Fault-tolerant k-fold Pivot Routing in Wireless Sensor Networks

Doina Bein
Department of Computer Science
University of Texas at Dallas
Richardson, TX 75083
Email: siona@utdallas.edu

Abstract

Selecting a small set of nodes called pivots*, from all the nodes in a network and maintaining the routing infrastructure to and among each other can reduce routing overhead and excessive broadcast redundancy.*

In this paper a new problem, called "k-fold cover t-set*" (for short, k-fold cover), is proposed: we select a smaller set of nodes than the k-fold dominating set, to act as alternative routers for messages in the network. An advantage of the k-fold cover is that it can be defined for graphs where the k-fold dominating set cannot be defined.*

Computing the minimum size k-fold cover for any $k \geq 1$ *is* $\mathcal{NP}$*-complete. Two approximation algorithms for the k-fold cover problem are proposed: one uses a greedy approach to compute the cover and the other uses randomization.*

Keywords*: cover, dominating set, fault tolerance, k-fold dominating set, pivots, routers, sensor network, wireless communication.*

1 Introduction

Network communication (wired or wireless) is frequently affected by link or node crashes, temporary failure, congestion (wired), bandwidth. In an ad hoc environment the set of immediate or nearer neighbors can change at arbitrary moments of time. In sensor networks the battery determines the lifetime of a node; a node failure can occur frequently as the node becomes older. Thus in order to increase the throughput on routing the packets, nodes should use fewer routes (a so-called *communication backbone* of a network). At the same time, since in an ad hoc or sensor network a node can fail or move somewhere else with high frequency, the selected nodes must be provided with enough redundancy for communication. This is done by them acting as alternative routers, and alternative routes have to be available before crashes affect the communication backbone.

Wireless communication is less stable than wired, since it depends on the nodes' power levels (which affect the transmission range of individual nodes), though it is more flexible and has lower cost. In a wireless environment, every node u in the transmission range of another node v can receive messages from v, and send them to other nodes outside the range of v. Unlike in the wired environment, bidirectional communication is not guaranteed between any pairs of nodes, since their communication range is not fixed and can vary also based on node power.

Current research in wireless networks focuses on networks where nodes themselves are responsible for building and maintaining proper routing (self-configure, self-managing). For all the above reasons, hierarchical structures as dominating sets or a link-cluster architecture are not able to provide sufficiently fast redundancy or fault-tolerance for high-speed or real-time networks, where the latency in every node should be very short. The time spent at every node to decide how to route messages should be comparable to the time of the propagation delay between neighboring nodes.

The dominating set problem (DS for short) is defined as follows. Given a weighted graph $G = (V, E)$ where each node $v \in V$ has associated a weight w_v, and a positive integer m, $0 < m \leq |V|$, find a subset D of size at most m of nodes in V such that every $v \in V$ is either in D or has at least one immediate neighbor in D, and the sum $\sum_{v \in D} w_v$ is minimized.

The corresponding minimization problem is to find the minimum dominating set (MDS), i.e. the size of D is to be minimized. It is approximable within a factor of $1 + log|V|$, but is not approximable within c for any constant $c > 0$. Generally, the graph G is considered unweighted ($w_v = 1$, for all $v \in V$).

DS is a particular case of the k-fold dominating set problem defined formally as follows. Given a weighted graph $G = (V, E)$ where each node $v \in V$ has associated a weight w_v, and two positive integers k and m, $0 < k \leq |V|$ and $0 < m \leq |V|$, find a subset C of size at most m of nodes in V such that every node $v \in V \setminus C$ has at least k immediate neighbors in C, every node in C has at least $k-1$ immediate neighbors in C, and the sum $\sum_{v \in C} w_v$ is minimized. We refer to the minimization version of the k-fold dominating set problem as k-fold MDS. Note that MDS is thus a particular case of k-fold MDS where $k = 1$. Not every graph has a k-fold dominating set for some arbitrary k; in that case it is called *unfolding*.

Obviously, a minimum requirement for a network to have a k-fold dominating set is that $k \leq c$, where c is the network connectivity. A graph has *connectivity* of c if by removing any $1, 2, \ldots, c-1$ nodes, the graph remains connected.

