

using k -dominating sets

Michael Q. Rieck

Drake University
Des Moines, Iowa 50311 USA
+1 515 271 3795
michael.rieck@drake.edu

Subhankar Dhar

San José State University
San José, CA 95192 USA
+1 408 924 3499
dhar_s@cob.sjsu.edu

Abstract—

For a connected graph, representing a sensor network, distributed algorithms for the Set Covering Problem can be employed to construct reasonably small subsets of the nodes, called k -SPR sets. Such a set can serve as a virtual backbone to facilitate shortest path routing, as introduced in [40], [12] and [13]. When employed in a hierarchical fashion, together with a hybrid (partly proactive, partly reactive) strategy, the k -SPR set methods become highly scalable, resulting in guaranteed shortest path routing with comparatively little overhead.

In this paper, we first discuss the notion of k -SPR sets, with the nodes of such a set functioning as routers for the network. These sets generalize our earlier k -SPR sets, which facilitated shortest path routing. We then introduce \mathcal{K} -SPR sequences that are used for hierarchical routing. We propose a distributed greedy algorithm for construction of \mathcal{K} -SPR sequences. The new sets facilitate minimal path routing, where “minimal path” here means “shortest weighted path based on edge weights”.

Finally, we introduce an efficient hybrid hierarchical routing strategy that is based on \mathcal{K} -SPR sequences. Our approach is unique in the sense that although dominating sets have been used to construct virtual backbones in ad hoc and sensor networks, this is the first attempt to use k -hop connected k -dominating sets for hierarchical routing that is also minimal path routing.

I. INTRODUCTION

Recent advances in micro-electro-mechanical systems (MEMS) and wireless research led to the development of sensor networks that show a lot of promise for future mobile applications [1]. Research efforts have been made to build low cost micro-sensors that possess processing capability as evidenced in the Smart Dust Project [21], [45], the PicoRadio Project [31] and WINS Project [32], [47]. A large number of wireless sensor networks consist of portable mobile devices with limited battery power. In order to address this limitation, energy-efficient routing algorithms and protocols are a major focus of current research.

In our work, we model sensor networks by a connected weighted graph having bidirectional links. For the sake of simplicity, the network nodes are presumed to be identical in nature and to have the same transmission radii. Edge weights are used as a measurement of the impact on the network of using a given link. These weights will be referred to as “costs”, and the exact details of how such costs are assigned will not be important in our discussion.

It will simply be understood that the higher the cost of a link, the less desirable it is to transmit using this link. Costs might be a function of the minimal transmission en-

ergy required for the link, and/or the relative impact on the battery levels of the nodes involved in the link. The minimal transmission energy is of course a function of the proximity of the two nodes, as well as any interference. The relative impact on a node’s battery energy level is additionally sensitive to the node’s current battery level. Ideally, the links at a node with a weak battery should all have a high cost.

Our contribution in this paper can be summarized as follows. We first discuss the notion of k -SPR sets, with the nodes of such a set functioning as routers for the network. These sets generalize our earlier k -SPR sets, which facilitated shortest path routing. We then introduce \mathcal{K} -SPR sequences that will be used for hierarchical routing. We propose a distributed greedy algorithm for construction of \mathcal{K} -SPR sequences. The new sets facilitate minimal path routing, where “minimal paths” in this paper means “shortest weighted path based on edge weights”.

II. OUR APPROACH

Our routing strategy is based on special k -dominating sets of nodes, namely k -SPR sets, that generalize similar sets from our earlier work ([12], [13], [14], [40], [39]). The nodes in such a set serve as “routers” and play a central role in facilitating route requests. Moreover, the nature of a k -SPR set is such that this guarantees minimal path routing under reasonable assumptions.

k -SPR sets can be used in a hierarchical way, based on an increasing finite sequence of numbers k_i , with one of these numbers corresponding to each of the levels of the hierarchy. This leads to an easily maintained and quite natural hybrid hierarchical routing strategy. It too guarantees minimal path routing. We supply detailed algorithms for forming such a hierarchy of k -SPR sets, which we call a \mathcal{K} -SPR sequence.

A reasonable choice for these numbers would be $k_i = k^i$, for some fixed integer $k \geq 2$. Since the largest k_i can be assumed not to exceed the diameter of G , the number of hierarchy levels in this case would be bounded by the logarithm of the diameter of G . Consequently, our hybrid routing strategy is highly scalable. Moreover, it is quite unique in its ability to also ensure minimal path routing.

Routing protocols for sensor networks are active areas of research and several researchers have proposed several protocols/heuristics in this regard. Here we only describe the ones that are closely related to our work.

Spin [17] is a flooding based protocol which has a broadcast system for data dissemination and works well for mobile sensors. It is also quite scalable and robust. The main drawback is nodes are always active and hence idle nodes consume energy.

The directed diffusion data dissemination protocol proposed in [16] uses the sink to send out interest, which is a task description, to all sensors. Each sensor node then stores the interest entry in its cache. The interest entry contains a timestamp field and several gradient fields. As the interest is propagated throughout the sensor network, the gradients from the source back to the sink are set up. When the source has data for the interest, the source sends the data along the interest's gradient path. The interest and data propagation and aggregation are determined locally. Also, the sink must refresh and reinforce the interest when it starts to receive data from the source. Note that the directed diffusion is based on data-centric routing where the sink broadcasts the interest. The main drawbacks of this protocol are (a) the gradient setup phase is expensive and (b) it is not energy aware as the best paths might be used too often.

A. Dominating-set-based routing

Since our framework for routing is based on minimum connected dominating set, we will here focus on only some of these, ones that are highly relevant to our own approach and that utilize a (k -)dominating set. The nodes in such a set provide a virtual backbone of router nodes, and in general, must be supplied with global routing information.

The Connected Dominating Set (CDS) problem is described as follows: find a minimal subset D of nodes, such that the subgraph induced by D is connected and D is a dominating set, *i.e.* it is a set in which each node is either in D or adjacent to some node in D . Finding a minimum connected dominating set is known to be an NP-complete problem [15]. Several authors have proposed algorithms for obtaining approximately minimal connected dominating sets [15].

