimistic and only considered the worst case. These results also hold for the virtual backbone construction (shown in Fig. 4(b)), since OBS plays a dominating role on the performance of constructing the virtual backbones in our algorithm.

In Fig. 5, the transmission range of each node is fixed to  $50m$ . We measured latency as well but the number of nodes varies from 50 to 300. The performance results are similar to Fig. 4 except that latency keeps increasing steadily as the number of nodes increases.

**Algorithm 6** Interference-Aware Leader Election (run at one node  $x$ )**Precondition:** Each  $x$  has a unique ID and knows  $C_{12}(x)$ ,  $C_{37}(x)$ , as well as  $\Delta$ .**Postcondition:** Exactly one node in the network declares itself as the leader and all other nodes know the leader's ID.

- 1: Run a distributed MIS algorithm (such as [16]) locally on  $x$  to determine whether itself is a dominator.
- 2: If  $x \in DOM$ , apply Alg. 3 on  $x$ . Otherwise, apply Alg. 4 on  $x$  to construct a virtual backbone  $DOM \cup CON$ .
- 3: **if**  $x \in DOM$  **then**
- 4:   Apply the leader election algorithm in [9] locally on  $x$  in conjunction with URS as follows. Whenever  $x$  is scheduled to transmit at  $Time = T$ , run URS on  $x$  with starting time  $1972(T - 1)$ .
- 5: **end if**

## VI. RELATED WORK

The idea of using a Connected Dominating Set (CDS) to serve as a virtual backbone in mobile ad hoc networks has been well-studied in the literature. Das and Bharghavan [11] used a Minimum Connected Dominating Set (MCDS) to support unicast, multicast, as well as fault-tolerant routing in ad hoc networks. In their algorithm, primary routing is still achieved by using shortest-paths, but the virtual backbone is used to provide backup routes to handle interim failures.

Sivakumar *et al* [18], [19], Das, Bharghavan *et al* [5], [12] also designed MCDS algorithms and used it extensively in

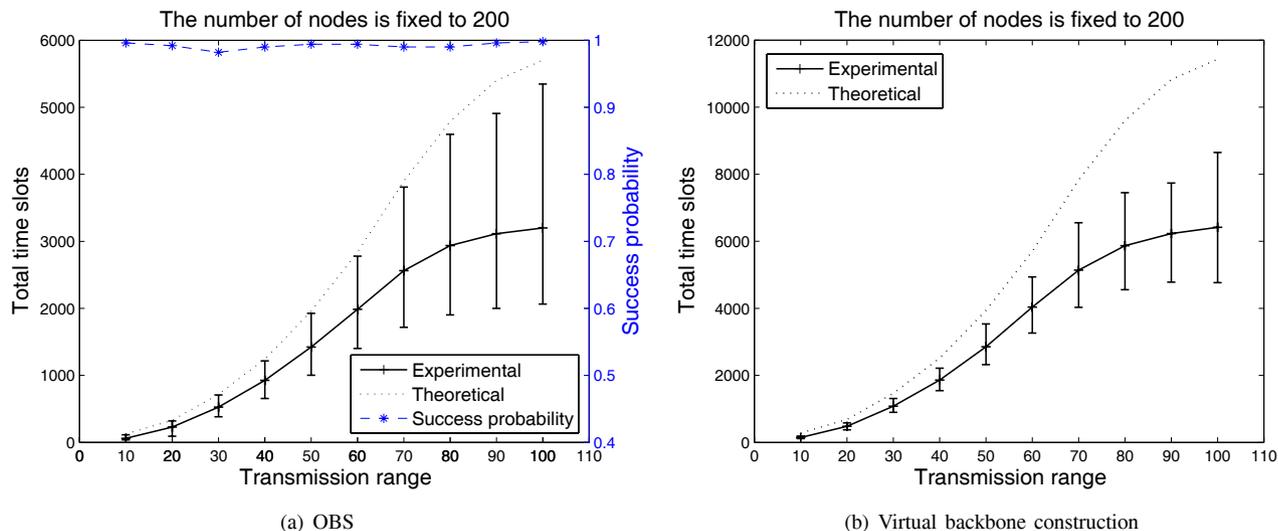


Fig. 4. Time vs. transmission range.