A natural way to minimize the number of nodes in the k-fold dominating set is to select not only nodes from the immediate neighborhood but also nodes located at two or more hops. In this way we do not only have the nodes in the k-fold cover set D act as a backbone for the communication in the network but they can also take on he role of alternative routers for the entire network. Thus the network can tolerate up to k node failures without losing the routing infrastructure.

A *k-fold cover t-set* (for short, k-fold cover) can be defined as a natural extension of the k-fold dominating set by replacing the term "immediate neighbors" with the term of "nearer neighbors". Given a fixed parameter t, every node v gathers information about its t nearest neighbors (in terms of distance) – the set called the t-ball of v, B_v (Eilam et al [5]). A set $D \subseteq V$ is a *k-fold cover t-set* if, for any node v, $|D \cap B_v| \geq k$ (D contains at least k nodes from any t-ball in the network). The elements of D are called *pivots*.

There is a tradeoff between selecting a smaller set of nodes and the power consumption of the selected nodes. Maintaining the routing infrastructure to only a subset of nodes reduces the routing overhead and excessive broadcast redundancy. At the same time, to save energy, unused non-backbone nodes can go into a sleeping mode and wake up only when they have to forward data or selected themselves in the k-fold cover set, due to failure or movement of nodes. At the same time, selecting fewer nodes to act as routers has a power consumption disadvantage: The routers will deplete their power faster than the non-router nodes. Thus once the power level of these nodes falls below a certain threshold, those nodes can be excludes as routers, and some other nodes have to replace them. Thus the k-fold cover set model proposes in this paper can fit better with wireless networks with low traffic or without power constraint.

Related Work. Two particular cases of the DS problem are the connected dominating set and the weakly connected dominating set. The *connected dominating set* problem requires the set of nodes selected as the dominating set to form a connected subgraph of the original graph. The *weakly connected dominating set* problem, defined by Dunbar *et al.* [4], requires the subgraph induced by the nodes in the set and their immediate neighbors to form a connected subgraph of the original graph.

Finding the *k-fold* dominating set for an arbitrary graph was suggested for the first time by Kratochvil [8]; the problem is $\mathcal{NP}$-complete (Kratochvil et al [9]). Liao and Change [12] showed that it is $\mathcal{NP}$-complete for split graphs (a subclass of chordal graphs) and for bipartite graphs. The particular case $k = 1$ is the minimum dominating set problem (MDS) and it is $\mathcal{NP}$-complete for any type of graphs.

Approximation algorithms for finding the *k-fold* dominating set in general graphs are given by Vazirani [18] and Kuhn et al [10]. The first algorithm for an arbitrary graph of n nodes and the maximum degree of a node Δ by Kuhn et al [10] runs in $O(t^2)$ and gives an approximation ratio of $O(t\Delta^{2/t} \log \Delta)$ for some parameter t. The second algorithm, for a unit disk graph, is probabilistic and runs in $O(\log \log n)$ time, with an $O(1)$ expected approximation.

The k-fold dominating set is related to the k-dominating set ([11, 19, 17]), and to the k-connected k-dominating set ([2, 3]). The k-dominating set $D \subseteq V$ has the property that every node in G is not further than $k-1$ hops from a node in D. Finding the minimum k-dominating set is $\mathcal{NP}$-hard, and a number of approximation algorithms are proposed in [11, 17]. The k-connected k-dominating set has the property that it is k-connected [1] and every node is either in the set or has k immediate neighbors in the set.

Contributions. If the degree of some node is smaller than k, the k-fold dominating set has no solution, while the graph may still have a k-fold cover set. The new problem presented in this paper, k-fold cover set, has the advantage that it can be defined for graphs in which the k-fold dominating set cannot be defined. We conjecture for any given graph that the size of the k-fold cover set is no greater than the size of the k-fold dominating set. Then the advantages of selecting fewer nodes to act as routers make the k-fold cover set

[1] by removing at most $k-1$ nodes the graph remains connected

a better choice for the different flavors of dominating set: connected dominating set, weakly connected dominating set, k-fold dominating set.

Finding the k-fold cover of small size can be used in the placement of base stations in cellular networks ([15, 6]), multicast systems ([20]), caching or partial replicas in databases ([16, 14, 7]).