Span [6] is one of several ad hoc networking protocols based on the notion of a dominating set. In Span, "coordinators" - a group of nodes that form a connected dominating set over the network - do not sleep. Non-coordinator nodes follow a synchronized sleep/wake cycle, exchanging traffic using an algorithm based on the beaconing and traffic announcement methods of IEEE 802.11 IBSS power save. The routing protocol is integrated with the coordinator mechanism so that only coordinators forward packets, acting as a low latency routing backbone for network. Span is intended to maximize the amount of time nodes spend in the sleep state, while minimizing the impact of energy management on latency and capacity.

The algorithm of J. Wu and H. Li is a distributed algorithm [51] that is used to construct a connected dominating set in a connected graph of radius at least two. The set produced by their algorithm is used to form a virtual backbone of a wireless ad hoc network. Later, J. Wu, F. Dai, M. Gao and I. Stojmenovic [49] further extended the work of Wu and Li and proposed an algorithm for creating power aware connected dominating sets. Basically, the idea is to select nodes with higher energy level. When selecting a node in the dominating set, if there is a tie, they use the node ID to break the tie. They demonstrated that their method of selection of nodes increases the lifetime of the network.

In [40], the authors generalized the Wu-Li algorithm so as to produce a k -hop connected k -dominating set that work as routers. (See Section IV for definitions.) One of the important aspect of their routing scheme was that it also guaranteed shortest path routing through the network along a path that was guaranteed at any point along the way, to encounter another router node within every k steps. Later the authors modified this algorithm and proposed a number of variations on it [12], [13]. These were largely motivated by the following study into k -hop dominating sets.

In [24], B. Liang and Z. J. Haas proposed a distributed greedy algorithm to produce a small k -dominating set. In order to do so, they reduced the problem to a special case of the Set Covering Problem. A similar but different reduction to this problem was also used in [12]. For a given value of k , though, the latter requires fewer steps than the Liang-Haas method. In addition it produces a set that is not only k -dominating, but is also k -hop connected, and has a special property to facilitate shortest path routing.

B. Hierarchical routing

Hierarchical routing has gained special attention for sensor networks for their scalability and flexibility. Several hierarchical routing strategies for ad hoc networks based on dominating sets are discussed in [9]. In order to orchestrate hierarchical routing, various clustering algorithms have been developed for this purpose. Both deterministic and probabilistic distributed algorithms are used to construct clusters. The average communication costs in construction and maintenance of such hierarchies is shown to be logarithmic with the network size. However, all these clustering strategies do not guarantee shortest path routing.

LEACH: Low-energy adaptive clustering hierarchy (LEACH) is a hierarchical-based protocol that minimizes energy dissipation in sensor networks [18]. The purpose of LEACH is to randomly select sensor nodes as cluster-heads, so the high energy dissipation in communicating with the base station is spread to all sensor nodes in the sensor network. Clusterhead selection is difficult to optimize in many situations.

PEGASIS & Hierarchical-PEGASIS: The Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [27] is another hierarchical protocol that is an improvement of the LEACH protocol. As opposed to forming clusters like

LEACH, PEGASIS first constructs chains consisting of sensor nodes so that each node transmits and receives from a neighbor and only one node is selected from that chain to transmit to the base station (sink). The data collected in this manner then moves from node to node, aggregated and eventually sent to the base station. The chain formation is done in a greedy way.

Performance evaluation of PEGASIS indicates that it outperforms LEACH for different network sizes and topologies. However, one of the major drawback of PEGASIS is that it introduces excessive delay for distant node on the chain. Moreover, the single node acting as a leader of the chain can sometimes become a bottleneck.

Hierarchical-PEGASIS [28], which is an extension of PEGASIS, is designed to address the delay incurred for packets during transmission to the base station. In order to improve the performance by reducing the delay in PEGASIS, messages are transmitted simultaneously. There are two approaches to avoid collisions and possible signal interference among the sensors. The first one uses CDMA type signal coding techniques. The protocol with nodes having CDMA capability, constructs a chain of nodes and forms a tree like hierarchy. Each selected node in a particular level transmits data to the node in the upper level of the hierarchy. This method guarantees that data is transmitted in parallel and reduces the delay significantly. Since the tree constructed in this manner is balanced, the delay will be in $O(\log N)$ where N is the number of nodes. The other approach (non-CDMA based) is quite different and allows only spatially separated nodes to transmit at the same time. Using this approach, a three-level hierarchy of the nodes is created first. The effects of interference is reduced by carefully scheduling simultaneous transmissions. The approach taken by PEGASIS avoids the clustering overhead of LEACH, but still requires dynamic topology adjustment since sensor's energy is not tracked. Hence every sensor must have the knowledge of its neighbor so that it knows where to route that data. So the topology needs adjustment and that can introduce significant overhead especially for networks with heavy traffic.

TEEN and APTEEN: Threshold sensitive Energy Efficient sensor Network protocol (TEEN) [30] is another hierarchical protocol coupled with data centric approach that is designed to react to sudden changes in the sensing attributes such as weather conditions. The approach here is more proactive rather than reactive. This is particularly important for time-sensitive applications. The sensor network architecture is based on a hierarchical grouping where closer nodes form clusters and this process goes on the second level until base station (sink) is reached.

After the clusters are formed, the cluster head broadcasts the thresholds based on certain sensed attributes to the nodes. Once a node senses a value that exceeds the threshold value, it transmits data. However, TEEN is not suitable for applications where periodic data are required since the user may not get any data at all if the thresholds are not reached.

The Adaptive Threshold sensitive Energy Efficient sen-

sor Network protocol (APTEEN) [30] is a further extension of TEEN and designed to capture both periodic data collections and reacting to time-critical events. The architecture is similar to TEEN. First the base station forms the clusters. Then the cluster heads broadcast the attributes, the threshold values, and the transmission schedule to all nodes. In addition, the cluster heads also perform data aggregation. Performance evaluation of TEEN and APTEEN has indicated that they outperform LEACH [18]. The main drawbacks are the overhead and complexity of forming clusters in multiple levels, implementing threshold-based functions and dealing with attribute-based naming of queries.

IV. k -SPR SETS AND \mathcal{K} -SPR SEQUENCES

The k -SPR sets to be presented are a straightforward generalization of the k -SPR sets defined in [12] (where they are called “ d -SPR sets”) and essentially introduced in [40]. The generalization is for the purpose of handling graphs that are equipped with link weights. After a discussion of k -SPR sets, sequences of such will be considered and ultimately used to facilitate hierarchical routing. Throughout this discussion, G will denote a finite connected graph representing a sensor network, with positive link weights referred to as “costs”.

