Available online at www.sciencedirect.com
SCIENCE DIRECT®
J. Parallel Distrib. Comput. 66 (2006) 947–958
Journal of Parallel and Distributed Computing
www.elsevier.com/locate/jpdc

On constructing k-connected k-dominating set in wireless ad hoc and sensor networks☆

Fei Dai[a,*], Jie Wu[b]

[a]Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND 58105, USA
[b]Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431, USA

Received 23 May 2005; received in revised form 23 September 2005; accepted 1 December 2005
Available online 12 May 2006

Abstract

An important problem in wireless ad hoc and sensor networks is to select a few nodes to form a virtual backbone that supports routing and other tasks such as area monitoring. Previous work in this area has focused on selecting a small virtual backbone for high efficiency. In this paper, we propose the construction of a k-connected k-dominating set (k-CDS) as a backbone to balance efficiency and fault tolerance. Four localized k-CDS construction protocols are proposed. The first protocol randomly selects virtual backbone nodes with a given probability p_k, where p_k depends on the value of k and network condition, such as network size and node density. The second one maintains a fixed backbone node degree of B_k, where B_k also depends on the network condition. The third protocol is a deterministic approach. It extends Wu and Dai's coverage condition, which is originally designed for 1-CDS construction, to ensure the formation of a k-CDS. The last protocol is a hybrid of probabilistic and deterministic approaches. It provides a generic framework that can convert many existing CDS algorithms into k-CDS algorithms. These protocols are evaluated via a simulation study.

Keywords: Connected dominating set (CDS); k-Vertex connectivity; Localized algorithms; Simulation; Wireless ad hoc and sensor networks

1. Introduction

In wireless ad hoc and sensor networks (or simply wireless networks), autonomous nodes form self-organized networks without centralized control or infrastructure. These networks can be modelled as unit disk graphs [10], where two nodes are neighbors if they are within each other's transmission range. To support various network functions such as multi-hop communication and area monitoring, some wireless nodes are selected to form a *virtual backbone*.

In many existing schemes [1,2,4,9,11,12,17,29,31,33,35], virtual backbone nodes form a connected dominating set (CDS) of the wireless network. A set of nodes is a dominating set if all nodes in the network are either in this set or have a neighbor in this set. A dominating set is a CDS if the subgraph induced from this dominating set is connected. For example, both node sets {8} in Fig. 1(a) and {5, 6, 7, 8} in Fig. 1(b) are CDSs in their corresponding networks. Applications of a CDS in wireless networks include:

- *Reducing routing overhead* [35]. By removing all links between non-backbone nodes, the size and maintenance cost of routing tables can be reduced. By using only backbone nodes to forward broadcast packets, the excessive broadcast redundancy can be avoided.
- *Energy-efficient routing* [9]. By putting non-backbone nodes into periodical sleep mode, the energy consumption is greatly reduced while network connectivity is still maintained by backbone nodes.
- *Area coverage* [8]. In densely deployed sensor networks, the node coverage of a CDS is a good approximation of area coverage. That is, the deployment area is within the sensing range of backbone nodes with high probability.

Previous studies in this area has focused on finding a minimal CDS for higher efficiency. However, recent studies

☆This work was supported in part by NSF Grants CCR 0329741, CNS 0434533, CNS 0422762, and EIA 0130806.
* Corresponding author.
E-mail addresses: Fei.Dai@ndsu.edu (F. Dai), jie@cse.fau.edu (J. Wu)
URLs: http://www.ece.ndsu.nodak.edu/~fdai
http://www.cse.fau.edu/~jie.

0743-7315/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2005.12.010

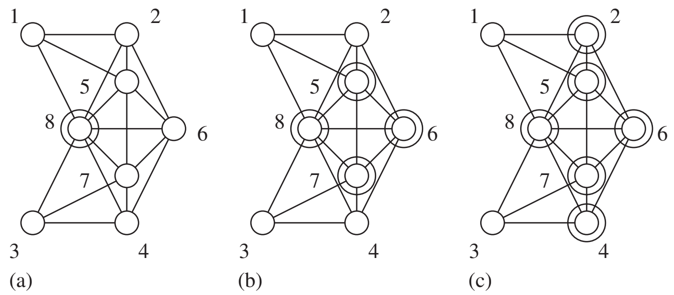

Fig. 1. k-Connected k-dominating sets constructed by applying k-coverage conditions with $k = 1, 2$, and 3. Virtual backbone nodes are represented by double circles.

[3,6,18,22,23] suggested that it is equally important to maintain a certain degree of redundancy in the virtual backbone for fault tolerance and routing flexibility. In wireless ad hoc networks, a node may fail due to accidental damage or energy depletion; a wireless link may fade away during node movement. In a wireless sensor network, it is desirable to have several sensors monitor the same target and to let each sensor report via different routes to avoid losing an important event.

We propose to construct a k-connected k-dominating set (or simply k-CDS) as a backbone of wireless networks. A node set is k-dominating if every node is either in the set or has k neighbors in the set. A k-dominating set is a k-CDS if its induced subgraph is k-vertex connected. A graph is k-vertex connected if removing any $k - 1$ nodes from it does not cause a partition. For example, backbone nodes 5, 6, 7, and 8 in Fig. 1(b) form a 2-CDS. Every non-backbone node has at least two neighboring backbone nodes, and the subgraph consisting of all backbone nodes is 2-vertex connected. Similarly, node set {2, 4, 5, 6, 7, 8} in Fig. 1(c) is a 3-CDS. When any $k - 1$ nodes are removed from a k-CDS, the remaining nodes still form a CDS (i.e., 1-CDS). Therefore, a k-CDS as a virtual backbone can survive failures of at least $k - 1$ nodes.

Four k-CDS construction protocols are proposed in this paper. All those protocols are *localized algorithms* that rely only on neighborhood information. In dynamic wireless networks, a localized algorithm has many desirable properties such as low cost and fast convergency. The first two protocols, called k-Gossip and k-Grid, respectively, are probabilistic schemes that construct a k-CDS with high probability. k-Gossip is a simple extension of an exiting probabilistic algorithm [17], where each node becomes a backbone node with a given probability p_k. This algorithm has very low overhead, but the global parameter p_k that maintains a k-CDS with high probability depends on network size and density. In addition, the randomized backbone node selection process usually produces a large backbone. The k-Grid is inspired by two probabilistic topology control schemes [7,26]; it reduces the k-CDS size via selecting B_k backbone nodes within the neighborhood of each node. This protocol incurs a slight overhead of neighborhood density estimation. Again, the global parameter B_k depends on network size. The third protocol extends our early deterministic CDS algorithm [33], where each node has a backbone status by default and becomes a non-backbone node if a *coverage condition* is satisfied. The proposed k-coverage condition guarantees that all backbone nodes form a k-CDS but has relatively high computation overhead. We further introduce a hybrid paradigm to extend many existing CDS algorithms for k-CDS formation. In this scheme, a wireless network is randomly partitioned into k subgraphs consisting of nodes with different colors (the probabilistic part). A colored virtual backbone is constructed for each subgraph using a traditional CDS algorithm (the deterministic part). We prove that in dense wireless networks, the union of all colored backbones is a k-CDS with high probability. Simulation study is conducted to compare performances of these protocols.

The remainder of this paper is organized as follows. Section 2 reviews existing virtual backbone construction protocols, including both probabilistic and deterministic schemes, and introduces the concept of k-CDS. In Section 3, we propose extensions of three virtual backbone protocols for k-CDS construction. Section 4 presents the color-based k-CDS formation paradigm. Section 5 gives simulation results, and Section 6 concludes this paper.

2. Background and related work

In this section, we first introduce three existing localized virtual backbone formation algorithms, two probabilistic and one deterministic, that will be extended for k-CDS construction in the next section. Then we review concepts of k-connectivity and k-CDS, and algorithms that verify k-connectivity and form a k-CDS.

2.1. Virtual backbone construction

A wireless network is usually modelled as a unit disk graph [10] $G = (V, E)$, where V is the set of wireless nodes and E the set of wireless links. Each node in V is associated with a coordination in 2-D or 3-D Euclidean space. A wireless link $(u, v) \in E$ if and only if the Euclidean distance between nodes u and v is smaller than a uniform transmission range R. In real wireless networks, the transmission range of each node may not be a perfect disk. In this case, the network is a quasi-unit disk graph [20], where a bidirectional link (u, v) definitely exists if the distance between u and v is less than a certain value $d < R$, and may or may not exist when the distance is larger than d but smaller than R.