Computing the minimum size k-fold cover for any $k \geq 1$ is $\mathcal{NP}$-complete. The two techniques from Awerbuch et al [1] are extended in this paper to generate relatively small k-fold covers for a given collection of sets of fixed size t. The first technique uses a greedy approach to select the nodes, and we show that a greedy k-fold cover t-set D for a graph G with n nodes has at most $n \cdot k \cdot (ln\ n + 1)/t$ elements. The second technique selects each node with a certain probability and we show that the expected size of a randomized k-fold cover for G is at most $2 \cdot c \cdot n \cdot k \cdot ln\ n/t$, with the probability of at least $1 - 1/n^{c-1}$ for some constant $c > 1$.

Outline. In Section 2 we describe the network model used, the ordering of nodes based on distance (weight), and the k-fold cover t-set. In Section 3 we present two approximation algorithms for the k-fold cover, and we give upper bounds for the constructed covers. A brief comparison between the k-fold cover set and k-fold dominating set is given in Section 4. We finish with concluding remarks in Section 5.

2 Preliminaries

A wireless network, which is a point-to-point communication network, is modeled as a connected, weighted, finite digraph $G = (V, E, w)$ that does not contain multi-arcs or self-loops.

The set of processors in the network have unique identification and they are represented by a set of nodes V, $|V| = n$. For the remainder of the paper, we assume that the unique IDs are a contiguous set $1..n$. Note that the unique identification of nodes induces a total order, thus for every pair of nodes $u, v \in V$, either $u < v$ or $v < u$.

An arc $e = (u, v) \in E$ is a unidirectional communication link that exists if the transmission range of u is large enough for any message sent by u to reach v. Every arc in the network $e = (u, v) \in E$ has associated a weight $w(e)$ that represents the cost (energy) spent by u to transmit a message that will reach v. Because of the non-linear attenuation property for radio signals, the energy consumption for sending is proportional to at least the square of the power of the transmission range. For simplicity we extend the domain of w to include all pairs of nodes in the network such that if node v is not within the transmission range of node u, then $w(u, v) = \infty$.

The wireless channel has a *broadcast* advantage that distinguishes it from the wired channel. When a node p uses an omnidirectional antenna every transmission by p can be received by all nodes located within its communication range. This notion is called *Wireless Multicast Advantage (WMA)*. Thus a single transmission from a node u suffices to reach all neighbors of u. Note that for any pair of nodes u and v, $w(u, v)$ is not necessarily equal to $w(v, u)$.

Given a simple directed path P in the graph the length of $|P|$ is the sum of the weights of the arcs of P. The distance $dist(u, v)$ in G is the length of the shortest path starting at u and ending at v.

For every node v we can order the rest of the nodes in the digraph G with respect to their distance from v, breaking ties by increasing node identification (Eilam et al [5]):

Formally, $x \prec_v y$ iff $dist(x, v) < dist(y, v)$ or $dist(x, v) = dist(y, v)$ and $x < y$. The *t-ball* of node v, $B_v(t)$, is the ordered set of the first t nodes according to the total order $\prec_v$.

Fact (Fact 2.1 from Eilam et al [5]) If $u \in B_v(t)$ then for every node x on the shortest path from v to u, $u \in B_x(t)$.

Definition 1 *Consider a collection $\mathcal{C}$ of subsets of size t of elements from the node set V. Each subset represents the t closest neighbors (in terms of distance) for some node v in V.*

Given a parameter k, $k > 0$, a set D is a k-fold cover t-set *for the collection $\mathcal{C}$, or* k-fold cover *for short, if for every $B \in \mathcal{C}$, D contains at least k elements from B, we write $|B \cap D| \geq k$.*

The problem of finding the minimum size k-fold cover set can be modeled as a linear program. Let $y_i \in \{0, 1\}$ be a binary variable associated with node $i \in \{1, \ldots, n\}$ such that $y_i = 1$ iff node i is in the k-fold cover. The goal is to minimize the sum of all y-variables such that in every t-ball of some node i there are at least k nodes from the k-fold cover, including node i if i is in the cover. In other words, nodes not in the k-fold dominating set have to be covered at least k times, and nodes in the k-fold dominating set have to be covered at least $k - 1$ times. The linear program follows:

$$min \sum_{i=1}^{n} y_i$$

$$\text{such that} \quad \sum_{j \in B_i(t) \cup \{i\}} y_j \geq k$$

$$y_i = 0 \vee y_i = 1$$

Next we present two techniques to approximate a solution for the minimum k-fold cover; one uses a greedy method to select the pivots, while the second one uses randomization.