A. Basic definitions and a relationship between these

Given a path in G , the *cost* of the path is the sum of the costs of the links along the path. Given two nodes, u and v , the *cost* $c(u, v)$ between these is the minimum of the costs of the paths connecting these two nodes. A path from u to v is said to be a *minimal path* if its cost is $c(u, v)$. The *radius* of G is the largest number $R \geq 0$ such that for each node u , there exists a node v satisfying $c(u, v) \geq R$. Let V denote the set of nodes of G . Let $N = |V|$.

Some fundamental definitions concerning subsets of V and claims about these required for the routing strategy to be described in the next section will now be presented.

Definition 1: Fix a positive number k . Fix a subset S of the set of nodes in V .

- (a) S is *k -dominating* if every node in V is within a cost k of some node in S .
- (b) S is *k -hop connected* if, given any two nodes u and v in S , there is a path in G from u to v such that the cost between consecutive elements of S along this path never exceeds k .
- (c) S is a *k -SPR set* if, given any two nodes u and v in V satisfying $c(u, v) > k$, there exists some node w in S such that $w \neq u$, $w \neq v$, and $c(u, w) + c(w, v) = c(u, v)$.

The definition of a k -SPR set was formally introduced in [12], and is a central concept in [40] as well. It essentially means that whenever two nodes are sufficiently far apart, there is certain to be at least one node from the k -SPR set lying between them along a minimal path. The three types of subsets of V are related via the following facts, which

generalized [12, Theorem 1]:

Theorem 1: Assume that S is a k -SPR set for G . Then the following are true.

- (a) Given any two nodes u and v of G , there exists a minimal path connecting u to v such that the set of nodes along this path that are also in $S \cup \{u, v\}$ is k -hop connected.
- (b) S is k -hop connected.
- (c) If the radius of G exceeds k , then S is k -dominating.

Proof: The first claim will be proved by contradiction, so assume that it is false. Suppose that $\lambda \geq 0$ is as small as possible such that there exist nodes u and v with $c(u, v) = \lambda$, but there is no path connecting u to v as described in part (a) of the theorem. Clearly $\lambda > k$. Let p be any minimal path from u to v . So the cost of p is λ . Let w be the first node along p whose cost from u exceeds k . Since $c(u, w) > k$, there is a node u' in S , not equal to u or w , such that $c(u, u') + c(u', w) = c(u, w)$. Now, $c(u, v) \leq c(u, u') + c(u', v) \leq c(u, u') + c(u', w) + c(w, v) = c(u, w) + c(w, v) = c(u, v)$, and hence $c(u, v) = c(u, u') + c(u', v)$, so that u' must lie along a minimal path connecting u and v .

Since $c(u', v) < \lambda$, by assumption there is a minimal path p' from u' to v , with the property described in the theorem (with u' in place of u). Since $u' \neq w$, $c(u, u') \leq k$. So the concatenation of a minimal path from u to u' followed by p' is certainly a minimal path from u to v that has the property stated in the theorem, and so yields a contradiction. This establishes the first claim.

By choosing u and v arbitrarily from S , it now follows immediately from part (a) that S is k -hop connected. To show that it is k -dominating, consider any node x . Assuming now that G has radius exceeding k , there exists a node y with $c(x, y) > k$. Assume that y is as close to x as possible with this property. Since S is a k -SPR set, it contains a node z with $c(x, z) + c(z, y) = c(x, y)$, and z is not equal to y . It follows that $c(x, z) \leq k$. Therefore, S is k -dominating. ■

B. Local views

When G represents an ad hoc network, [40], [12] and [13] produce a k -SPR set to serve as a virtual backbone for routing purposes. To achieve practical distributed algorithms for finding such a k -SPR set, the following subgraphs of G need to be considered. These generalize similar subgraphs in [12] and [13], but the terminology is altered slightly. A “ $(d + 1)$ -local view” there is called an “extended d -local view” here.

Definition 2: Let v be a node of V . Let $r \geq 0$. The r -local view of v is the subgraph induced by all of the nodes within a cost r of v . The extended r -local view of v is the subgraph of G obtained by extending the r -local view of v by including also any nodes at a cost greater than r from v that are adjacent to a node in the r -local view, plus the links that realize these adjacencies.

It is clear that the cost from v to another node u in v 's r -local view is also the cost between these nodes in G , that is, $c(v, u)$. We will suppose that nodes employ some sort of “extended hello” messages in order that each node be able to learn about its extended r -local view, for some r . It is important for the purposes of shortest path routing to know when the cost between two nodes in some extended r -local view agrees with the corresponding cost in the graph G as a whole. This issue is partly addressed in the first part of [12, Theorem 2]. A somewhat more general claim is the following.

Lemma 1: Let x and y be in the extended r -local view of v . Let c' denote the cost between x and y as measured in this r -local view. Then the following claims are valid:

- (a) If $c(v, x) + c(v, y) + c(x, y) \leq 2r$, then $c' = c(x, y)$.
- (b) If $c(v, x) + c(v, y) + c' \leq 2r$, then $c' = c(x, y)$.

Proof: First assume that $c(v, x) + c(v, y) + c(x, y) \leq 2r$. Consider a closed path beginning at v that follows a minimal path to x , then follows a minimal path p from x to y , and then follows a minimal path from y to v . The cost of this closed path does not exceed $2r$, and hence every node on it is within a cost (in G) r of v . So a minimal path from x to y is entirely contained within the r -local view of v . Thus p is contained in the r -local view of v . So $c' = c(x, y)$.

Next assume instead that $c(v, x) + c(v, y) + c' \leq 2r$. Clearly $c(x, y) \leq c'$, so that this situation reduces to the prior one, and again it may be concluded that $c' = c(x, y)$. ■

C. A covering problem

Another common feature of the routing algorithms to be considered is that they all rely on a bipartite graph $B = B(G)$, based on G , a portion of which is maintained in a data structure by each network node. The bipartite graph B is described as follows.

Definition 3: The nodes of the bipartite graph $B = B(G)$ constitute two sets V and P , each of which is an independent set in B . V is simply the set of all nodes of G . The elements of P are certain unordered pairs of nodes $\{x, y\}$ of G . To describe which, first consider the set \hat{P} of all such pairs satisfying $c(x, y) > k$. Partially order \hat{P} by taking $\{x', y'\} \leq \{x, y\}$ if (after possibly reordering x' and y') $c(x, x') + c(x', y') + c(y', y) = c(x, y)$. (This means that x' and y' lie along some minimal path connecting x and y .) Now P is defined to be the subset of \hat{P} consisting of the minimal elements with respect to this partial order. The description of the bipartite graph B is completed by indicating that $v \in V$ is taken to be adjacent to $\{x, y\} \in P$ if and only if $c(x, v) + c(v, y) = c(x, y)$, but $v \neq x$ and $v \neq y$.