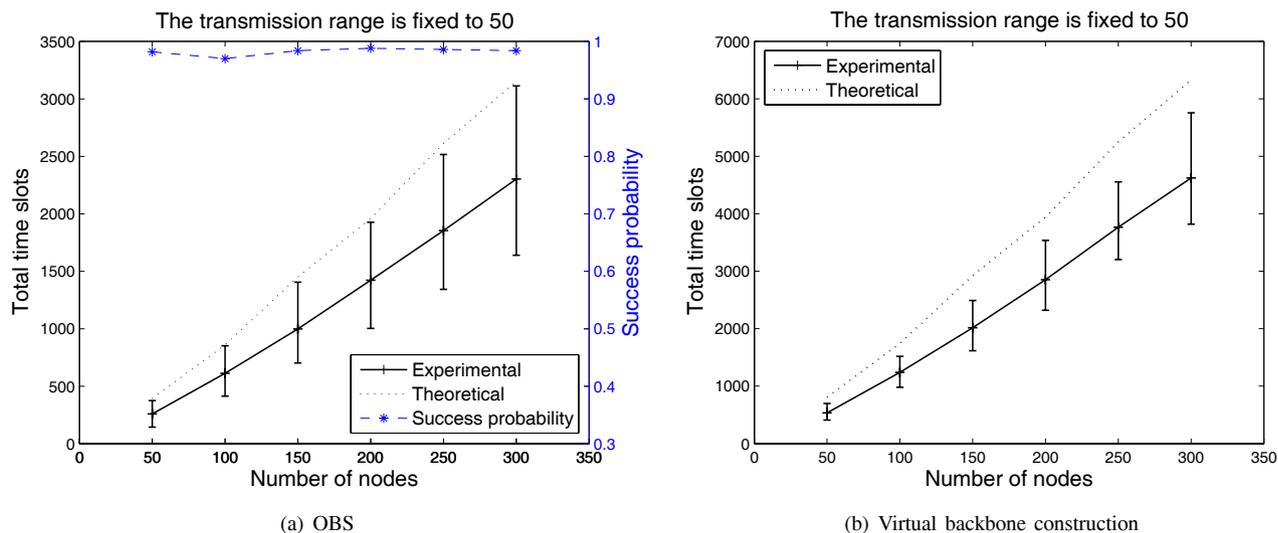


Fig. 5. Time vs. number of nodes.

their protocol design. Their MCDS algorithm guarantees an approximation ratio of  $3H(\Delta)$ , where  $H$  is the harmonic function. Guha and Khuller [14] designed centralized approximation algorithms for CDS having approximation ratio of order  $O(\log \Delta)$ . Wu and Li [24] designed a distributed CDS algorithm with approximation ratio  $O(n/2)$ , message complexity  $\Theta(n)$ , and time complexity  $O(\Delta^3)$ . Stojmenovic *et al* [20] presented a distributed construction of CDS with approximation ratio at least  $n/2$  and both the time and message complexity  $O(n)$ . Butenko *et al* [6] designed centralized and distributed CDS algorithms with proven bounds on message and time complexity.

Alzoubi, Wan, and Frieder [2]–[4] improved these CDS algorithms and further reduced those bounds on both time and message complexity. In [4], the approximation ratio is a constant 8, time complexity is  $O(n)$ , and message complexity is  $O(n \log n)$ . In [3], the authors studied distributed algorithms for MCDS in unit disk graphs and their algorithm is constant approximation and both the time and message complexity are  $O(n)$ . Min *et al* [15] employed Steiner tree techniques and

proposed an MIS algorithm with performance ratio 6.8. Wu *et al* [23] extended [24] to calculate a CDS based on each node's degree and energy level. Their objective is to balance the overall energy consumption in the network while generating a relatively small CDS. [10], [25] further improved these works.