3 Distributed k-fold Pivot t-set

Computing the minimum size k-fold cover for any $k \geq 1$ is $\mathcal{NP}$-complete. In this section we extend the two techniques from Awerbuch et al [1] to generate relatively small k-fold covers for a given collection of sets of fixed size t. The first technique (Algorithm $Cover1$) uses a greedy approach to select the nodes. The second technique (Algorithm $Cover2$) selects each node with a certain probability.

Algorithm $Cover1$ starts with the set D to be initially $\emptyset$ and iteratively adds to D an element in V occurring in the most uncovered sets. A set is *covered* if it contains at least k elements which are also in D. The set D is then called a *greedy k-fold cover* for $\mathcal{C}$.

As in [10] we use a coloring mechanism (variable c_i for some node i to distinguish between two types of nodes: *(i) gray* color for nodes that are not in the cover and are covered by at least k nodes in the cover, or nodes that are in the cover and are covered by at least $k-1$ nodes in the cover, and *(ii) white* color for the nodes that are not in the cover and are not covered by k nodes in the cover, and nodes that are in the cover and are not covered by at least $k-1$ nodes in the cover.

Each node in the graph knows n, the total number of nodes in the graph. Besides c_i it uses a variable x_i which takes discrete values in the set $\{1/(n-1), 2/(n-1), \ldots, (n-1)/(n-1)\}$. Initially, all the nodes are white, $D = \emptyset$, and all x-values are 0. Each node i sends a value $1/(n-1)$ to all the nodes in its t-ball $B_i(T)$, and value 0 to the nodes outside its t-ball, excluding itself.

There are at most $n+1$ rounds (where $n = |V|$).

In round 0, nodes will collect the values received from all the other nodes and the sum of all the received values is stored in x_i. For some node i variable x_i has a discrete value from the set $\{1/(n-1), 2/(n-1), \ldots, (n-1)/(n-1)\}$. The larger the value the higher is the chance for node i to be selected in the set D.

In round 1, all nodes with $x_i = (n-1)/(n-1) = 1$ announce themselves as pivots (they add themselves to D). Their x-value is already 1. Some white node i receives these messages and stores the IDs of the pivots that are also in its t-ball $B_i(t)$ in variable D_i. If $|D_i| \geq k$ then node i changes its color to gray.

If at the end of round 1 there are any white nodes, then the algorithm continues with round 2.

In round 2 all nodes with $x_i = (n-2)/(n-1)$ announce themselves as pivots (they add themselves to D), and set their x-value to 1. Some white node i receives these messages and adds the IDs of the pivots that are also in its t-ball $B_i(t)$ to variable D_i. If $|D_i| \geq k$ then node i changes its color to gray.

At the end of round 2, if there are white nodes, then the algorithm continues with round 3.

If at the end of round k, $1 \leq k \leq n-1$, all nodes are gray then the algorithm executes the last round, called *terminate*.

In round *terminate*, all nodes not selected in the k-fold cover ($x_i \neq 1$) reset their x-variable to 0.

Algorithm 3.1 Algorithm for $Cover1$

Actions executed at node i

Initialization (Round 0):
$x_i = 0$
$c_i = white$
$D_i = \emptyset$
$D = \emptyset$
send the value $1/(n-1)$ to each node in $B_i(t)$ and
the value 0 to each node in $(V \setminus \{i\}) \setminus B_i(t)$
$x_i = \sum_{\text{all } j \neq i}$ values sent by node j

Round k, $1 \leq k \leq n-1$:
if $x_i = (n-k)/(n-1)$ then
$x_i = 1$
broadcast the message $msg(pivot, i)$ that node i is a pivot
on receiving $msg(pivot, j)$
$D \leftarrow D \cup \{j\}$
if $(j \in B_i(t))$ then $D_i \leftarrow D_i \cup \{j\}$
if $(|D_i| \geq k) \vee (x_i = 1 \wedge |D_i| \geq k-1)$ then $c_i = gray$
if all nodes are gray, then terminate

Round terminate
if $x_i \neq 1$ then $x_i = 0$

Lemma 1 *Let D be a greedy k-fold cover for $\mathcal{C}$. Then $|D| < |V| \cdot k \cdot (ln|\mathcal{C}| + 1)/t$.*

Proof. If D_1 is a simple, greedy cover for $\mathcal{C}$, then $|D_1| < |V|(ln|\mathcal{C}| + 1)/t$ (Lovacs [13]).