When all the link costs are one, B is the same as the bipartite graph considered in [12]. The following claim is straightforward to check using Definition 2 and part (c) of

Definition 1.

Theorem 2: A subset S of V is a k -SPR set for G if and only if every element of P is adjacent in B to some element of S .

When this adjacency condition holds, we say that S covers P . The second part of [12, Theorem 2] may now be generalized to produce the following needed fact.

Lemma 2: Let e be an upper bound on the link costs of G . Suppose that $u, v \in V$ both cover the pair $\{x, y\} \in P$. Then $c(u, v) \leq k + e$.

Proof: $2c(u, v) \leq c(u, x) + c(x, v) + c(u, y) + c(y, v) = 2c(x, y)$. So $c(u, v) \leq c(x, y) \leq k + e$. ■

D. \mathcal{K} -SPR sequences

The constructs presented in this subsection anticipates the hierarchical nature of the routing strategy to be introduced in the next section. Given a k -SPR set, it will be helpful to consider the following derived link-weighted graph.

Definition 4: Let S be a k -SPR set for G . Define the link-weighted graph $G[S, k]$ as follows. The node set for the graph $G[S, k]$ is S . Two elements u and v of S are made adjacent in $G[S, k]$ if $c(u, v) \leq k$ (in G). In this case, the link connecting u and v in $G[S, k]$ is assigned the cost $c(u, v)$.

By part (b) of Theorem 1, this graph is connected. Moreover, the cost between any two nodes in $G[S, k]$ when measured in this graph agrees with the cost between them when measured in G . To accommodate a hierarchical version of k -SPR routing, this derived graph notion will now be used to introduce a generalization of the notion of a k -SPR set.

Definition 5: Fix a set of positive numbers $\mathcal{K} = \{k_1, \dots, k_l\}$ with $k_1 < k_2 < \dots < k_l$. A \mathcal{K} -SPR sequence for G is a collection $\mathcal{S} = \{V_1, \dots, V_l\}$ of sets of nodes of G with the following property. Letting $G_0 = G$ and $V_0 = V$, and letting G_i denote $G_{i-1}[V_i, k_i]$ for $i = 1, 2, \dots, l$, the set V_i is required to be a k_i -SPR set for the graph G_{i-1} , for $i = 1, 2, \dots, l$. The following numbers will also be needed. Let $r_0 = k_1$ and for $i > 0$, let $r_i = k_{i+1} + 2k_i + \dots + 2k_1$.

Thus $V = V_0 \supseteq V_1 \supseteq \dots \supseteq V_l$. Part (a) of Theorem 1 now generalizes as follows.

Theorem 3: Let $\mathcal{K} = \{k_1, \dots, k_l\}$ be a set of positive numbers with $k_1 < k_2 < \dots < k_l$. Let $\mathcal{S} = \{V_1, \dots, V_l\}$ be a \mathcal{K} -SPR sequence for G . Given any two nodes u and v in V , there exists a minimal path p connecting u to v such

that, for $i = 1, 2, \dots, l$, the set of nodes consisting of the all the nodes along p and belonging to V_i , together with the first and last nodes along p and belonging to V_{i-1} , form a k_i -hop connected set for G . Moreover, V_i is a r_{i-1} -SPR set for G , for $i = 1, 2, \dots, l$.

Proof: If $k = 1$, then this reduces to part (a) of Theorem 1 since V_1 is by definition a k_1 -SPR set for G , and the minimal path p guaranteed by part (a) of Theorem 1 works here. u and v would be the first and last nodes along it that belong to V_0 . The argument for the general case can now be made using induction on k . To this end, fix some $k > 1$. We will show that the claim holds for any \mathcal{S} of cost (size) k , assuming that it holds for shorter k -SPR sequences. Fix $u, v \in V$. Let h be a minimal path from u to v that works as described in part (a) of Theorem 1 using V_1 for S . If h contains any nodes from V_1 , then let u' and v' denote the first and last such nodes, respectively. If no such nodes exist or if $u' = v'$, nothing more needs to be shown.

Otherwise, let $G_1 = G[V_1, k_1]$. By the induction hypothesis, applied to G_1 , there exists a minimal path in G_1 from u' to v' as described in Theorem 3, using the $\{k_2, \dots, k_l\}$ -SPR sequence $\{V_2, \dots, V_l\}$ in place of the \mathcal{K} -SPR sequence \mathcal{S} . This minimal path can be realized as a minimal path g' from u' to v' in G by replacing each link of the minimal path in G_1 with a minimal path connecting its endpoints in G . Now define p to be the minimal path from u to v that first follows an initial portion of h from u to u' , then follows g' from u' to v' , and then follows a final portion of h from v' to v . It is now straightforward to see that p fulfills the requirements of Theorem 3. The initial portion and final portion of h here each have cost at most k_1 . The fact that V_i is an r_{i-1} -SPR set for G now follows by induction as well. ■

E. An example

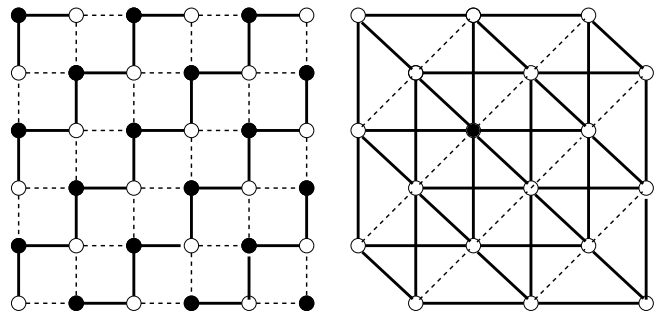


Fig. 1. G and G_1

Consider the following example using $k_1 = 3$ and $k_2 = 9$. The graph on the left in Figure 1 is the original graph G . The dashed edges have cost one, while the solid edges have cost two. The dark vertices form a 3-SPR set V_1 for G . The graph on the right is then $G_1 = G[V_1, 3]$. It has two types of edges. The dashed edges have cost two, while the solid edges have cost three. Here V_2 consists of the lone dark vertex in the figure. This is a 9-SPR for G_1 . Thus

$G_2(= G_1[V_2, 9])$ would consist only of one vertex, and the process terminates.