Dubhashi *et al* [13] designed a  $O(\log \Delta)$ -approximation MCDS algorithm that has a *low stretch property* that any two adjacent nodes have their dominators at most  $O(\log n)$  apart. Acharya *et al* [1] also studied power-aware aspects of MCDS construction. They looked for alternative power-aware dominating sets once energy levels of certain nodes (of initial dominating set) fall below certain threshold. Cheng *et al* [8] proposed a Polynomial-Time Approximation Scheme (PTAS) for MCDS in unit disk graphs. They designed  $(1 + 1/s)$ -approximation algorithms with running time  $n^{O((s \log s)^2)}$ . Note that all of the above algorithms need global or quasi-global information such as a spanning tree or a known leader node.

## VII. CONCLUSION AND FUTURE WORK

In this work we designed an interference-aware virtual backbone construction algorithm with a proven bound. The results made many fundamental operations feasible in multi-hop wireless networks. We believe the  $O(\Delta \log n)$ -bound is the best we can achieve in this respect. We showed that our proposed algorithm can be applied to the leader election problem such that the modified algorithm is fully-distributed and interference-aware. As the future work, we believe we can find more application on certain class of problems.

There are other things we can work on as future work. First, our goals were achieved by taking advantages of MIS's geometrical properties. This may not be a must, although MIS-based CDS have many good properties such as the stretching factor. If we use other approaches to build a CDS, we may get different bounds as well as properties. Second, latency is our main interest in this work. We may consider other aspects such as saving energy, prolonging network lifetime, as well as reducing message complexity without sacrificing too much of latency.

## APPENDIX

Here in the appendix we provide the proofs for Theorems 1–8. We also introduce Lemmas 2–4, which guarantee the fact that all messages in Algorithms 3–5 are of constant size. In order to prove Theorem 1, we state the following fact first. We omit its proof because it can be calculated in a straightforward way using elementary Calculus.

*Fact 1:*  $\forall t, m \in \mathbb{R}$  such that  $m \geq 1$  and  $|t| \leq m$ ,

$$e^t \left(1 - \frac{t^2}{m}\right) \leq \left(1 + \frac{t}{m}\right)^m \leq e^t$$

Note that this holds even for negative values of  $t$ . Now we establish the following lemma to be used to prove Theorem 1.

*Lemma 1:* Let  $b \stackrel{\text{def}}{=} e(\ln(n\Delta + 1) + C)$ . If each node runs Alg. 1 simultaneously in parallel, then the probability that a node receives collision-free from a fixed neighbor is at least  $1 - e^{-b/e}$ .

(*Proof.*) A node receives from a neighbor collision-free if and only if this node is not transmitting, one neighbor is transmitting, and no other neighbors are transmitting in a time slot. Therefore,

$$\begin{aligned} & Pr[\text{a node receives from a neighbor in a fixed time slot}] \\ &= \left(1 - \frac{1}{\Delta}\right) \frac{1}{\Delta} \left(1 - \frac{1}{\Delta}\right)^d \geq \frac{1}{\Delta} \left(1 - \frac{1}{\Delta}\right) \left(1 - \frac{1}{\Delta}\right)^\Delta \\ &\geq \frac{1}{e\Delta/(1-1/\Delta)} \end{aligned}$$

where  $d < \Delta$  is the degree of the receiving node and the last inequality was obtained by applying Fact 1.

$$Pr[\text{a node doesn't receive from a specific neighbor in a fixed time slot}] \leq 1 - \frac{1}{e\Delta/(1-1/\Delta)}.$$

$$\begin{aligned} & Pr[\text{a node doesn't receive from a specific neighbor within } b\Delta/(1-1/\Delta) \text{ time slots}] \\ &\leq \left(1 - \frac{1}{e\Delta/(1-1/\Delta)}\right)^{b\Delta/(1-1/\Delta)} \\ &= \left(\left(1 + \frac{-1/e}{\Delta/(1-1/\Delta)}\right)^{\Delta/(1-1/\Delta)}\right)^b \\ &\leq (e^{-1/e})^b = e^{-b/e}. \end{aligned}$$