Many schemes have been proposed to construct a CDS as a virtual backbone to support routing activities [1,2,4,9,11,12,17,29,31,33,35] or maintain target coverage [27] in wireless networks. A set $V' \subseteq V$ is a CDS of network G, if all nodes in $V - V'$ are neighbors of (i.e., dominated by) a node in V' and, in addition, the subgraph $G[V']$ induced from V' is connected. The problem of finding a minimum CDS is NP-complete. Centralized [12] and cluster-based [2,4] CDS algorithms provide hard performance guarantees (i.e., upper bounds on CDS size) in wireless networks. However, those schemes require either global information or global coordination, which limit their applications to static or almost static networks [5]. In dynamic networks, most existing CDS formation algorithms are *localized*; that is, the status of each node, backbone or non-backbone, depends on its h-hop neighborhood information only with a small h. By eliminating those long distance information propagations in centralized or cluster-based schemes, a localized algorithm can achieve fast convergence ($O(1)$ rounds) with low maintenance cost ($O(1)$ messages per node).

Localized CDS algorithms are either *probabilistic* or *deterministic*. A probabilistic scheme incurs very low overhead and maintains a CDS with a high probability. A typical probabilistic scheme is the gossip-based algorithm [17,15].

Gossip [17]: Each node has a backbone status with probability p.

The selection of backbone nodes in Gossip is purely random without using any neighborhood information. Simulation results show that when p is larger than a threshold, these backbone nodes form a CDS with very high probability. This threshold depends on network size and density and is determined based on experimental data. To maintain high success ratio (i.e, the probability of constructing a CDS) under unpredictable network conditions, the selection of p is usually conservative, which produces a large backbone.

In wireless networks with a non-uniform node distribution, grid-based [7,26] algorithms can be used to control backbone node density. These schemes are originally proposed as topology control schemes, but can be modified for virtual backbone construction. The basic idea is that if every node has B backbone neighbors, then all backbone nodes form a CDS with high probability. The value of B is also determined based on experimental data.

Grid: Each node has at least B neighboring backbone neighbors.

Deterministic algorithms [1,9,11,29,31,35] guarantee a CDS in connected networks. They usually select fewer backbone nodes than probabilistic schemes, because their selections are "smarter" using 2-hop neighborhood information (or simply 2-hop information). For each node v, its 2-hop information consists of its neighbor set $N(v)$ and neighbor sets $N(u)$ of all neighbors $u \in N(v)$, and is collected via 2 rounds of "Hello" exchanges among neighbors. The *complete 2-hop information* of v is a subgraph of G, including v's entire 2-hop neighbor set, and all adjacent links of v's 1-hop neighbors. Some algorithms use v's *restricted 2-hop information*, which is the subgraph $G[N(v)]$ induced from v's 1-hop neighbor set. One reason to use restricted 2-hop information is that, in quasi-unit disk graphs, a bidirectional link (u, w) between a 1-hop neighbor u and a 2-hop neighbor w cannot be confirmed via 2 rounds of "Hello" exchanges. Another reason is that applying a localized algorithm on a smaller subgraph can reduce the computation cost.

In [35], Wu and Li proposed a deterministic CDS algorithm called marking process and two backbone node pruning rules called Rules 1 and 2, which were later replaced by an enhanced rule called Rule k [11]. Stojmenovic et al. [31] reduced the message cost of the marking process using position information. Chen et al. [9] designed a backbone formation protocol called Span, which is similar

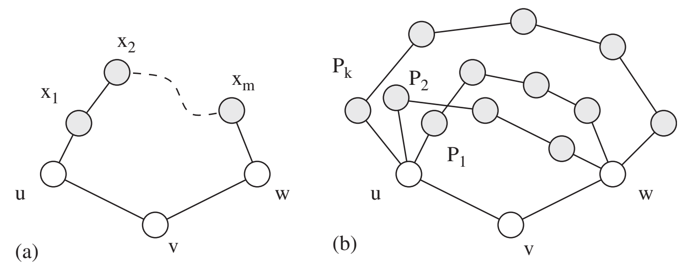

Fig. 2. Replacement paths between two neighbors u and w of node v. Gray nodes have higher priorities than that of v.

to the combination of the marking process and Rules 1 and 2. Qayyum et al. [29] provided another backbone formation scheme called MPR, and Adjih et al. [1] enhanced this scheme to construct a smaller CDS. Wu and Dai [33] showed that all above algorithms are special cases of the following coverage condition.

Coverage condition [33]: Node v has a non-backbone status if for any two neighbors u and w, a *replacement path* exists that connects u and w via several intermediate nodes (if any) with higher priorities than v.

When applying the coverage condition, each node tries to find a replacement path between every pair of its neighbors. Fig. 2(a) shows a sample replacement path $(u, x_1, x_2, \ldots, x_m, w)$ that connects two neighbors of the current node v. Since node v knows only its 2-hop information, all intermediate nodes $x_1, x_2, \ldots, x_m$ are within 2 hops of v. In addition, all intermediate nodes must have a higher priority than node v. A priority is a unique attribute of a node, such as node ID or the combination of node degree (i.e., $|N(v)|$) and node ID. Node priorities establish a total order among nodes to avoid simultaneous withdrawals that may cause a partition in the virtual backbone. If every node pair of v's neighbors are connected via high priority nodes, then v can be safely removed from the backbone while the remaining nodes still form a CDS.

In Fig. 1(a), node 1 is a non-backbone node based on the coverage condition, because its neighbors 2, 5, and 8 are directly connected. Node 2 has two neighbors 1 and 6 that are not directly connected. However, nodes 1 and 6 are connected via a replacement path $(1, 5, 6)$. Here we assume node ID is used as priority, and node 5 has a higher priority than 2. Therefore, node 2 is a non-backbone node. Similarly, nodes 3, 4, 5, 6, and 7 are also non-backbone nodes. The resultant backbone, consisting of node 8 only, is a CDS of the network.

2.2. k-connectivity and k-domination

Many existing works [3,6,18,22,23] suggested to maintain k-vertex connectivity (or simply k-connectivity) in wireless networks for fault tolerance and/or high throughput.

Definition 1 (*k-vertex connectivity*). A network G is k-vertex connected if it is connected and removing any $1, 2, \ldots, k-1$ nodes from G will not cause partition in G.

An equivalent definition is that a network is k-vertex connected if any two nodes in the network are connected via k-node disjoint paths (Menger's theorem [28]). The network in Fig. 1 is 3-connected, since any two nodes are connected via three node disjoint paths. For example, nodes 1 and 3 are connected via node disjoint paths (1, 8, 3), (1, 5, 7, 3), and (1, 2, 6, 4, 3). Maximal flow (minimal cut) algorithms [14] are usually employed to discover all node disjoint paths between a pair of source/sink nodes. A general purpose maximum flow algorithm has a computation complexity of $O(|V||E|)$. If one only needs to verify whether there are k-node disjoint paths between two nodes, a variation of Edmonds and Karp's flow augmentation algorithm [13] can do the job in $O(k|E|)$ time. This is because each augmentation (i.e., the process of finding a new path) is a breadth-first search in G, which takes $O(|E|)$ time, and it takes at most k augmentations to find (or verify the non-existence of) k-node disjoint paths. The time complexity for verify the k-connectivity of a graph G is $O(k^2|V||E|)$.

Definition 2 (*k-connected k-dominating set*). A node set $V' \subseteq V$ is a k-dominating set (or simply k-DS) of G if every node not in V' has at least k neighboring nodes in V'. A k-DS is a k-connected k-dominating set (or simply k-CDS) of G if the subgraph $G[V']$ induced from V' is k-vertex connected.