Now, in Theorem 3, consider the case where u and v are the top-left node and bottom-right node of G , respectively. There are several minimal paths connecting u and v , and we see that their cost is 15. One of these path starts at u , and repeatedly moves down one hop and then right one hop, zigzagging until arriving at v . Call this path p . Notice that it goes through the only node in V_2 , which we'll call w . Consider the claim in Theorem 3 when $i = 2$. The first and last nodes along p that belong to V_1 are u and v . The fact that $\{u, v, w\}$ is 9-hop connected in G gives evidence in support of Theorem 3.

Let's try a different choice for u and v , say by taking these to be the top-right node and the bottom-left node, respectively. Now the cost between u and v is only 10 and there is an evident unique minimal path connecting them. Let p now denote this path, which uses only edges of cost one, and which alternates between nodes in V_1 and nodes not in V_1 . Letting x and y denote the first and last nodes along the path that belong to V_1 , we see that $c(x, y) = 8$. There are no nodes from V_2 along p . So using $i = 2$ again, we now notice that $\{x, y\}$ is 9-hop connected in G , as required.

V. HIERARCHICAL ROUTING VIA \mathcal{K} -SPR SEQUENCES

A. Establishing a \mathcal{K} -SPR sequence

Let k_0 be an upper bound on the link costs of G . Let $\mathcal{K} = \{k_1, \dots, k_l\}$ be a set of positive numbers satisfying $k_0 < k_1 < \dots < k_l$. The distributed algorithms of [40], [39] [12] and [13], [14] can now be altered to handle graphs with weighted links. By iteratively applying such an algorithm, it then becomes straightforward to obtain a \mathcal{K} -SPR sequence for G . Once this has been accomplished, the routing strategies described in the next section can be implemented.

For example, the greedy algorithm approach in [12] is easily adapted to handle a graph with link costs, as will now be outlined. The following algorithm shows how this would proceed at level i , that is, when applied to the graph G_i in order to find a k_{i+1} -SPR set for it. Note however that when $i > 0$, the processing at level i begins locally only after processing at level $i - 1$ has completed locally. The distributed greedy algorithm used here, at each level, does not require strict synchronization though.

Each node in the network has a unique ID number. Each node that becomes a level- i node (element of V_i) begins participating in the process of selecting level- $(i + 1)$ nodes (elements of V_{i+1}). Initially it is in the "undecided" state, but ultimately ends up in either the "selected" or "not selected" state after completing the algorithm. The selected nodes are of course the level- i nodes that are selected to become level- $(i + 1)$ nodes, that is, the nodes of G_{i+1} . The distributed greedy algorithm is as follows.

Distributed greedy algorithm

Step 1: Each node $v \in V_i$ gathers information about its r_i -local view of G_i , which will henceforth be referred to as v 's *level- i view*. This requires several rounds of passing local link-state information. Some nodes in this local view may still be actively participating in the greedy algorithm at a lower level. If this happens, then the level- i algorithm must stall until these nodes complete the lower level algorithms.

Step 2: v determines P_v and C_v , where these are defined as follows. P_v denotes the set of all the nodes pairs $\{x, y\}$ covered by v in the bipartite graph B . C_v denotes the set of all the nodes that cover some node pair in P_v . ($v \in C_v$, and by Lemma 2, v is able to "see" the elements of P_v and C_v . Actually, only a $(k_{i+1} + k_i)$ -local view is required for this.) v also computes its current *covering number* $|C_v|$ (the size of C_v).

Step 3: v multi-casts a message containing its covering number and its status (undecided, selected or not selected) to each node in C_v . (Note that the first time this step is executed, v is undecided, and the last time it executes this step, it will be in one of the two decided states.)

Step 4: If v has entered one of the two decided states (selected or not selected), then it essentially terminates its participation in this algorithm (at the current level), except to help route messages between other nodes. Otherwise, if it is still undecided, then

Step 5: v waits until it receives messages as in Step 3 from each node in C_v . For each such node u that has become decided, v removes u from C_v , and if u has become selected, then v also removes any pairs from P_v that u covers. Accordingly, v recomputes its covering number as necessary.

Step 6: If v 's covering number is now zero, then v enters the "not selected" state, and loops back to Step 3. Otherwise....

Step 7: v checks to see if its own *priority* is the highest among all the nodes of C_v . Priority here is defined to be the ordered pair (**covering_number**, **ID**), lexicographically ordered (as in [19, Subsection 2.1]). If v has the highest priority, then v enters the "selected" state. In either case, it loops back to Step 3.

Remarks:

1. Once a selected node has terminated the greedy algorithm at level i , it can proceed to initiate its participation in the greedy algorithm at level $i + 1$, where it is of course initially undecided at this level.
2. In Step 3, a node v is obliged to send a message to some of the nodes in its level- i view. This can be handled efficiently by means of "optimal routing trees" and lower level local routing.
3. It is also possible to let the ultimate number of levels be initially unspecified, perhaps until a level is reached consisting of a single node. The set \mathcal{K} would then grow according to some formula, as new levels are constructed.

Other algorithms can be used in place of the greedy algorithm. For example, the probabilistic methods in [35] and [19] can easily be adapted to handle the covering problem associated with finding a k -SPR set, as opposed to the problem of just finding a k -dominating set. Also, it is possible to adapt the “ d -SPR-C method” of [12]. Unlike the greedy algorithm, and assuming that link costs reflect transmission time delays, this algorithm completes in a time period that does not depend on the overall size of the network, but rather only depends on the maximum link cost and maximum node degree. The algorithm is as follows:

A faster algorithm

Step 1: Each node v gathers information about its r_i -local view as in the greedy algorithm.

Step 2: v determines P_v and C_v as in the greedy algorithm.

Step 3: v sends its covering number $|C_v|$ to each member of C_v .

Step 4: If v observes that there is a pair $\{x, y\} \in P_v$ such that v has the highest priority (see Step 7 of the greedy algorithm) among all the nodes that cover $\{x, y\}$, then v becomes “selected”. Otherwise, it becomes “unselected”.

This algorithm clearly terminates after the four steps, and produces a k_{i+1} -SPR set by Theorem 3. In general though, this set is substantially bigger than the one produced by the greedy algorithm.