The last inequality was obtained by applying Fact 1 again.  $\square$

**Proof of Theorem 1:** According to the precondition,  $C > 0$ , we have

$$\begin{aligned} & Pr[\text{all nodes receive successfully from all neighbors}] \\ &= (Pr[\text{one node receives successfully from all neighbors}])^n \\ &\geq ((Pr[\text{one node receives successfully from one neighbor}])^\Delta)^n \\ &\geq \underbrace{(1 - e^{-b/e})^{n\Delta}}_{(\text{by Lemma 1})} = \left(1 - \frac{1}{e^{\ln(n\Delta+1)+C}}\right)^{n\Delta} \\ &= \left(1 + \frac{-e^{-C}}{n\Delta + 1}\right)^{n\Delta}. \end{aligned}$$

Plug in  $m = n\Delta + 1$ ,  $t = -e^{-C}$  and apply Fact 1 again, we get

$$\left(1 + \frac{-e^{-C}}{n\Delta + 1}\right)^{n\Delta+1} \geq e^{-e^{-C}} \left(1 - \frac{e^{-2C}}{n\Delta + 1}\right).$$

Since  $1 - \frac{e^{-2C}}{n\Delta+1} > 1 - \frac{e^{-C}}{n\Delta+1}$ , we can divide both sides by  $1 - \frac{e^{-C}}{n\Delta+1}$  and obtain

$$\begin{aligned} & \left(1 + \frac{-e^{-C}}{n\Delta + 1}\right)^{n\Delta} \\ &\geq e^{-e^{-C}} \left(1 - \frac{e^{-2C}}{n\Delta + 1}\right) / \left(1 - \frac{e^{-C}}{n\Delta + 1}\right) \\ &> e^{-e^{-C}}. \end{aligned} \quad (1)$$

$\square$

**Proof of Theorem 2:** By the geometrical properties of MIS, tessellation, and 12-coloring, we know the followings.

- 1) In each hexagon there can be at most one dominator.
- 2) The distance between any two different hexagons of the same color is at least two (Fig. 2(d)).
- 3) Any two dominators of the same color will not cause any collision at any time. (because, if collision happens, then there must be a point within the transmission range of these two dominators and the distance between these two dominators must be less than two).

These three observations guarantee us that, if we schedule each dominator to transmit according to its color, there will be no interference for the following reason. If there is interference at a dominatee  $v$ , caused by the transmission from dominators  $x$  and  $y$ , then  $dist(v, x) < 1$  and  $dist(v, y) < 1$  (in which  $dist(a, b)$  denotes the Euclidean distance between  $a$  and  $b$ ). Therefore,  $dist(x, y) < dist(v, x) + dist(v, y) < 2$ ,

violating property (2) above. Therefore each node receives successfully from all of its neighboring dominators.  $\square$

**Proof of Theorem 3:** Postcondition (1) can be proved in the following way. Without loss of generality we may assume  $ID(d_1) > ID(d_2)$ . If  $c_1 \neq c'_2$ , we will have a contradiction for the following reason. If  $ID(c_1) > ID(c'_2)$ , since  $c_1$  is a neighbor of  $d_1$  and a 2-hop neighbor of  $d_2$  via  $c_2$ ,  $c_1$  will definitely be in the  $V_C''$  from  $d_2$ 's point of view according to line 4 in Alg.3, and  $c'_2$  will never be chosen by  $d_2$  as the second designated connector. If  $ID(c'_2) > ID(c_1)$ , then  $c'_2$  is a neighbor of  $w$  and a 2-hop neighbor of  $d_2$  via  $c'_1$ . From  $d_1$ 's point of view,  $c_1$  will not be chosen as the first designated connector. Therefore  $c_1 = c'_2$ . We can prove  $c_2 = c'_1$  from a similar argument.

Postcondition (2) and (3) can be proved trivially by making the following observations.