The previous definition of CDS (also called 1-CDS) is a special case of k-CDS with $k = 1$. Several schemes [3,22,23] have been proposed to maintain the k-connectivity in topology control. Basu and Redi [6] designed a centralized algorithm for achieving 2-connectivity in wireless networks using mobile nodes. Jorgic et al. [18] suggested to use local k-connectivity to approximate global k-connectivity based on neighborhood information. The problems of constructing double dominating sets and k-dominating sets in general graphs have been studied in [16,24]. In [19], three heuristic algorithms are provided to construct a double dominating set. Localized double dominating set algorithms were discussed in [30]. The localized construction of a k-CDS has not been discussed.

3. *k*-Extensions of existing CDS algorithms

In this section, we extend both probabilistic and deterministic localized CDS algorithms (Gossip, Grid, and the Wu and Dai's coverage condition) to construct k-CDS in wireless networks, and show limits of these extensions. In the next section, we will introduce a new approach, color-based coverage condition (CBCC), to overcome those limits. These three localized K-CDS algorithms are compared in Table 1.

3.1. Probabilistic approach

The gossip-based algorithm can be easily extended to construct a k-CDS with high probability. The extended rule for selecting backbone nodes is as follows:

Table 1
Comparison of k-CDS algorithms

Algorithm	Guarantees k-CDS	Backbone size (expected)	Comm. rounds	Message size	Computation cost
k-Gossip	No	np_k	0	N/A	$O(1)$
k-Grid	No	$O(B_k A)$	1	$O(1)$	$O(\Delta)$
k-Coverage	Yes	$O(k) \cdot OPT$	2	$O(\Delta)$	$O(k\Delta^4)$
CBCC	No	$O(1) \cdot OPT$	2	$O(\Delta)$	$O(\Delta^3)$

k-Gossip: Each node has a backbone status with probability p_k.

Note that the above rule is almost identical to its 1-CDS version. The difference is that the probability p_k that any node becomes a backbone node is now a function of k. In k-Gossip, the perfect value of p_k, which constructs a small virtual backbone while maintaining a k-CDS with high probability, depends not only on k, but also on total node number n, deploy area A, and transmission range R. Some analytical study has provided an upper bound of p_k that almost always achieves k coverage in a network [21]. However, these upper bounds are conservative estimations of the perfect p_k, which usually need adjustments based on experimental data. Fig. 3(a) shows our experiment results in a sample network, where 200 nodes with transmission range 250 m are randomly deployed in a 1000 m × 1000 m region. For each k, there exists a p_k that almost always (i.e., with a probability very close to 1) selects a k-CDS. For example, when $k = 2$, using $p_k = 48\%$ constructs a 2-CDS with probability 91.0%. When $k = 3$, using $p_k = 62\%$ achieves a success ratio of 92.4%.

As in its 1-CDS counterpart, k-Gossip incurs very low overhead at each node. It requires no information exchange among neighbors and very low ($O(1)$) computation cost. Therefore, the backbone construction process completes almost instantaneously. The major drawback is that it requires some global information, such as network size and density, to be effective. The expected number of backbone nodes in k-Gossip is np_k. If different values of p_k are used under different circumstances, global network information, such as node number n and deployment area A, must be collected and broadcast to each node. If the above global information is unknown and a single p_k is used for different network situations, the selection of p_k must be very conservative to maintain a k-CDS in the worst case scenario, which yields a larger backbone size of $O(n)$.

Similarly, we can extend the grid-based rule as follows:

k-Grid: Each node has at least B_k backbone neighbors.

In k-Grid, the ideal number of neighbors B_k depends less on network size or density, and adapts better to a non-uniform node distribution. Since the average density of backbone nodes is $O(B_k)$, the resultant backbone size is expected to be $O(B_k A)$. The major problem in the k-Grid implementation is how to control the density of backbone nodes. The traditional clustering approach [25] involves node status information propagation, and suffers from a long expected delay $O(\log n)$. We propose the following pure localized algorithm to approximate the k-Grid rule.

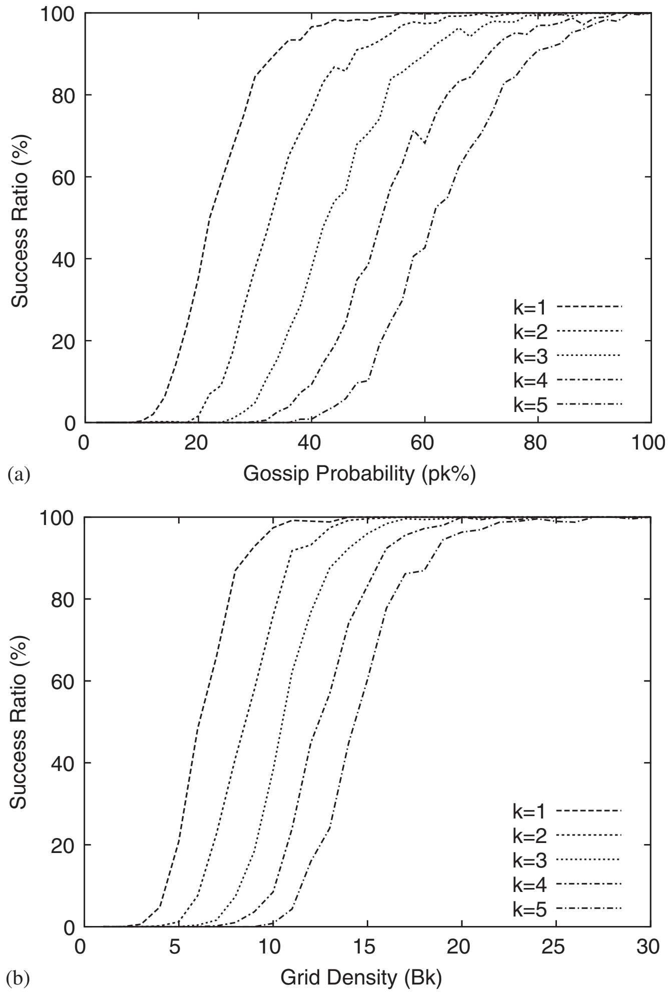

Fig. 3. Success ratio of probabilistic k-CDS protocols in random networks with 200 nodes deployed in a 1000×1000 area. The transmission range is 250: (a) k-Gossip with varying p_k, (b) k-Grid with varying B_k.

Randomized k-Grid

(1) Each node v computes its degree $\delta(v) = |N(v)|$ via exchanging "Hello" messages among neighbors.
(2) Each node v has a backbone status with probability $\min\{1, \frac{B_k}{\delta(v)+1}\}$.

In the above algorithm, the node density in the neighborhood is estimated as $\delta(v) + 1$. Although a more accurate estimation can be computed from degrees of all neighbors $(\delta(u) : u \in N(v))$, such an enhancement requires 2-hop information and doubles the message overhead. To compute node degree, each node sends a "Hello" message with $O(1)$ size. The corresponding computing cost is $O(\Delta)$ per node, where Δ is the maximal node degree. Fig. 3(b) shows values of B_k that construct a k-CDS with high probability. Especially, using $B_k = 11$ constructs a 2-CDS with probability 91.8%; using $B_k = 14$ constructs a 3-CDS with probability 92.4%.

3.2. Deterministic approach

The original coverage condition [33] that constructs a 1-CDS can be extended as follows to construct a k-CDS.

k-Coverage condition: Node v has a non-backbone status if for any two neighbors u and w, *k node disjoint replacement paths* exist that connect u and w via several intermediate nodes (if any) with higher IDs than v.

In the original coverage condition, a node can be removed from a CDS if all its neighbors are inter-connected via a replacement path. In the k-coverage condition, the criterion is more strict: if a node is to be removed from a k-CDS, all its neighbors must be k-connected with each other via higher priority nodes. This criterion is shown by Fig. 2(b), where two neighbors u and w of the current node v are connected via node disjoint paths $P_1, P_2, \ldots, P_k$ consisting of high priority (gray) nodes. The following theorem shows that k-coverage condition guarantees a k-CDS in a k-connected network.