B. Local unicast routing at a given level

Once a \mathcal{K} -SPR sequence has been established up to some level, say i , it is possible for a level- i node v to efficiently route a message to another level- i node u within its level- i view, as follows. Recall that if $j < i$, a level- i node is also a level- j node. Now v can easily discover a minimal path in the level- i view connecting it to u . Let u_{i-1} denote the first node on this minimal path after v . Since $c(v, u_{i-1}) \leq k_i$, the node u_{i-1} is visible to v in its level- $(i-1)$ view. It can then find a minimal path connecting itself to u_{i-1} at this level. Let u_{i-2} be the node after v on this minimal path. And so forth, down to level zero.

Letting $u_i = u$, v can append the sequence $\{u_j\}_{j=1}^i$ as routing information to the message, before sending it to its neighbor (in G) u_0 . The level-zero views of the nodes along the way now aid to easily route the message to u_1 . By similar reasoning, requiring both level-one and level-zero views, the message can then be delivered to u_2 . And so forth, until it ultimately arrives at u . Moreover, the path (in G) used to route the message from v to u is guaranteed to be a minimal path.

C. Special multicasting to routers

We now consider a very specific multicasting problem for a network with an established \mathcal{K} -SPR sequence. This will

be employed for both the proactive and reactive aspects of the hybrid routing scheme proposed in the next subsection. We will need the following definition and lemma.

Definition 6: Consider an arbitrary node v . For $i \geq 1$, a level- i node v_i will be called a *level- i router for v* if the only level- i node u satisfying $c(v, u) + c(u, v_i) = c(v, v_i)$ is $u = v_i$.

Thus a level- i router for v is a level- i node such that any shortest path connecting it to v contains no other level- i nodes.

Lemma 3: A level- i router v_i for v satisfies $c(v, v_i) \leq k_1 + k_2 + \dots + k_i$.

Proof:

The claim can be seen by induction on i . Since the level-one nodes form a k_1 -SPR set, it is clear that if v_1 is a level-one router for v , then $c(v, v_1) \leq k_1$. Consider next a level-two router v_2 for v . Along a minimal path from v to v_2 , a level-one node v_1 (other than v_2) might occur. By altering the minimal path if necessary, it may be assumed that v_1 is a level-one router for v . Besides v_2 , the path cannot contain any other level-two nodes. But v_1 and v_2 and both in the graph $G[V_1, k_1]$ for which V_2 is a k_2 -SPR set. Here V_1 and V_2 denote the sets of level-one and level-two sets, respectively. Thus, $c(v_1, v_2) \leq k_2$. Since $c(v, v_1) \leq k_1$, we get $c(v, v_2) \leq k_1 + k_2$. And so forth. ■

The goal now is to allow v to send a message to all of its routers, at all levels. In fact, this goal will be accomplished in such a way that forwarded messages always move along minimal paths, moving away from the source node v . Moreover, there will be no redundancy in the message forwarding, in the sense that no node will receive more than one copy of the message. That is, the message will move along a tree rooted at v , and each path from v in this tree will be a minimal path. This sort of “multicasting to routers” will provide a basis for the hybrid routing scheme described in the next subsection.

To manage the proposed multicasting, it is necessary for a level- i router v_i for v that receives the message along a given minimal path, to decide to which of the level- $(i+1)$ routers for v it must forward the message. As a technical detail, in order for v_i to make this decision, it will be necessary that a list of all the level- i routers for v , along with their costs from v , be included in the header of the message that v_i receives. Under reasonable conditions, this list will not be large. Before v_i forwards the message to level- $(i+1)$ the routers, it will likewise be necessary for it to append a list of all the level- $(i+1)$ routers for v , and their costs from v . However, the level- i router information can be removed from the header at this point.

Now v_i is within a cost $k_1 + \dots + k_i$ of v , as are all of the level- i routers for v . Moreover, v_i has received a list of these together with their costs from v . Let u denote one such

level- i router. Consider a level- $(i+1)$ node w within a cost k_{i+1} of u . Such a node is potentially a level- $(i+1)$ router for v , and all level- $(i+1)$ routers for v fit this description for some u . Now, with w fixed, it turns out that v_i is able to determine which level- i routers u lie along a minimal path in G connecting v to w . In the first place, w is in the level- i view of v_i , which can be seen by considering a shortest possible path from v_i to v , and then to u , and then to w . The cost of this does not exceed r_i , so $c(v_i, w) \leq r_i$. Also, $c(v_i, u) \leq 2(k_1 + \dots + k_i)$ and $c(u, w) \leq k_{i+1}$. It follows by Lemma 2 that v_i is able to correctly compute $c(u, w)$, using its level- i view (of level- i nodes within a cost r_i). It is now straightforward to see that v_i is able to determine whether or not w is a level- $(i+1)$ router for v . If it is, then v_i is also able to determine any level- i routers for v that lie along a minimal path connecting v and w .

There is one last detail. In order to avoid redundant messages, for each level- $(i+1)$ router v_{i+1} for v , exactly one of the level- i routers for v lying between v and v_{i+1} along a minimal path should be selected to forward the message to v_{i+1} . Each of these routers is aware of the others and so some criterion can be used that they will all agree on in order to make the selection. For example, this decision could be made by using a simple criterion such as choosing the level- i router for v with the largest ID.

D. A hybrid hierarchical routing strategy

The routing strategy that will be developed here has the following theorem as its foundation.

Theorem 4: Given any two nodes u and v , there exists a minimal path p connecting u and v , and a positive integer i , such that p contains a level- i router u_i for u and a level- i router v_i for v with $c(u_i, v_i) \leq c(u, v) \leq k_{i+1} \leq r_i$.

Proof:

Let i be as large as possible such that $c(u, v) > k_i$. Some minimal path connecting u and v must contain a level- i node. By altering the initial and terminal portions of this path as necessary, it is straightforward to see that some minimal path connecting u and v must contain a level- i router u_i for u as well as a level- i router v_i for v . Now $c(u_i, v_i) \leq c(u, v) \leq k_{i+1} \leq r_i$. ■

During the process of establishing a \mathcal{K} -SPR sequence in a sensor network, say by the greedy algorithm method, it is easy to arrange for each node v to be known to all of the nodes within a cost k_1 , as well as all of the level-one nodes that are within a cost k_2 of a level-one node that is within a cost k_1 of v , as well as all of the level-two nodes that are within a cost k_3 of a level-two node that is within a cost k_2 of a level-one node that is within a cost k_1 of v , and so forth. In fact, this does not require any additional messages, but rather only the inclusion of more information in the already required selection overhead messages.