Selecting an element for a simple cover of $\mathcal{C}$ corresponds to selecting at most k elements for a k-fold cover. Thus , if D is a greedy k-fold cover for $\mathcal{C}$, then $|D| < |V| \cdot k \cdot (ln|\mathcal{C}| + 1)/t$. □

Corollary 1 *If D is a k-fold cover constructed by Algorithm $Cover1$ for G, then $|D| < n \cdot k \cdot (ln\ n + 1)/t$.*

Proof. For our collection $\mathcal{C}$, $|\mathcal{C}| = |V| = n$ (one t-set for each node). Thus a greedy k-fold cover for G will have at most $n \cdot k \cdot (ln\ n + 1)/t$ elements. □

Given that $|D| \leq n$ one can derive some relationships between k and t from Lemma 1 and Corollary 1. If k is a constant and given that $t < n$, it follows that $t > k(ln\ n + 1)$. Conversely, if t is fixed, i.e. there is an upper bound on how many other nodes a node can keep track of, then it follows that $k < \frac{t}{(ln\ n+1)}$. Furthermore, if $D << n$ it follows that $t >> k(ln\ n + 1)$ (when k is fixed), respectively $k << \frac{t}{(ln\ n+1)}$ (when t is fixed).

For example, consider the network in Figure 1 ($n = 10$). Each node has associated a pair (ID_i, r_i), where ID_i is the unique identification of the node taken from the set $\{1, \ldots, n\}$, and r_i is the transmission radius of node i. As a result, the weight of any outgoing arc of i has the same value, which is proportional with r_i^2.

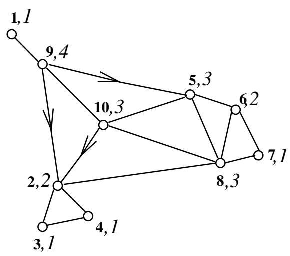

Figure 1. Network of $n = 10$ nodes

Let $t = \lfloor \sqrt{n \cdot ln(n)} \rfloor = 4$ and $k = 3$. Note that the minimum 3-fold cover is $D_{opt} = \{2, 5, 8, 10\}$.

While selecting the t-balls of size 4, ties are broken by node ID. We enumerate the nodes in a t-ball of some node i in non-ascending order of its distance with respect to i. The t-balls for the nodes in the network are: $B_1(4) = \{9, 2, 5, 10\}$, $B_2(4) = \{3, 4, 8, 5\}$, $B_3(4) = \{2, 4, 8, 5\}$, $B_4(4) = \{2, 3, 8, 5\}$, $B_5(4) = \{6, 8, 10, 2\}$, $B_6(4) = \{5, 7, 8, 2\}$, $B_7(4) = \{6, 8, 2, 5\}$, $B_8(4) = \{2, 5, 6, 7\}$, $B_9(4) = \{1, 2, 5, 10\}$, $B_{10}(4) = \{2, 5, 8, 9\}$,

At the end of round 0, all nodes are white (see Figure 2) and the x-values are: $x_1 = 1/9$, $x_2 = 9/9$, $x_3 = x_4 = 2/9$, $x_5 = 9/9$, $x_6 = 3/9$, $x_7 = 2/9$, $x_8 = 7/9$, $x_9 = 2/9$, $x_{10} = 3/9$.

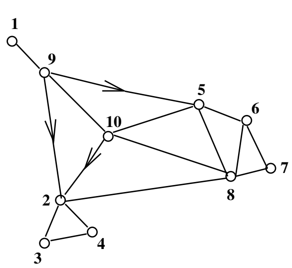

Figure 2. End of round 0

In round 1, nodes 2 and 5 are selected in the 3-fold cover (marked with a cross), and no node becomes gray (see Figure 3).

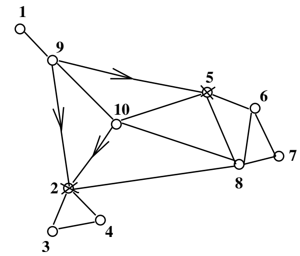

Figure 3. End of round 1

In round 2 no node is selected. In round 3, node 8 is selected in the 3-fold cover, and all nodes except 1 and 9 become gray (see Figure 4).