- 1) By  $Time = T_s + 12$ , each dominatee has already gathered information about all of its neighboring dominators according to Alg. 3 (line 2) and Theorem 2.
- 2) By  $Time = T_s + 12 + T_{OBS}$ , each dominatee has already gathered information (with probability  $e^{-e^{-C}}$ ) about all dominators within 2 hops according to Alg. 4 (line 3) and Theorem 1.
- 3) By  $Time = T_s + 12 + 2T_{OBS}$ , each dominator has already gathered information (with probability  $e^{-2e^{-C}}$ ) about all dominators within 3 hops according to Alg. 4 (line 4) and Theorem 1.
- 4) Finally each dominator releases all connector information to its neighboring connectors.

Now we prove postcondition (4). Let  $J$  be defined as follows.  $J = (DOM, E_{DOM}^3)$  where  $(u, v) \in E_{DOM}^3$  if and only if  $u, v$  are 3-hop neighbors in the network. Our objective is to prove that  $J$  is connected. Assume the contrary that  $J$  is not connected, then there must be a connected component  $W \subsetneq DOM$  such that  $W$  is at least 4 hops from any other node in  $DOM - W$ . It follows immediately that we can pick up a node independent of both  $DOM - W$  and  $W$ , and therefore  $DOM$  is not maximal, contradicting to the maximality of  $DOM$  (since it is a maximal independent set).  $\square$

**Proof of Theorem 4:** The execution time of Alg. 3 and Alg. 3 is clearly at most  $24 + T_{OBS} = O(\Delta \log n)$   $\square$

**Proof of Theorem 5:** We will use the three observations of  $C_{12}$  in the proof of Theorem 2. For  $C_{37}$ , we also make the following similar observations.

- 1) In  $C_{37}$ , the distance between any two dominators of the same color is more than 4.
- 2) If two dominators have the same color, then at any time there will be no collision between their first designated connectors.

In Alg. 5, clearly any two dominators will not cause interference due to Theorem 2. Now we claim any two connectors will not cause interference either. First, we observe that all first or sole designated connectors are all schedule before  $T_s + 12 + 37 * 40 = T_s + 1492$  and all

second designated connectors are scheduled later than that, so first or sole designated connectors will never interfere with second designated connectors. Second, if  $c_1, c_2$  are first or sole designated connectors, they will not interfere with each other for the following reason. Suppose  $c_1, c_2$  are scheduled to transmit in time slots  $T_s + 12 + (C_{37}(y_1) - 1) * 40 + j_1$ , and  $T_s + 12 + (C_{37}(y_2) - 1) * 40 + j_2$  (for some  $1 \leq j_1, j_2 \leq 40$ ) respectively. If they interfere with each other, then  $T_s + 12 + (C_{37}(y_1) - 1) * 40 + j_1 = T_s + 12 + (C_{37}(y_2) - 1) * 40 + j_2$ . The only possibility to make this happen is when  $C_{37}(y_1) = C_{37}(y_2)$  and  $j_1 = j_2$ . It follows immediately that  $dist(y_1, y_2) < 4$  and this contradicts with our observation (1) for  $C_{37}$  coloring. Finally, we show that if  $d_p$  is a dominator, then for each dominator  $d_q$  within 3 hops, it will receive successfully from the second designated connector  $c''_{qp}$  for  $d_q$  and  $d_p$ . Note that we are NOT trying to prove that second dominators do not interfere with each other. We just show that they do not interfere *at any targeted dominators*. In other words, they may be interfering somewhere else, but that does not matter. To prove this, suppose  $d_r, d_s$  are another pair of dominators separated by 3 hops,  $c''_{rs}$  is the second designated connector for  $d_r$  and  $d_s$ , and  $c''_{rs}$  interfere with  $c''_{qp}$  at  $d_p$  at some time. According to Alg. 5,  $c''_{qp}$  and  $c''_{rs}$  are scheduled to transmit at  $T_s + 1492 + (C_{12}(d_p) - 1) * 40 + k_1$  and  $T_s + 1492 + (C_{12}(d_s) - 1) * 40 + k_2$  for some  $1 \leq k_1, k_2 \leq 40$ . Since  $T_s + 1492 + (C_{12}(d_p) - 1) * 40 + k_1 = T_s + 1492 + (C_{12}(d_s) - 1) * 40 + k_2$ , we have  $(C_{12}(d_p) - C_{12}(d_s)) * 40 = k_2 - k_1$ . If  $C_{12}(d_p) = C_{12}(d_s)$  then  $d_p$  and  $d_s$  must be separated by at least distance 2, which contradicts the assumption that  $c''_{rs}$  is adjacent to both of them. If  $C_{12}(d_p) \neq C_{12}(d_s)$ , then  $|k_2 - k_1| > 40$ , which is impossible. Therefore, we have proved this theorem.  $\square$