It may be assumed that in this way each level- i router v_i of v maintains a list of nodes $\{v = v_0, v_1, v_2, \dots, v_i\}$ with the property that there exists a minimal path in G connecting v to v_i such that v_j is a level- j router for v ($j = 1, \dots, i$). In addition, all level- i nodes within a cost k_{i+1} of one of the level- i routers v_i of v will be made aware of v , and we may assume that these too have been provided with routing information to v . If the network is allowed to change dynamically, then any new node that joins the network later would be obliged to announce itself to its routers and to each level- i node within a cost k_{i+1} of one of its level- i routers. This could be managed using a variation of the multicasting to routers method discussed in the previous subsection.

Now, after establishing the \mathcal{K} -SPR sequence and the above routing information, suppose that a node u has a need to contact a node v , say to establish a virtual circuit in order to conduct an extended conversation with v . Suppose too that u is currently unaware of where v is in the network, and so has no routing information concerning it, other than the ID number of v or some other identifier such as a unique name. In particular, this would mean that $c(u, v) > k_1 = r_0$.

As a result of Theorem 3 and the assumptions we are making about the local information maintained by each node, at each level, the node u is able to find the node v as follows. u multicasts a request message to its routers, as described in the previous subsection. Eventually some node receiving the request will know about the existence of v , and will know a shortest path to it. This node can then reply by relaying this information back to u along with the information that describes a minimal path from itself to u . It does not need to forward the message to higher level routers. In this way, u learns a path to v , as well as its cost. At least one of the paths thus discovered will be a minimal path from u to v .

Point-to-point communication between u and v can now be effected via routing information placed in the header. However, this only needs to involve a sequence $u = u_0, u_1, \dots, u_i, v_i, \dots, v_1, v_0 = v$ of nodes, where u_j and v_j are level- j routers for u and v , respectively ($j = 1, \dots, i$) and $c(u_i, v_i) \leq k_{i+1}$. The routing between these nodes can be managed by means of the appropriate local views of the various nodes along the way.

VI. CONCLUSION

In this paper, we first discuss the notion of k -SPR sets, with the nodes of such a set functioning as routers for the network. These sets generalize our earlier k -SPR sets, which facilitated shortest path routing. We then introduce k -SPR sequences that are used for hierarchical routing. We propose a distributed greedy algorithm for construction of k -SPR sequences. The new sets facilitate minimal path routing, where “minimal paths” here means “shortest weighted path based on edge weights”.

Finally, we introduce an efficient hybrid hierarchical routing strategy that is based on our approach of building k -SPR sequences. When the k -SPR sequence used

consists of successive power of a fixed base (a reasonable choice), then the number of levels in the hierarchy is logarithmic dependent on the diameter of the network graph. This makes the strategy highly scalable. Our approach is unique in the sense that although dominating sets have been used to construct virtual backbones in ad hoc and sensor networks, this is the first attempt to use k -hop connected k -dominating sets for hierarchical routing that is also minimal path routing.