Lemma 1. *A node set V' is a k-CDS of network G if after removing any $k-1$ nodes from V', the remaining part of V' is a CDS of the remaining part of G.*

Proof. First, V' is a k-dominating set of G. Because otherwise, there exists a node v in G with less than k neighbors in V'. After removing all those neighbors from V', node v is no longer dominated by V', which contradicts the assumption that the remainder of V' dominates the remainder of G. Second, $G[V']$ is still connected after removing any $k-1$ nodes; that is, V' is k-connected. □

Theorem 1. *If the k-coverage condition is applied to a k-connected network G, the resultant virtual backbone V' forms a k-CDS of G.*

Proof. Let V be the set of all nodes and X be the set of any $k-1$ nodes from V'. Since G is k-connected, its subgraph G' induced from $V - X$ is also connected. Let v be any non-backbone node in $V - V'$. Based on the k-coverage condition, any two neighbors u and w of v are connected via k-node disjoint replacement paths. After removing $k-1$ nodes from G, u and w are still connected via at least one replacement path in G'. Since all non-backbone nodes in G' satisfy the original coverage condition, the remaining nodes in $V - V'$ form a CDS of G' [33]. From Lemma 1, V' is a k-CDS of G. □

When $k = 1$, the k-coverage condition is equivalent to the original coverage condition. Fig. 1(b) shows a 2-CDS constructed by the k-coverage condition with $k = 2$. Here node 5 becomes a backbone node, because two of its neighbors, nodes 1 and 6, are connected by only one replacement path. On the other hand, nodes 1, 2, 3, and 4 are non-backbone nodes, because all their neighbors are connected via 2-node disjoint replacement paths. The resultant virtual backbone, containing nodes 5, 6, 7, and 8, is a 2-CDS of the network. Similarly, nodes 2, 4, 5, 6, 7, and 8 in Fig. 1(c) are selected as backbone nodes when $k = 3$. Here we assume each node uses complete

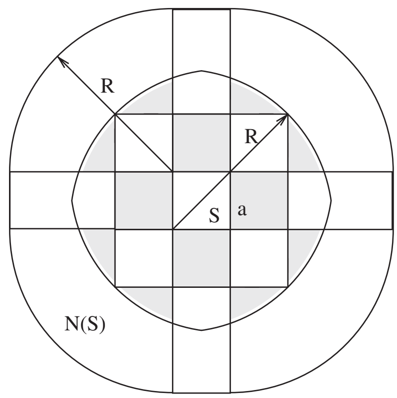

Fig. 4. For any node in region S, having k high priority nodes in each gray region guarantees k-node disjoint replacement paths between each pair of its neighbors.

2-hop information; otherwise, both nodes 1 and 3 will be backbone nodes. When node 1 uses restricted 2-hop information, it can only find two replacement paths between neighbors 2 and 8: (2, 8) and (2, 5, 8). The third node disjoint path (2, 6, 8) is invisible in restricted 2-hop information.

It has been proved in [11] that the expected size of the resultant CDS derived from the original coverage condition is $O(1)$ times the size of a minimal CDS in an optimal solution. Unfortunately, we cannot prove a similar bound for k-CDS with $k > 2$. Another extension of the coverage condition that holds this bound will be discussed in the next section.

Theorem 2. *The expected number of backbone nodes selected by the k-coverage condition is $O(k)$ times the size of a minimal k-CDS.*

Proof. Consider a square region S with side $a = R/2\sqrt{2}$ in Fig. 4. Neighbors of nodes in S are within a finite region $N(S)$. Without loss of generality, assume nodes are randomly "thrown" into $N(S)$ in the descending order of priority. Let X be a random variable such that, after the first X nodes has been thrown, there are at least k nodes in each of the 12 gray regions. Afterwards, all nodes thrown into S has a non-backbone status because their neighbors are k-connected via high priority nodes in these gray regions. Therefore, the number of backbone nodes in S is upper bounded by X.

Dai and Wu [11] showed that $E[X] = O(1)$ for $k = 1$. In order to get a bound for the general case, each node is given a color c $(1 \leqslant c \leqslant k)$. Specifically, the node has the mth highest priority in region $N(S)$ has a color $(m \bmod k) + 1$. Let X_i $(1 \leqslant i \leqslant k)$ be the random variable such that after throwing kX_i nodes, there is at least one node with color i in each of these gray regions. From the proof in [11], it is easy to see that $E[X_i] = O(1)$. Since $X \leqslant \max_{i=1}^{k}(kX_i) \leqslant k\sum_{i=1}^{k} X_i$, we have $E[X] \leqslant k\sum_{i=1}^{k} E[X_i] = O(k^2)$.

From above discussion, the expected backbone size in k-coverage condition is $O(k^2A')$, where A' is the total number of non-empty regions of size S in the deployment area. On the other hand, the size of a minimal k-CDS is $\Omega(kA')$. The ratio of the two is $O(k)$. □

The k-coverage condition depends on local information only. No global information such as network size is required. The size of the resultant virtual backbone is barely affected by the network density. The k-coverage condition has the same message size and rounds of information exchange as the original coverage condition. When 2-hop information is collected, each node sends two messages with size $O(\Delta)$. However, the k-coverage condition is more complex than the original condition. Each node needs to compute the vertex connectivity among $O(\Delta^2)$ pairs of neighbors using the maximal flow algorithm with time complexity $O(k|E|)$ as discussed in Section 3.1. When the algorithm uses restricted 2-hop information, $|E| = O(\Delta^2)$ and it takes $O(k\Delta^2)$ time to verify whether two neighbors are k-connected. The overall computation cost at each node is $O(k\Delta^4)$, which is higher than that of the original coverage condition ($O(\Delta^3)$). Although some density reduction methods [34] can be employed to reduce Δ in very dense networks, these methods also introduce extra overhead and slower convergency.

4. Color-based k-CDS construction

This section introduces a hybrid paradigm that enables 1-CDS algorithms to construct a k-CDS with high probability in relatively dense networks. Unlike pure probabilistic schemes, this approach does not depend on any network parameter. This approach is also easier to implement and has lower overhead than the deterministic algorithm discussed in the previous section. We use Wu and Dai's [33] coverage condition as an example to show how to convert a CDS algorithm using this paradigm.

4.1. A hybrid paradigm

As shown in the last section, when extending an existing CDS algorithm to construct k-CDS, the original algorithm needs to be modified, and usually becomes more complex in concepts and implementation techniques. In this section, we propose a hybrid paradigm, called color-based k-CDS construction (CBKC), to make the migration process simpler. The basic idea is to randomly partition the network into several subnetworks with different colors, and apply a traditional CDS algorithm to each subnetwork. The first step is probabilistic; when the network is sufficiently dense, colored nodes in each partition almost always form a CDS of the original network. The second step is deterministic; each *colored backbone* constructed within a subnetwork by a CDS algorithm is still a CDS of the entire network. Together, k-colored backbones form a k-CDS. Since any CDS algorithm $\mathcal{A}$ can be used in constructions of colored backbones, our color-based scheme provides a general framework for extending a wide range of existing CDS algorithms to construct k-CDS in relatively dense wireless networks.

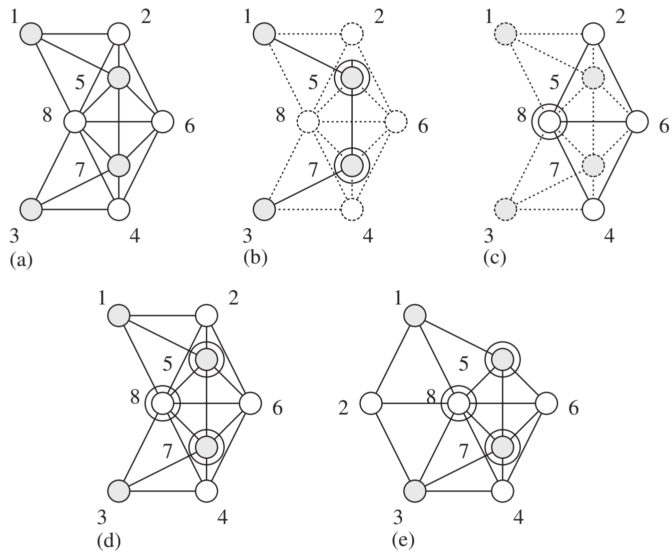

Fig. 5. Color-based coverage condition. (a) Nodes with odd ID numbers are of color 1 (gray), and nodes with even IDs are of color 2 (white). (b, c) Two colored virtual backbones (represented by double circles) are constructed using the coverage condition. Nodes with different colors and their adjacent links (represented by dotted circles and lines) are not considered by CBCC-II. (d) The final 2-CDS consists of all backbone nodes. (e) CBCC-II fails when a colored backbone does not form a CDS of the entire network.