**Proof of Theorem 6:** According to line 7, clearly this algorithm terminates at  $T_s + 1492 + 12 * 40 =$ . Therefore, its execution time is 1972.  $\square$

**Proof of Theorem 7:** Since we run [9] in a way that each transmission slot of the dominator is replaced by an URS (Alg. 5) execution, which guarantees the reception of all neighbors according to Theorem 5. Therefore, the postcondition holds.  $\square$

**Proof of Theorem 8:** This follows directly from the followings. (1) The execution time of [16] is  $O(\log^2 n)$ . (2) The execution time of [9] is  $O(n \log n)$ . (3) We replace each time slot in [9] by 1972 time slots, which increases the execution of [9] only by a linear factor.  $\square$

The following lemmas are presented to guarantee the messages in Algorithms3–5 are of constant size.

*Lemma 2:* A unit disk can contain at most 5 points such that the distance between any two of them is (strictly) greater than one.

(Proof.) Given a unit disk, let  $a$  be its center and suppose there are 6 points in this disk. First we know that none of these 6 points can be the center  $a$ , since any points in the disk is within a distance at most one from  $a$ . Second, for

any two points  $x, y$  of the 6 points, the angle  $\angle xay$  is more than  $60^\circ$  (since  $|x - y| > 1$ ). Now, take  $a$  as the origin and fix an arbitrary direction as the  $x$ -axis. We consider the polar coordinates of these 6 points and sort them according to their angles w.r.t.  $a$  and the  $x$ -axis. We name them  $x_1, x_2, \dots, x_6$  according to this order. Now we sum up the angles  $\angle x_1ax_2 + \angle x_2ax_3 + \dots + \angle x_5ax_6 + \angle x_6ax_1$ , which will be strictly greater than  $6 * 60^\circ = 360^\circ$ . Therefore, we get a contradiction the this lemma is proved.  $\square$

We state the following theorem from [21] and we'll use it to prove Lemma 3.

**Theorem 9:** (Wegner Theorem) The area of the convex hull of any  $n \geq 2$  non-overlapping unit-radius circular disks is at least  $2\sqrt{3}(n-1) + (2-\sqrt{3})[\sqrt{12n-3}-3] + \pi$ .

**Lemma 3:** There can be at most 21 black nodes within any disk of radius 2.

(Proof.) Fix a disk  $D_2$  centered at a point  $u$ . Let  $S$  denote the set of black nodes in  $D_2$ . We know that black nodes are mutually independent. If, for each node in  $S$ , we consider a disk of radius 1/2 centered at this node, then all of those disks must be disjoint. Therefore, the convex hull of  $S$  must be contained in the disk of radius 2.5 centered at  $u$ . By applying Wegner Theorem with proper scaling, we have

$$2\sqrt{3}(|S|-1) + (2-\sqrt{3})[\sqrt{12|S|-3}-3] + \pi < 25\pi.$$

Straightforward calculation shows that the maximum integer to make the above expression hold is  $|S| = 21$ .  $\square$

Similarly, we state the following lemma without proof.