REFERENCES

- [1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, (2002) Wireless Sensor Networks: A Survey, Computer Networks, Vol 38, pp 393-422.
- [2] K.M. Alzoubi, P. Wan, O. Frieder, New Distributed Algorithm for Connected Dominating Set in Wireless Ad Hoc Networks, *Proc. 35th Hawaii Int. Conf. System Sciences*, (2002).
- [3] A.D. Amis, R. Prakash, T.H.P. Vuong, D.T. Huynh, Max-Min D-Cluster Formation in Wireless Ad Hoc Networks, *Proc. IEEE INFOCOM*, (2000).
- [4] S. Banerjee, S. Khuller, A clustering scheme for hierarchical routing in wireless networks, *IEEE Infocom 2001*, Anchorage, Alaska, April 2001.
- [5] C. Basile, M.-O. Killijian, D. Powell, A survey of dependability issues in mobile wireless networks, Tech. Report, LAAS CNRS Toulouse, France (2003).
- [6] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks", *Proc. Mobicom 2001*, pp. 85-96.
- [7] G. Chen, I. Stojmenovic, Clustering and routing in mobile wireless networks, Tech. Report TR-99-05, Dept. Computer Sci., SITE, U. Ottawa (1999)
- [8] Y. P. Chen, A. L. Liestman, Approximating Minimum Size Weakly-Connected Dominating Sets for Clustering Mobile Ad Hoc Networks, *Third ACM Int. Symp. Mobile Ad Hoc Networking and Computing (MobiHoc 02)*, 157-164 (2002).
- [9] Y. P. Chen, A. L. Liestman, J. Liu, Clustering Algorithms for Ad Hoc Wireless Networks, in *Ad Hoc and Sensor Networks*, Edited by Y. Xiao and Y. Pan, Nova Science Publisher, 2004.
- [10] V. Chvátal, A greedy heuristic for the set-covering problem, *Math. Operations Research* 4:233-235 (1979).
- [11] B. Das, V. Bharghavan, Routing in Ad-hoc networks using minimum connected dominating sets, *IEEE Int. Conf. Communications (ICC '97)* 1:376-380 (1997)
- [12] S. Dhar, M. Q. Rieck and S. Pai. On Shortest Path Routing Schemes for Wireless Ad Hoc Networks. *Proceedings of the 10th International Conference on High Performance Computing (HiPC '03)*, Lecture Notes in Computer Science, (LNCS 2913), Hyderabad, Pages 130-141, Springer Verlag (2003).
- [13] S. Dhar, M. Q. Rieck, S. Pai and E. J. Kim. Various Distributed Shortest Path Routing Strategies for Wireless Ad Hoc Networks, *Proceedings of the 5th International Workshop on Distributed Computing (IWDC 2003)*, Lecture Notes in Computer Science, (LNCS 2918), Calcutta, Pages 255-264, Springer Verlag (2003).
- [14] S. Dhar, M. Q. Rieck, S. Pai and E. J. Kim. Distributed Routing Schemes for Ad Hoc Networks Using d-SPR Sets, *Microprocessors and Microsystems, Special Issue on Resource Management in Wireless and Ad Hoc Mobile Networks*, Volume 28, Issue 8, October 2004, Pages 427-437.
- [15] S. Guha, S. Khuller, Approximation algorithms for connected dominating sets, *Algorithmica*, 20(4), pp. 374-387, 1998.
- [16] C. Intanagonwivat, R. Govindan, D. Estrin, (2000), Directed diffusion: a scalable and robust communication paradigm for sensor networks, Proceedings of the ACM MobiCom'00, Boston, MA, pp. 56-67.
- [17] W.R. Heinzelman, J. Kulik, H. Balakrishnan, (1999) Adaptive protocols for information dissemination in wireless sensor networks, Proceedings of the ACM MobiCom'99, Seattle, Washington, pp. 174-185.
- [18] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, (2000) Energy-efficient communication protocol for wireless microsensor networks, IEEE Proceedings of the Hawaii International Conference on System Sciences, January, pp. 1-10.
- [19] L. Jia, R. Rajaraman, T. Suel, An efficient distributed algorithm for constructing small dominating sets, *Proc. Annual ACM Symposium on Principles of Distributed Computing*, 33-42, (2001).
- [20] D. Johnson, Approximation algorithms for combinatorial problems, *J. Computer and System Sciences*, 9:256-278 (1974).
- [21] J.M. Kahn, R.H. Katz, K.S.J. Pister, Next century challenges: mobile networking for smart dust, in: Proceedings of ACM MobiCom 99, August 1999, pp. 271-278.
- [22] S. Kutten, D. Peleg, Fast Distributed construction of small k -dominating sets and applications, *J. Algorithms*, 28:40-66 (1998).
- [23] S.-J. Lee, M. Gerla, C.-C. Chiang, On-demand multicast routing protocol, *WCNC 99*, 1298-1302 (1999)
- [24] B. Liang, Z. J. Haas, Virtual backbone generation and maintenance in ad hoc network mobility management, *Proc. 19th Ann. Joint Conf. IEEE Computer and Comm. Soc. INFOCOM*, 3:1293-1302 (2000).
- [25] H. Lim, C. Kim, Multicast tree construction and flooding in wireless ad hoc networks, *Proc. 3rd ACM Int. Wksp. Modeling, Analysis and Simulation of Wireless and Mobile Systems* (2000)
- [26] C. R. Lin, M. Gerla, Adaptive clustering for mobile wireless networks, *IEEE J. Selected Areas in Communications* 15:7:1265-1275 (1996)
- [27] S. Lindsey and C. S. Raghavendra, "PEGASIS: Power Efficient Gathering in Sensor Information Systems," in the Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, March 2002.
- [28] S. Lindsey, C. S. Raghavendra and K. Sivalingham, "Data Gathering in Sensor Networks using the Energy*Delay Metric", in the Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.
- [29] L. Lovász, On the ratio of optimal integral and fractional covers, *Discrete Mathematics*, 13:383-390 (1975).
- [30] A. Manjeshwar and D. P. Agrawal, TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks," in the Proceedings of the 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA, April 2001.
- [31] PicoRadio: http://bwrc.eecs.berkeley.edu/Research/Pico_Radio.htm.
- [32] G.J. Pottie, W.J. Kaiser, Wireless integrated network sensors, Communications of the ACM 43 (5) (2000) 5158.
- [33] D. Peleg, E. Upfal, A tradeoff between size and efficiency for routing tables, *J. of the ACM*, 36:510-530 (1989).
- [34] C. E. Perkins, E. M. Royer, Ad-hoc on-demand distance vector routing, *Proc. 2nd IEEE Wksp. Mobile Computer Systems and Appl.*, 90-100 (1999)
- [35] S. Rajagopalan, V. V. Vazirani, Primal-dual RNC approximation of covering integer programs, *SIAM J. Computing*, 28:525-540 (1998).
- [36] R. Ramanathan, M. Streenstrup, A survey of routing techniques for mobile communication networks, *Mobile Networks and Appl.* 1:2:89-104 (1996)
- [37] R. Ramanathan, M. Streenstrup, Hierarchically-organized, multi-hop mobile wireless networks for quality-of-service support, *Mobile Networks and Appl.*, 3:101-119 (1998).
- [38] T. S. Rappaport, *Wireless Communications*, Prentice Hall, 1996
- [39] M. Q. Rieck, S. Pai, S. Dhar, Distributed Routing Algorithms for Multi-hop Ad Hoc Networks Using d-hop Connected d-Dominating Sets, *Computer Networks*, Volume 47, Issue 6, April 2005, Pages 785-799.
- [40] M. Q. Rieck, S. Pai, S. Dhar, Distributed routing algorithms for wireless ad hoc networks using d-hop connected d-hop dominating sets, *Proc. 6th Int. Conf. High Performance Computing Asia Pacific*, 2:443-450 (2002).
- [41] E. M. Royer, C. Perkins, Multicast operations of ad-hoc on-demand distance vector routing protocol, *Mobicom 99* 207-218 (1999)
- [42] E. M. Royer, C. Toh, A review of current routing protocols for ad-hoc mobile wireless networks, *IEEE Personal Comm.* 46-55 (1999).
- [43] I. Stojmenovic, M. Seddigh, J. Zunic, Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks, *IEEE Trans. Parallel and Distributed Systems*, 13:1:14-25 (2002).
- [44] C.-K. Toh, *Ad Hoc Wireless Mobile Networks*, Prentice Hall Inc, 2002.
- [45] B. Warneke, M. Last, B. Liebowitz, K.S.J. Pister, Smart dust: communicating with a cubic-millimeter computer, Computer Magazine (January) (2001) 4451.

- [46] J. Wu, F. Dai, Broadcasting in ad hoc networks based on self-pruning, *Proceedings of IEEE INFOCOM*, April 2003.
- [47] WINS: http://www.janet.ucla.edu/WINS/wins_intro.htm.
- [48] J. Wu, M. Gao, I. Stojmenovic, On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks, *Proc. Int. Conf. Parallel Processing* 346-356 (2001)
- [49] J. Wu, F. Dai, M. Gao, and I. Stojmenovic, "On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks", *IEEE/KICS J. Communication Networks*, 4(1), March 2002, pp. 59-70.
- [50] J. Wu, H. Li, A dominating-set-based routing scheme in ad hoc wireless networks, *Special Issue on Wireless Networks, Telecommunication Systems J.*, 3:63-84 (2001).
- [51] J. Wu, H. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, *Proc. 3rd Int. Wksp. Discrete Algorithms and Methods for Computing and Communications* 7-14 (1999).