Color-based k-CDS construction (CBKC)

(1) Each node v selects a random color c_v $(1 \leqslant c_v \leqslant k)$ for itself. As a result, the node set V is divided into k disjoint subsets $V_1, V_2, \ldots, V_k$, with each subset V_c containing nodes with color c.
(2) For each color c, a localized CDS algorithm $\mathcal{A}$ is applied to construct a virtual backbone $V'_c \subset V_c$ that covers the original network.
(3) The final k-CDS is the union $\bigcup_{c=1}^{k} V'_c$ of all colored virtual backbones.

Fig. 5 illustrates CBKC process. In Fig. 5(a), all nodes are randomly assigned color (1) gray and (2) white. In Fig. 5(b), two gray nodes 5 and 7 are selected to form a CDS of the entire network. In Fig. 5(c), a single node 8 is selected from white nodes to form a CDS. The set of all backbone nodes {5, 7, 8} forms a 2-CDS of the network, as shown by Fig. 5(d). The following theorem shows that the above generic scheme almost always constructs a k-CDS in dense networks.

Theorem 3. *If all nodes in the network are randomly placed in a finite square region, then CBKC almost always constructs a k-CDS when the node number exceeds a constant n_k.*

Proof. We first show that each node set V_c formed at step 1 is a CDS of the network G with high probability when node number is sufficiently large. It has been proved in [21] that given a probability p and a radius r, there exists a $n(p, r)$ such that when $n \geqslant n(p, r)$ nodes are randomly deployed in a unit square, and each node is marked a color c with probability p, then the entire region is almost always covered by those marked nodes (i.e., every point in this region is within distance r of a marked node). Suppose both the actual square area A and the actual transmission range R are fixed. Let $n_k = n\left(\frac{1}{k}, \frac{R}{2\sqrt{A}}\right)$. It is easy to see that when $n \geqslant n_k$ nodes are randomly and uniformly divided into k sets $V_1, V_2, \ldots, V_k$, each set V_c almost always covers the region under transmission range $R/2$. It has been proved in [32] that a set achieving area coverage with covering radius $R/2$ is connected under transmission range R. Therefore, each V_c is a CDS of G.

When each set V_c is a CDS of G, the virtual backbone V'_c selected by algorithm $\mathcal{A}$ in step 2 is also a CDS of G. Let $V' = \bigcup_{c=1}^{k} V'_c$ be the union of k-node disjoint CDSs of G. After removing $k-1$ from V', there is at least one V'_c untouched. Therefore, the remaining nodes in V' still form a CDS of G. From Lemma 1, V' is a k-CDS of G. □

4.2. *Color-based coverage condition*

We use the coverage condition as an example to illustrate the effectiveness of the color-based paradigm. When the original coverage condition is extended using the CBKC framework, only one modification is needed in the following revised rule:

Color-based coverage condition (*CBCC*): Node v has a non-backbone status if for any two neighbors u and w, a replacement path exists that connects u and w via several intermediate nodes (if any) *with the same color* and higher priorities than that of v.

Fig. 5(a–d) shows an example of CBCC. Note that with the CBCC, the search for a replacement path is now restricted to nodes with the same color. This modification actually reduces the average computation cost, but the worst case computation complexity is still the same ($O(\Delta^3)$). CBCC also inherits the constant probabilistic bound of the original coverage condition [11].

Theorem 4. *The expected number of backbone nodes selected by CBCC is $O(1)$ times the optimal value.*

Proof. It was shown in [11] that the expected number of backbone nodes selected by the original coverage condition is $O(A/R^2)$, where A is the area of a rectangular deployment region and R is the transmission range. Since the virtual backbone constructed by CBCC consists of k-colored backbones, the total number of backbone nodes is $O(kA/R^2)$. Note that any k-dominating set needs at least $O(kA/R^2)$ nodes to maintain k-coverage. Therefore, the expected backbone size of CBCC is $O(1)$ times the minimal k-dominating set, which is no larger than a minimal k-CDS. □

To further reduce the message and computation cost, we consider a more aggressive variation of CBCC. The original color-based coverage condition (called CBCC-I) covers all neighbors regardless of their colors; that is, any two neighbors of a non-backbone node must be connected via a replacement path. For example, node 3 in Fig. 5(e) is a backbone node in CBCC-I, because it has two neighbors 2 and 7 that are not connected via a gray replacement path. In the more aggressive variation (called CBCC-II), only neighbors with the same color are considered. As shown in Fig. 5(b), when a gray node is applying CBCC-II, all white nodes are excluded from its 2-hop information. The same rule also applies in white backbone construction, as shown in Fig. 5(c).

Compared to CBCC-I, CBCC-II uses smaller "Hello" messages to collect 2-hop information, has lower computation cost, and constructs a smaller backbone. However, the worst case performance and overhead of both variations are the same. Since CBCC-II is more aggressive than CBCC-I, its probability of constructing a k-CDS is lower than CBCC-I. As shown in Fig. 5(e), when node 3 uses CBCC-II to determine its status, it becomes a non-backbone node because it has only one visible neighbor 7. However, the resultant gray backbone $\{5, 7\}$ is not a CDS of the entire network, and union of all backbone nodes $\{5, 7, 8\}$ is not 2-dominating. The failure of node 8 will leave node 2 uncovered. Note that, however, when the network is very dense and node coverage is a good approximation of area coverage, the probability is high that CBCC-II selects a CDS of the entire network for each color, and the final backbone is a k-CDS.

5. Simulation

We conduct simulation study to evaluate the performance of three proposed k-CDS construction algorithms. Simulation results show that a small k-CDS can be formed with high probability and relatively low overhead in those schemes.

5.1. Implementation

All proposed protocols have been implemented on a custom simulator *ds*.[1] All simulations are conducted in randomly generated static networks. We assume an ideal situation where all messages are received by their neighbors without losses. Simulations in more realistic networks with mobility, collision, and contention will be future work. To generate a network, n nodes are randomly placed in a $1000\,\text{m} \times 1000\,\text{m}$ region. The transmission range R is 250 m. Any two nodes with distance less than R are considered neighbors. Each simulation is repeated 500 times, and uses the average data as the final result. Both k-coverage condition and color-based schemes use restricted 2-hop information to reduce computation overhead.

All k-CDS protocols, k-Gossip, k-Grid, k-coverage condition (k-Coverage), and two variations of the color-based coverage condition (CBCC-I and CBCC-II), are evaluated with $k = 2$ and 3, where the following performance metrics are compared:

- *Success ratio*, defined as S/T, where T is total number of networks that are k-connected, and S is the count that a protocol successfully forms a k-CDS. High success ratio is essential for the reliability of a k-CDS protocol.
- *Backbone size*, i.e., average number of backbone nodes selected by a protocol. A smaller backbone size means lower bandwidth and energy consumption by the k-CDS.
- *Tolerable failure ratio*. We define the failure ratio as the fraction of failed backbone nodes. For a k-CDS protocol, its tolerable failure ratio is the maximal average failure ratio that the remaining backbone nodes still form a 1-CDS. It indicates the robustness of a backbone.
- *Message overhead*, measured as the average number of bytes sent by each node during the k-CDS construction process. Note the number of messages sent by each node is equivalent to the number of "Hello" exchange rounds, as shown in Table 1.

Fig. 6 shows sample virtual backbones constructed by five protocols with $k = 2$ in a network with 100 nodes. We selected $p_k = 48\%$ in k-Gossip for a high success ratio. The resultant virtual backbone consists of 48 nodes and a 2-CDS of the network (as shown in Fig. 6(a)). Such a backbone survives five random node failures, but was disconnected after the sixth failure (as shown in Fig. 6(b)). In k-Grid, using a $B_k = 11$ produces a large 4-CDS with 77 backbone nodes (as shown in Fig. 6(c)). k-Coverage selects 53 nodes and forms a 2-CDS (as shown in Fig. 6(d)). Both color-based schemes divide the network into two equal partitions with different colors (represented by different node shapes). CBCC-I selects 57 backbone nodes (as shown in Fig. 6(e)), and CBCC-I selects 45 backbone nodes (as shown in Fig. 6(f)). Both schemes constructs a 2-CDS. Overall, CBCC-II has the smallest backbone size, and k-Grid achieves the highest connectivity in this specific network.