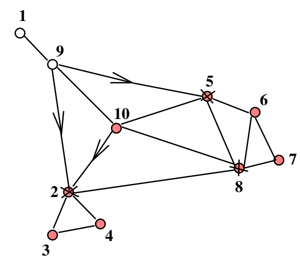

Figure 4. End of round 3

In round 4, 5, and 6 no node is selected. In round 7, nodes 6 and 10 are selected, and all the nodes are gray (see Figure 5).

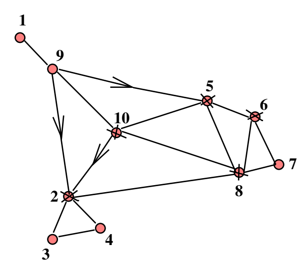

Figure 5. End of round 7

After round *terminate* executes, the greedy k-fold cover is $D = \{2, 5, 6, 8, 10\}$, which has only one extra node (node 6) compared to the optimum 3-fold cover $D_{opt} = \{2, 5, 8, 10\}$.

Randomization provides another method for building the cover. Algorithm Cover2 simply selects into D each element of V with the probability $c \cdot k \cdot ln|\mathcal{C}|$, for some constant $c > 1$. D is called a *randomized k-fold cover* for $\mathcal{C}$.

Algorithm 3.2 Algorithm for $Cover2$

Actions executed at node i

with the probability $c \cdot k \cdot ln|\mathcal{C}|$ set $x_i = 1$ else set $x_i = 0$
if $x_i = 1$ then broadcast $msg(pivot, i)$ (node i is a pivot)

Lemma 2 *Assuming $|V| > 2t$ and $ln|\mathcal{C}| = o(t)$, let D be a randomized k-fold cover for $\mathcal{C}$. Then with probability of at least $1 - 1/|\mathcal{C}|^{c-1}$, D is a k-fold cover for $\mathcal{C}$ and $|D| < 2c \cdot |V| \cdot k \cdot ln|\mathcal{C}|/t$, for some constant $c > 1$.*

Proof. With probability of at least $1 - 1/|\mathcal{C}|^{c-1}$, a randomized, simple cover set D_1 can be constructed for $\mathcal{C}$, and the expected size of the constructed cover is $|D_1| < 2c \cdot |V| \cdot ln|\mathcal{C}|/t$, for some constant $c > 1$ (Awerbuch *et al.* [1]).

Selecting an element for a simple cover of $\mathcal{C}$ corresponds to selecting at most k elements for a k-fold cover. Thus, with probability of at least $1 - 1/|\mathcal{C}|^{c-1}$, a randomized k-fold cover D can be constructed for $\mathcal{C}$, and the expected size of D is $|D| < 2c \cdot |V| \cdot k \cdot ln|\mathcal{C}|/t$. □

Corollary 2 *If D is a k-fold cover constructed by Algorithm Cover2 for G, then the expected size of D is $|D| < 2 \cdot c \cdot n \cdot k \cdot ln\ n/t$, with probability of at least $1 - 1/n^{c-1}$, for some constant $c > 1$.*

Proof. For our collection $\mathcal{C}$, we have $|\mathcal{C}| = |V| = n$ (one t-set for each node). Thus the expected size of D is $|D| < 2 \cdot c \cdot n \cdot k \cdot ln\ n/t$, with probability of at least $1 - 1/n^{c-1}$. □

4 Comparing k-fold cover set with k-fold dominating set

From the definition of k-fold cover set it can simply be observed that if the degree of some node is smaller than k, the k-fold dominating set has no solution. Thus the degree of one node can compromise the existence of a k-fold cover set.

But then the k-fold cover set depends less on k and more on t, the number of nodes some node keeps track off. The value of t can vary, it is generally assumed to be more than k. Thus the k-fold cover set has the advantage that it can be defined for graphs in which the k-fold dominating set cannot be defined. The graph of Figure 1 has no 3-fold dominating set, since the degree of nodes $1, 2, 3, 4, 7$ is less than 3.

On the other hand, if D is a k-fold dominating set for G, then D is also a k-fold cover t-set for G, provided that $t \geq k$.