**Lemma 4:** There can be at most 41 black nodes within any disk of radius 3.

## REFERENCES

- [1] T. Acharya, "Distributed algorithm for power aware minimum connected dominating set for routing in wireless ad hoc network," in *Proc. International Conf. Parallel Processing Workshops (ICPPW)*, 2005, pp. 387-394.
- [2] K. Alzoubi, P.-J. Wan, and O. Frieder, "Distributed heuristics for connected dominating set in wireless ad hoc networks," *KICS J. Commun. Networks*, vol. 4, no. 1, 2002.
- [3] K. Alzoubi, P.-J. Wan, and O. Frieder, "Message-optimal connected dominating sets in mobile ad hoc networks," in *Proc. 3rd ACM International Symposium Mobile Ad Hoc Networking Computing*, 2002, pp. 157-164.
- [4] K. Alzoubi, P.-J. Wan, and O. Frieder, "New distributed algorithm for connected dominating set in wireless ad hoc networks," in *Proc. 35th Annual Hawaii International Conf. System Sciences (HICSS)*, 2002, vol. 9, p. 297.
- [5] V. Bharghavan and B. Das, "Routing in ad hoc networks using minimum connected dominating sets," in *Proc. IEEE International Conf. Commun.*, 1997.
- [6] S. Butenko, X. Cheng, C. A. Oliveira, and P. M. Pardalos, "A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks," *Cooperative Control Optimization*, pp. 61-73, 2004.
- [7] X. Cheng, M. Ding, D. H. Du, and X. Jia, "Virtual backbone construction in multihop ad hoc wireless networks: research articles," *Wireless Commun. Mobile Computing*, vol. 6, no. 2, pp. 183-190, 2006.
- [8] X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du, "A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks," *Networks*, vol. 42, no. 4, pp. 202-208, 2003.
- [9] I. Cidon and O. Mokryn, "Propagation and leader election in a multihop broadcast environment," in *Proc. 12th International Symposium Distributed Computing*, 1998, pp. 104-118.
- [10] F. Dai and J. Wu, "An extended localized algorithm for connected dominating set formation in ad hoc wireless networks," *IEEE Trans. Parallel Distributed Syst.*, vol. 15, no. 10, pp. 908-920, 2004.

- [11] B. Das and V. Bharghavan, "Routing in ad-hoc networks using minimum connected dominating sets," in *Proc. IEEE International Conf. Commun.*, 1997, pp. 376-380.
- [12] B. Das, R. Sivakumar, and V. Bharghavan, "Routing in ad hoc networks using a spine," in *Proc. IEEE International Conf. Computers Commun. Networks*, 1997.
- [13] D. Dubhashi, A. Mei, A. Panconesi, J. Radhakrishnan, and A. Srinivasan, "Fast distributed algorithms for (weakly) connected dominating sets and linear-size skeletons," *J. Computer System Sciences*, vol. 71, no. 4, pp. 467-479, 2005.
- [14] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374-387, 1998.
- [15] M. Min, H. Du, X. Jia, C. X. Huang, S. C.-H. Huang, and W. Wu, "Improving construction for connected dominating set with steiner tree in wireless sensor networks," *J. Global Optimization*, vol. 35, no. 1, pp. 111-119, 2006.
- [16] T. Moscibroda and R. Wattenhofer, "Maximal independent sets in radio networks," in *Proc. PODC*, 2005, pp. 148-157.
- [17] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proc. 5th ACM/IEEE International Conf. Mobile Computing Networking (MobiCom)*, 1999, pp. 151-162.
- [18] R. Sivakumar, B. Das, and V. Bharghavan, "An improved spine-based infrastructure for routing in ad hoc networks," in *Proc. IEEE Symposium Computers Commun.*, 1998.
- [19] R. Sivakumar, B. Das, and V. Bharghavan, "Spine-based routing in ad hoc networks," *ACM/Baltzer Cluster Computing J.*, vol. 1, pp. 237-248, 1998.
- [20] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks," in *Proc. 34th Hawaii International Conf. System Sciences*, 2001.
- [21] G. Wegner, "Über endliche Kreispackungen in der Ebene," *Studia Scientiarum Mathematicarum Hungarica*, vol. 21, pp. 1-28, 1986.
- [22] D. B. West, *Introduction to Graph Theory*, 2nd edition. Upper Saddle River, NJ: Prentice Hall, 2001.
- [23] J. Wu, M. Gao, and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks," in *Proc. International Conf. Parallel Processing*, 2001, p. 346.
- [24] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proc. 3rd International Workshop Discrete Algorithms Methods Mobile Computing Commun.*, 1999, pp. 7-14.
- [25] W. Wu, H. Du, X. Jia, Y. Li, and S. C.-H. Huang, "Minimum connected dominating sets and maximal independent sets in unit disk graphs," *Theoretical Computer Science*, vol. 352, no. 1, pp. 1-7, 2006.
- [26] H. Zhai, X. Chen, and Y. Fang, "Improving transport layer performance in multihop ad hoc networks by exploiting MAC layer information," *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1692-1701, 2007.
- [27] H. Zhai, J. Wang, X. Chen, and Y. Fang, "Medium access control in mobile ad hoc networks: challenges and solutions," *Wireless Commun. Mobile Computing*, vol. 6, no. 2, pp. 151-170, 2006.