5.2. Simulation results

Success ratio: Fig. 7 compares the success ratio of four algorithms in constructing 2-CDS and 3-CDS, when the node number n varies from 100 to 300. The probability p_k in k-Gossip and density B_k in k-Gossip are determined based on our previous experimental data in networks with 200 nodes (as shown in Fig. 3). We assume that each node has no access to global information, and uses fixed a p_k or B_k in all networks.

As shown in Fig. 7, k-Coverage has 100% success ratio in all circumstances, which confirms our claim in Theorem 1: k-Coverage guarantees a k-CDS in all k-connected networks. CBCC-I has a very high success ratio in relatively dense networks. For $k = 2$, it has 99% success ratio in networks with more than 150 nodes. For $k = 3$, its success ratio is larger than 97% when $n \geqslant 200$. Again, these results confirm our conclusion in Theorem 3: the original color-based scheme almost always forms a k-CDS in dense networks.

The success ratio of k-Gossip is low in sparse networks. When $n = 100$, its success ratio is 22.0% when $k = 2$ and 12.9% when $k = 3$. However, its success ratio improves as the network density increases, and exceeds 90% after $n \geqslant 200$. Similarly, there is no single perfect B_k in k-Grid for all networks. The success ratio is high in sparse networks and becomes lower in dense networks, but the difference is not as obvious as in k-Gossip. CBCC-II has the lowest success ratio,

[1] Check http://sourceforge.net/projects/wrss/ for more details of the simulator.

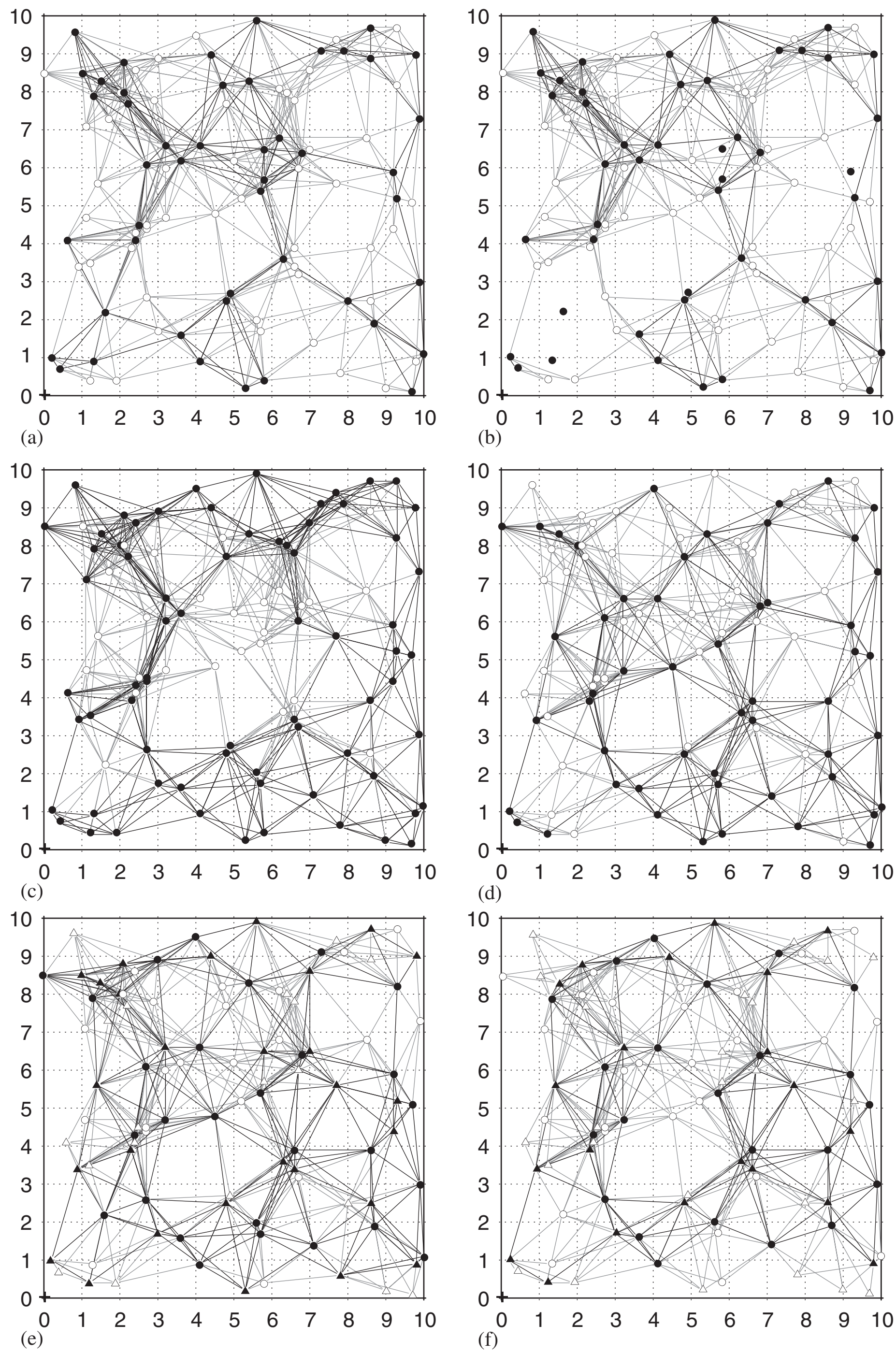

Fig. 6. Sample virtual backbones constructed by different protocols with $k = 2$: (a) k-Gossip, $p_k = 48\%$, size 48, 2-CDS, (b) partition after 6 node failures, (c) k-Grid, $B_k = 11$, size 72, 4-CDS, (d) k-Coverage, size 53, 2-CDS, (e) CBCC-I, size 57, 2-CDS, and (f) CBCC-II, size 45, 2-CDS.

except when $k = 2$ and $n \leqslant 150$. Its best performance is 84% for $k = 2$ and 73% for $k = 3$. The assumption behind CBCC-II is that node coverage is a good approximation of area coverage in very dense networks. Obviously, the simulated networks are not sufficiently dense to make this scenario really happen.

Backbone size: Fig. 8 compares virtual backbone size in 2-CDS and 3-CDS construction. k-Gossip usually produces the largest backbone. This is because we use a fixed p_k in the simulation, which selects np_k nodes on average. That is, the backbone size increases with the node number n. Using a variable p_k in k-Gossip is possible, but requires global information and experimental data to determine a perfect value of p_k for each network configuration. The first requirement incurs extra runtime overhead, and the second increases the preparation cost.

The other three algorithms have relatively small backbone sizes, which increases slightly as n increases. Among them, k-Coverage has the best performance in dense networks, CBCC-II produces the smallest backbone in sparse networks, and

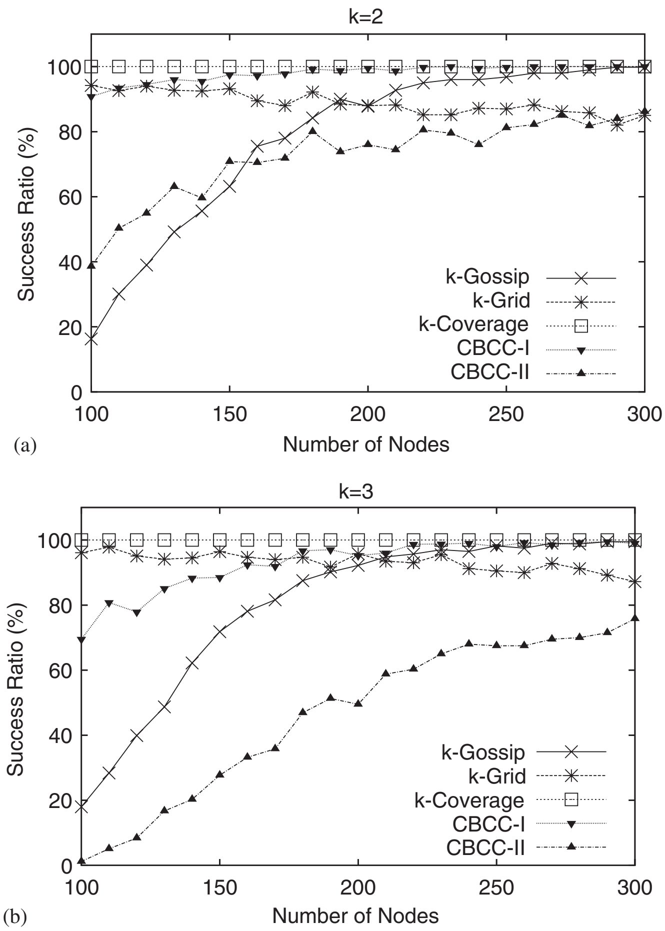

Fig. 7. Success ratio: (a) 2-CDS ($p_k = 48$, $B_k = 11$), (b) 3-CDS ($p_k = 62$, $B_k = 14$).