5 Conclusion

Selecting a small set of nodes among all the nodes in a network, and maintaining the routing infrastructure to only a subset of nodes reduces the routing overhead and excessive broadcast redundancy. Since nodes can maintain shortest paths to a small number of other nodes with limited memory, a simple cover set guarantees coverage, but not fault-tolerance. (A simple cover set guarantees that every node is "covered" by at least one pivot, the nearest one in terms of distance or weight.) By imposing that every node should be "covered" by at least k pivots, we guarantee a f-fault tolerant communication, where $f = min(k, c)$, c being the network connectivity.

An important difference between the k-fold MDS and k-fold cover is that we can extend the range of searching for a pivot. Pivots are not necessarily selected among the immediate neighbors, but from a larger neighborhood further away, thus allowing smaller subsets and a better change of stability when an event affects an entire region of nodes.

The relationship between the size of the k-fold cover set and k-fold dominating set is still open. We conjecture that the size of the k-fold cover set is no greater than the size of the k-fold dominating set.

References

[1] B. Awerbuch, A. Bar-Noy, N. Linial, and D. Peleg. Compact distributed data structures for adaptive routing. *Symposium on Theory of Computing (STOC)*, volume 2, pages 230–240, 1989.

[2] F. Dai and J. Wu. On constructing k-connected k-dominating set in wireless networks. *Proceedings of IEEE International Parallel & Distributed Processing Symposium (IPDPS)*, April 2005.

[3] F. Dai and J. Wu. On constructing k-connected k-dominating set in wireless ad hoc and sensor networks. *Journal of Parallel and Distributed Computing*, 66(7):947–958, 2006.

[4] J. E. Dunbar, J. W. Grossman, J. H. Hattingh, S. T. Hedetniemi, and A. A. McRae. On weakly-connected domination in graphs. *Discrete Mathematics*, volume 167/168, pages 261–269, 1997.

[5] T. Eilam, C. Gavoille, and D. Peleg. Compact routing schemes with low stretch factor. *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 11–20, 1998.

[6] C. Glaßer, S. Reith, and H. Vollmer. The complexity of base station positioning in cellular networks. *Discrete Applied Mathematics*, 148(1):1–12, 2005.

[7] I. Jaluta. Index cache consistency in client-server database systems. *Sixth IEEE International Conference on Computer and Information Technology (CIT 06)*, 0:9, 2006.

[8] J. Kratochvil. Problems discussed at the workshop on cycles and colourings. *univ.science.upsj.sk/c&c/rhistory/cc95prob.htm*, 1995.

[9] J. Kratochvil, P. Manuel, M. Miller, and A. Proskurowski. Disjoint and fold domination in graphs. *Australasian Journal of Combinatorics*, 18:277–292, 1998.

[10] F. Kuhn, T. Moscibroda, and R. Wattendorf. Fault-tolerant clustering in ad hoc and sensor networks. *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS 06)*, page 68, 2006.

[11] S. Kutten and D. Peleg. Fast distributed construction of small k-dominating sets and applications. *Journal of Algorithms*, pages 40–66, 1998.

[12] C. S. Liao and G. J. Change. k-tuple domination in graphs. *Information Processing Letters*, 87:45–50, 2003.

[13] L. Lovasz. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

[14] Z. Lu and K. S. McKinley. Partial collection replication versus caching for information retrieval systems. *Proceedings of the 23rd annual International ACM SIGIR conference on Research and development in information retrieval*, pages 248–255, New York, NY, USA, 2000. ACM Press.

[15] R. Mathar and T. Niessen. Optimum positioning of base stations for cellular radio networks. *Wireless Networks*, 6(6):421–428, 2000.

[16] D. Peleg. Distributed data structures: a complexity-oriented view. *Proceedings of the 4th International Workshop on Distributed Algorithms*, pages 71–89, 1990.

[17] L. D. Penso and V. C. Barbosa. A distributed algorithm to find k-dominating sets. *Discrete Applied Mathematics*, 141(1-3):243–253, 2004.

[18] V. Vazirani. *Approximation Algorithms*. Morgan Kaufmann Publishers, Springer, 2001.

[19] F. H. Wang, J. M. Chang, Y. L. Wang, and S. J. Huang. Distributed algorithms for finding the unique minimum distance dominating set in split-stars. *Journal of Parallel and Distributed Computing*, 63:481–487, 2003.

[20] R. Wittmann and M. Zitterbart. *Multicast Communication: Protocols and Applications*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2001.