Topology control algorithms for wireless sensor networks: A critical survey

Alexis Papadimitriou and Dimitrios Katsaros and Yannis Manolopoulos

***Abstract*— In a densely deployed wireless sensor network, a single node has many neighboring nodes with which direct communication would be possible when using sufficiently large transmission power. This is, however, not beneficial; high transmission power requires lots of energy, many neighbors are a burden for a MAC protocol, and routing protocols suffer from volatility in the network when nodes move around. To overcome these problem topolgy control can be applied. The idea is to deliberately restrict the set of nodes that are considered neighbors of a given node. This article surveys the most popular and efficient topology control algorithms for wireless ad hoc sensor networks.**

I. Introduction

The rapid technological advances in low-power hardware design have enabled the development of tiny battery-powered sensor nodes which are able to compute, sense physical "parameters" and communicate with each other. A wireless sensor network (WSN) is a network of large numbers of sensors nodes, where each node is equipped with limited on-board processing, storage and radio capabilities [1]. Sensor nodes are quasi stationary, densely deployed and with limited capabilities. Nodes sense and send their signals towards a data center which is called "information sink". The design of protocols and applications for such networks has to be energy aware in order to prolong the lifetime of the network because it is quite difficult to recharge node batteries. Additionally, it has to take into account the multi-hop communication nature. Communication in a WSN between any two nodes that are out of one another's transmission range is achieved through intermediate nodes, which relay messages to set up a communication channel between the two nodes.

One typical characteristic of ad hoc wireless sensor networks is the possibility of deploying many nodes in a relatively small area. While a dense deployment offers advantages such as sufficient coverage control, there are also disadvantages due to the large number of nodes. Many nodes interfere with each other, there are a lot of routes, nodes might use large transmission power to send packets to relatively remote sensor nodes, and so on.

Many of these problems can be alleviated by *topology control* techniques; instead of using the possible connectivity of a network to its maximum possible extent, a deliberate choice is made to restrict the topology of the network. Topology control for ad hoc networks aims to achieve network-wide or session-specific objectives, such as reduced interference, reduced energy consumption, and increased network capacity, while maintaining network connectivity.

A. Papadimitriou and Y. Manolopoulos are with the Department of Informatics, Aristotle University, Thessaloniki, Greece. D. Katsaros is with the Department of Computer & Communication Engineering, University of Thessaly, Volos, Greece.

II. Physical topology control

DIMITRIS-THA TO PROSESO EGO SIMERA.

III. Graph-based topology control

Sparsing a topology can be efficiently done locally if information about distances between nodes and their relative positions is available. Several constructions for such proximity graphs exist with different properties. In the next subsections, we describe the Relative Neighborhood Graph (RNG), Grabriel Graph (GG), and Localized Minimum Spanning Tree (LMST).

A. Relative Neighborhood Graph

The relative neighborhood graph (RNG) [2] of a point set is a straight line graph that connects two points from the point set if and only if there is no other point in the set that is closer to both points than they are to each other. A triangu-lation of a point set is a maximal set of nonintersect- ing line segments (called edges) with vertices in the point set.

The relative neighborhood graph of a graph G = (V, E), denoted by RNG(G), is the set of all edges uv ε E such that there is no vertex or point w where $uw \ \varepsilon \ E$, $wv \ \varepsilon \ E$ and $||uw|| < ||uv||$ and $||wv|| < ||uv||$.

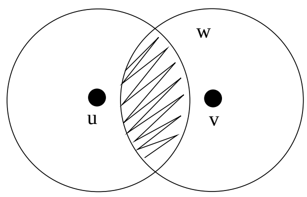

Fig. 1. Construction of RNG: Shaded region must not contain another node for two nodes u and v to be connected.

B. Gabriel Graph

Gabriel Graph has been introduced by Gabriel and Sokal in [3]. Formally, given a graph $G = \langle V, E \rangle$ and two vertices v_1 and v_2 in V, we say that v_1 and v_2 are *adjacent* if the closed disc of diameter v_1v_2 does not contain other vertices of V. In the context of sensor networks, we extend the basic adjacency concept above and we say that a sensor node s_i is *connected* with a sensor node s_j, who lies within the s_i's transmission range, if there not exist another node s_k which is contained by the closed disc of diameter $s_i s_j$. This simple-yet-effective method is used by algorithm GG to find logical neighbors of a given sensor node.