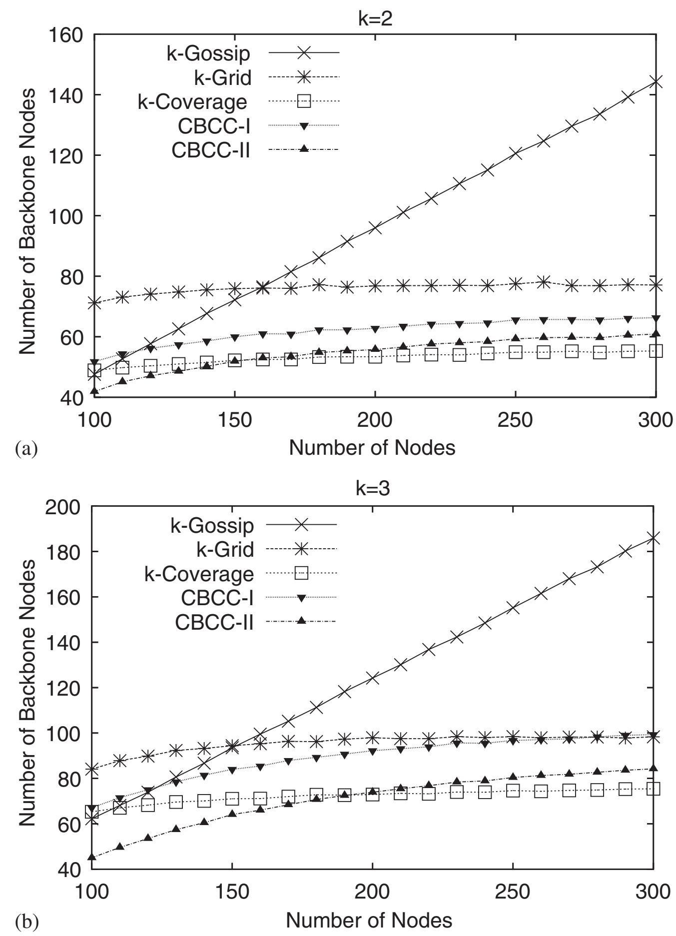

Fig. 8. Backbone size: (a) 2-CDS ($p_k = 48$, $B_k = 11$), (b) 3-CDS ($p_k = 62$, $B_k = 14$).

CBCC-I has the worst performance in all scenarios. Since CBCC-II can merely maintain a k-CDS in sparse networks, k-Coverage is actually the best choice in terms of virtual backbone size. Our explanations to this phenomenon are: first, all coverage condition-based schemes seems to have probabilistic upper bound in dense networks (even though we cannot prove it for k-Coverage). Therefore, we will not see a proportional increase of the backbone size as in k-Gossip. Second, maintaining k separate 1-CDSs incurs higher redundancy than preserving a single k-CDS, which results in more backbone nodes in the color-based schemes.

Tolerable failure ratio: When a k-CDS is constructed for fault-tolerance, i.e., maintaining a 1-CDS with high probability, its ability to survive a large percentage of backbone node failures is essential. As shown in Fig. 9(a), when all protocols aim to construct a 2-CDS, they can usually tolerate 35–45% backbone node failures. Two exceptions are k-Gossip and CBCC-II, which have low (15–25%) tolerable failure ratios in sparse networks ($n \leqslant 150$). In dense networks, k-Gossip can tolerate a large percent (70%) of node failures.

Generally speaking, a large backbone tolerates more node failures. One interesting observation is that both CBCC-I and CBCC-II have similar tolerable failure ratios to that of k-Grid, although k-Grid has a large backbone size. The color-based schemes are more effective than k-Grid because they make smarter decisions using 2-hop information. Another conclusion is that, in dense networks, CBCC-II is as good as CBCC-I in terms of fault tolerance.

Message overhead: Fig. 9(b) shows the average number of bytes sent by each node during k-CDS construction. k-Gossip does not use neighborhood information; each node sends no messages. k-Grid uses 1-hop information and sends one message with a fixed size. k-Coverage and two color-based schemes use 2-hop information; each node sends two messages. The first one has a fixed size; the size of the second one is proportional to the number of neighbors. CBCC-II has a lower message overhead than k-Coverage and CBCC-I, because its second message carries only information of neighbors in the same color. When $k = 2$, the average bytes sent by CBCC-II is roughly half that sent by k-Coverage and CBCC-II.

Simulation results can be summarized as follows:

(1) k-Gossip has the lowest overhead and high success ratio in dense networks, but also has serious problems. When a fixed p_k is used, it has low reliability in sparse networks and a large backbone size in dense networks.

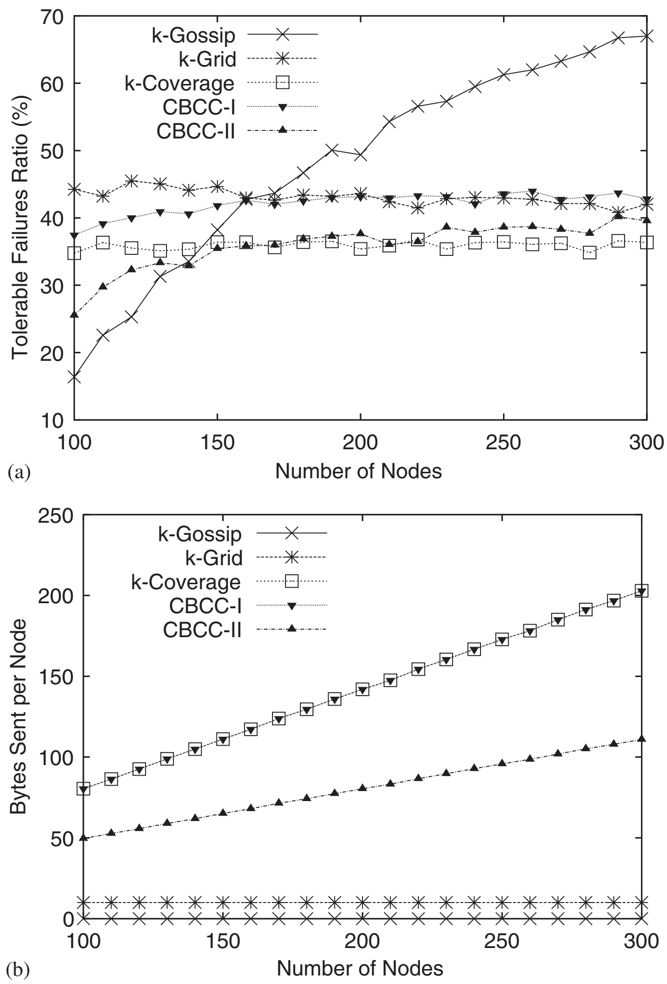

Fig. 9. Reliability and overhead ($k = 2$): (a) tolerable failure ratio, (b) average message overhead.

(2) k-Grid incurs a low message overhead and a reasonable success ratio, but produces a relatively large backbone.
(3) k-Coverage guarantees 100% success ratio and selects a smallest backbone in most scenarios. Its only weakness is the relatively complicated algorithm and high computation cost.
(4) CBCC-I has lower overhead than k-Coverage, and almost always constructs a k-CDS in relatively dense networks. The resultant backbone size is larger than in k-Coverage, but much smaller than k-Gossip.
(5) CBCC-II has lower overhead than CBCC-I, but does not show a satisfactory success ratio in our simulation. However, high success ratio may still be observed in very dense networks.