**Scott Huang** received his B.S. degree from National Taiwan University, and his Ph.D. degree from the University of Minnesota, Twin Cities. He is currently an assistant professor in the Department of Electrical Engineering, National Tsing Hua University. His research interests include wireless networking, security, and communication theory, as well as combinatorial optimization.



**Min-Te Sun** (S'99-M'02) received his B.S. degree in mathematics from National Taiwan University, Taipei, Taiwan, R.O.C. in 1991; the M.S. degree in computer science from Indiana University, Bloomington, in 1995; and the Ph.D. degree in computer and information science from the Ohio State University, Columbus, in 2002. Since 2008, he has been with Department of Computer Science and Information Engineering at National Central University, Taiwan, R.O.C. His research interests include distributed algorithm design, wireless communications,

and mobile computing.

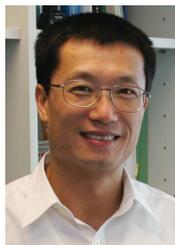


**Qilian Liang** is an Associate Professor at the Department of Electrical Engineering, University of Texas at Arlington. He received the B.S. degree from Wuhan University in 1993, the M.S. degree from Beijing University of Posts and Telecommunications in 1996, and the Ph.D. degree from the University of Southern California (USC) in May 2000, all in electrical engineering. Prior to joining the faculty of the University of Texas at Arlington in August 2002, he was a Member of Technical Staff for Hughes Network Systems Inc. in San Diego, California. His

research interests include radar sensor networks, wireless sensor networks, wireless communications, wireless networks, communication system and communication theory, signal processing for communications, fuzzy logic systems and applications, collaborative and distributed signal processing, etc. Dr. Liang has published more than 160 journal and conference papers, 7 book chapters, and has 6 U.S. patents pending. He received the 2002 IEEE TRANSACTIONS ON FUZZY SYSTEMS Outstanding Paper Award, the 2003 U.S. Office of Naval Research (ONR) Young Investigator Award, the 2005 UTA College of Engineering Outstanding Young Faculty Award, and the 2007, 2009, and 2010 U.S. Air Force Summer Faculty Fellowship Program Award.



**Peng-Jun Wan** received the B.S. degree from Tsinghua University, the M.S. degree from the Chinese Academy of Science, and the Ph.D. degree from the University of Minnesota. He is currently a professor of computer science in the Department of Computer Science, Illinois Institute of Technology, Chicago. His research interests include wireless networks, and algorithm design and analysis.



**Xiaohua Jia** received his BSc (1984) and MEng (1987) from the University of Science and Technology of China, and the DSc (1991) in information science from the University of Tokyo, Japan. He is currently a Chair Professor in the Department of Computer Science at the City University of Hong Kong. Prof. Jia is an editor of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS (06–09), *Wireless Networks*, the *Journal of the World Wide Web*, the *Journal of Combinatorial Optimization*, etc. His research interests include

distributed systems, computer networks, wireless sensor and mobile ad hoc networks.