To depict GG's distributed operation, the logical neighbors of a given sensor node are found according to the following steps:

1) each sensor node broadcasts its location – at the end, every node in the sensor network knows its neighbors and their locations;
2) each sensor node s_i determines its logical neighbor set L_i by computing the closed discs of diameters equal to the distance between the location of s_i and each other physical node belonging to the s_i's physical neighborhood set P_i – for each physical neighbor s_j in P_i, if the disc of diameter $s_i s_j$ does not contain other physical neighbors of P_i then s_j becomes a logical neighbor of s_i.

C. Localized Minimum Spanning Tree

Li et al. [4] proposed the algorithm *Local Minimum Spanning Tree* (LMST), which computes a "power-reduced" network topology by constructing a minimum spanning tree over the network in a fully-distributed manner. The aim of this approach relies in the evidence that the power-reduced network is less energy-consuming than the original network.

IV. Hierarchical networks

The clustering formation procedure involves the election of a *cluster head* (CH) node in each cluster, in order to coordinate the cluster nodes. Cluster head is responsible for getting the measured values from its cluster's nodes, aggregate them and send the aggregates to the sink(s) through other cluster heads. Several studies [5], [6] indicate that clustering increases the *network lifetime*. Although the definition of the network lifetime depends on the applications' semantics, a widely accepted definition is the time until the first/last node of the network depletes its energy [7].

A. Hierarchical networks by dominating sets

The network node clustering technique has been widely investigated in the context of mobile ad hoc networks [8], [9], [10], [11], [12], [13], [14], [7]. The proposed protocols are distributed, localized and select the most significant nodes as cluster heads. In order to achieve this they compute a dominating set (DS). In [9], the author assumes quasi-stationary nodes with real-valued weights, while the Weighted Clustering Algorithm (WCA [11]) combines several properties in one parameter that is used for clustering. With Max-Min D-cluster, the authors [8] propose a new distributed cluster head election procedure, where no node is more than d (d is a value selected for the heuristic) hops away from the CH.

Wu & Li [14] proposed a distributed algorithm to find a connected dominating set (CDS) in order to design efficient routing schemes for a MANET. Every node v exchange its neighbor list with all its neighbors. A node set itself as a dominating node if it has at least two unconnected neighbors. In order to reduce the size of a CDS, some extension rules are proposed by the authors. According to first rule, a node deletes itself from the CDS when its close neighbor set (includes all its direct neighbors as well as itself) is completely included in the neighbor set of a neighboring dominating node and it has smaller ID than the neighboring dominating node. According to second rule, a node deletes itself from the CDS when its open neighbor set (includes all its direct neighbors) is completely included in the neighbor sets of a two connected neighboring dominating nodes and has the smallest ID. Stojmenovic [13] proposed an algorithm for improving the performance of the protocol that has been proposed in [14]. Nodes classified as follows. A node is called intermediate if there are two neighbors that are not directly connected. Intergateway node is called a node that is not deleted from dominating nodes after applying Rule 1 from Wu & Li protocol, while gateway is called a node that is not deleted after applying Rule 2. The author replaced node IDs with a record that includes node's degree and node's x,y coordinates. The only nodes that allowed to retransmit a message are intergateway and gateway nodes. Finally, before a node rebroadcast a message it computes the number of one-hop neighbors that have been covered from the previous rebroadcasting. In case there are uncovered neighbors, then proceed broadcasting.

A high degree of localization is presented by the protocol proposed in [12]. The authors focus on reduction of the duplicate message retransmissions while the messages are being forwarded to the destination nodes, in order to achieve efficient flooding in mobile wireless networks. The relay points of a given source or retransmitting node *u* are defined by the authors of [12] as follows. A node is assumed "covered" if it received a message originated at *u* either directly or through retransmissions by other nodes. Relay points of *u* are one-hop neighbors of *u* that cover all the two-hop neighbors of *u*. The proposed algorithm includes three phases. Initially, each node *u* starts with an empty multipoint relay set. In the second phase, node *u* selects as multipoint relays those one-hop neighbors that are unique neighbors of some nodes in *u's* two-hop neighborhood and add them in multipoint relay set. In the second phase, while there are uncovered nodes from the multipoint relay set in *u's* two-hop neighborhood, then for each one-hop neighbor not included in multipoint relay set compute the number of two-hop neighbors that it covers and are uncovered yet. Finally, add in multipoint relay set the node with the biggest number.

B. Comparison of dominating sets and LMST

In this subsection we perform a comparison of the main dominating set-based clustering topology control algorithms and of the LMST, along with a dynamic dominating set-based topology control algorithm, namely $\mathcal{NIDD}$. Firstly, we present the primitive for estimating the importance of a sensor node in participating in the CDS. The construction of the CDS is done "on-the-fly", i.e., after the broadcast of the original message by the source sensor node, and it is not calculated in advance; so we term this as the *dynamic CDS*.