6. Conclusion

This paper proposes four localized protocols that construct a k-connected k-dominating set (k-CDS) as a virtual backbone of wireless networks. Three protocols are extensions of existing CDS algorithms. The fourth scheme is a generic paradigm, which enables many existing virtual backbone formation algorithms to produce a k-CDS with high probability. Our simulation results show that these protocols can select a small k-CDS with relatively low overhead.

As future work, we plan to conduct extensive simulation study on the performance of k-CDS in carrying out important tasks such as routing and area monitoring. We will also try to find a tighter probabilistic approximation ratio of the k-coverage condition (if one exists).

References

[1] C. Adjih, P. Jacquet, L. Viennot, Computing connected dominated sets with multipoint relays, Technical Report 4597, INRIA-Rapport de recherche, October 2002.
[2] K.M. Alzoubi, P.-J. Wan, O. Frieder, Distributed heuristics for connected dominating sets in wireless ad hoc networks, J. Comm. Networks 4 (1) (2002) 22–29.
[3] M. Bahramgiri, M. Hajiaghayi, V.S. Mirrokni, Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks, in: Proceedings of ICCCN, 2002, pp. 392–397.
[4] L. Bao, J.J. Garcia-Luna-Aceves, Topology management in ad hoc networks, in: Proceedings of MobiHoc, 2003, pp. 129–140.
[5] S. Basagni, M. Mastrogiovanni, A. Panconesi, C. Petrioli, Localized protocols for ad hoc clustering and backbone formation: a performance comparison, IEEE Trans. Parallel Distributed Syst. 17 (4) (2006) 292–306.
[6] P. Basu, J. Redi, Movement control algorithms for realization of fault tolerant ad hoc robot networks, IEEE Networks 18 (4) (2004) 36–44.
[7] D. Blough, M. Leoncini, G. Resta, P. Santi, The K-Neigh protocol for symmetric topology control in ad hoc networks, in: Proceedings of MobiHoc, 2003, pp. 141–152.
[8] J. Carle, D. Simplot-Ryl, Energy-efficient area monitoring for sensor networks, IEEE Comput. 37 (2) (2004) 40–46.
[9] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks, ACM Wireless Networks J. 8 (5) (2002) 481–494.
[10] B.N. Clark, C.J. Colbourn, D.S. Johnson, Unit disk graphs, Discrete Math. 86 (1990) 165–177.
[11] F. Dai, J. Wu, An extended localized algorithm for connected dominating set formation in ad hoc wireless networks, IEEE Trans. Parallel Distributed Syst. 15 (10) (2004) 908–920.
[12] B. Das, R. Sivakumar, V. Bhargavan, Routing in ad hoc networks using a spine, in: Proceedings of ICCCN, 1997, pp. 1–20.
[13] J. Edmonds, Paths, trees and flowers, Canad. J. Math. (17) (1965) 449–467.
[14] A. Gibbons, Algorithmic Graph Theory, Cambridge University Press, Cambridge, 1985.
[15] Z.J. Haas, J.Y. Halpern, L.Li, Gossip-based ad hoc routing, in: Proceedings of INFOCOM, vol. 3, 2002, pp. 1707–1716.
[16] F. Harary, T. Haynes, Double domination in graphs, ARS Combin. 55 (2000) 201–213.
[17] X. Hou, D. Tipper, Gossip-based sleep protocol (GSP) for energy efficient routing in wireless ad hoc networks, in: Proceedings of WCNC, 2004.
[18] M. Jorgic, I. Stojmenovic, M. Hauspie, D. Simplot-Ry1. Localized algorithms for detection of critical nodes and links for connectivity in ad hoc networks, in: Proceedings of 3rd IFIP Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET 2004).
[19] H. Koubaa, E. Fleury, On the performance of double domination in ad hoc networks, in: Proceedings of IFIP Medhoc, 2000.
[20] F. Kuhn, T. Moscibroda, R. Wattenhofer, Initializing newly deployed ad hoc and sensor networks, in: Proceedings of MobiCom, 2004, pp. 260–274.
[21] S. Kumar, T.H. Lai, J. Balogh, On k-coverage in a mostly sleeping sensor network, in: Proceedings of MobiCom, 2004, pp. 144–158.

[22] N. Li, J.C. Hou, FLSS: a fault-tolerant topology control algorithm for wireless networks, in: Proceedings of MobiCom, 2004, pp. 275–286.
[23] X.-Y. Li, P.-J. Wan, Y. Wang, C.-W. Yi, Fault tolerant deployment and topology control in wireless networks, in: Proceedings of MobiHoc, 2003, pp. 117–128.
[24] C.S. Liao, G.J. Change, *k*-tuple domination in graphs, Inform. Process. Lett. 87 (1) (2003) 45–50.
[25] C.R. Lin, M. Gerla, Adaptive clustering for mobile wireless networks, IEEE J. Selected Areas Commun. 15 (7) (1996) 1265–1275.
[26] J. Liu, B. Li, MobileGrid: capacity-aware topology control in mobile ad hoc networks, in: Proceedings of ICCCN, 2002, pp. 570–574.
[27] M. Lu, J. Wu, M. Cardei, M. Li, Energy-efficient connected coverage of discrete targets in wireless sensor networks, in: Proceedings of ICCNMC, 2005.
[28] K. Menger, Zur allgemeinen kurventheorie, Fund. Math. (10) (1965) 96–115.
[29] A. Qayyum, L. Viennot, A. Laouiti, Multipoint relaying for flooding broadcast message in mobile wireless networks, in: Proceedings of HICSS, vol. 9, 2002, p. 298.
[30] J. Shaikh, J. Solano, I. Stojmenovic, J. Wu, New metrics for dominating set based energy efficient activity scheduling in ad hoc networks, in: Proceedings of WLN, 2003, pp. 20–24.
[31] I. Stojmenovic, M. Seddigh, J. Zunic, Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks, IEEE Trans. Parallel Distributed Syst. 13 (1) (2002) 14–25.
[32] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration in wireless sensor networks, in: Proceedings of SenSys, 2003, pp. 28–39.
[33] J. Wu, F. Dai, A generic distributed broadcast scheme in ad hoc wireless networks, IEEE Trans. Comput. 53 (10) (2004) 1343–1354.
[34] J. Wu, F. Dai, A distributed formation of a virtual backbone in manets using adjustable transmission ranges, in: Proceedings of ICDCS, 2004, pp. 372–379.
[35] J. Wu, H. Li, On calculating connected dominating sets for efficient routing in ad hoc wireless networks, in: Proceedings of DialM, 1999, pp. 7–14.

Fei Dai is an Assistant Professor in the Department of Electrical and Computer Engineering at North Dakota State University. Dr. Dai received his Ph.D. in the Department of Computer Science and Engineering at Florida Atlantic University, and M.S. in the Department of Computer Science and Technology at Nanjing University, China. He has been a senior programmer in Greatwall Computer and a software architect and team leader in J&A Securities, both in China. His research interests include networking, mobile computing, parallel and distributed computing, artificial intelligence, and software engineering.

Jie Wu is a professor in the Department of Computer Science and Engineering, Florida Atlantic University. He has published more than 200 papers in various journal and conference proceedings. His research interests are in the area of mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. Dr. Wu served as a program vice chair for the 2000 International Conference on Parallel Processing (ICPP) and a program vice chair for the 2001 IEEE International Conference on Distributed Computing Systems (ICDCS). He is a program cochair for the IEEE First International Conference on Mobile Ad-Hoc and Sensor Systems (MASS ’04). He was a co-guest-editor of a special issue in Computer on ad hoc networks. He also edited several special issues in the Journal of Parallel and Distributing Computing (JPDC) and the IEEE Transactions on Parallel and Distributed Systems (TPDS). He is the author of the text Distributed System Design (CRC Press). He served as an IEEE Computer Society Distinguished Visitor and is currently the chair of the IEEE Technical Committee on Distributed Processing (TCDP). He is a member of the ACM and a senior member of the IEEE and the IEEE Computer Society.