1) Measuring sensor node importance: An ad hoc WSN is abstracted as a graph $G(V, E)$, where V is the set of its nodes, and E is the set of radio connections. An edge $e = (u, v),\ u, v \in E$ exists if and only if u is in the transmission range of v and vice versa. All links are bidirectional. The network is assumed to be in a connected state. The set of neighbors of a node v is represented by $N_1(v)$, i.e., $N_1(v) = \{u : (v, u) \in E\}$. The set of two-hop nodes of node v, i.e., the nodes which are the neighbors of node v's neighbors except for the nodes that are the neighbors of node v, is represented by $N_2(v)$, i.e., $N_2(v) = \{w : (u, w) \in E$, where $w \neq v$ and $w \notin N_1$ and $(v, u) \in E\}$. We define $N_{12}(v)$ as $N_{12}(v) = N_1(v) \cup N_2(v)$.

Definition 1 (Local network view w.r.t. node v): The local network view, denoted as LN_v, of a graph $G(V, E)$ w.r.t. a node $v \in V$ is the *induced subgraph* of G associated with the set of vertices in $N_{12}(v)$.

A *path* from $u \in V$ to $w \in V$ has the common meaning of an alternating sequence of vertices and edges from u to w. The *length* of a path is the number of intervening edges. We denote by $d_G(u, w)$ the *distance* between u and w, i.e., the minimum length of any path connecting u and w in G, where by definition $d_G(v, v) = 0,\ \forall v \in V$ and $d_G(u, w) = d_G(w, u),\ \forall u, w \in V$. Note that the distance is not related to network link costs (e.g., latency); it is a purely abstract metric.

Let $\sigma_{uw} = \sigma_{wu}$ denote the number of shortest paths from $u \in V$ to $w \in V$ (by definition, $\sigma_{uu} = 0$). Let $\sigma_{uw}(v)$ denote the number of shortest paths from u to w that some vertex $v \in V$ lies on. Then, we define the *node importance* index $\mathcal{NI}(v)$ of a vertex v as:

Definition 2: The $\mathcal{NI}(v)$ of a vertex v is equal to:

$$\mathcal{NI}(v) = \sum_{u \neq v \neq w \in V} \frac{\sigma_{uw}(v)}{\sigma_{uw}}. \tag{1}$$

Large values for the $\mathcal{NI}(v)$ indicate that v can reach other nodes on relatively short paths, or that v lies on considerable fractions of shortest paths connecting other nodes (Figure 2).

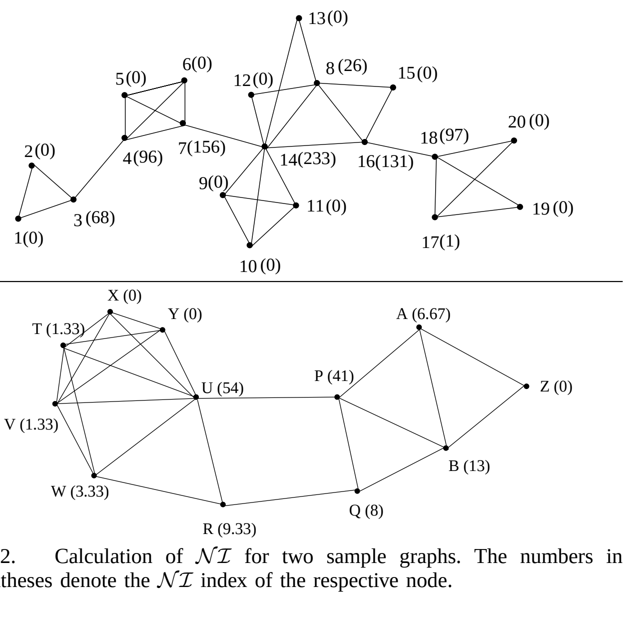

Fig. 2. Calculation of $\mathcal{NI}$ for two sample graphs. The numbers in parentheses denote the $\mathcal{NI}$ index of the respective node.

2) The distributed broadcast protocol: Exploiting the $\mathcal{NI}(\cdot)$ of each sensor, we design a broadcasting protocol to disseminate messages over the entire sensor network; we name this protocol as $\mathcal{NIDD}$, after the initials of the words Node Importance Data Dissemination protocol.

STEP 1. Assuming that node v has just gathered the collection of its neighbors and their neighbors by "Hello" messages, it calculates the $\mathcal{NI}(\cdot)$ for all sensors over its 2-hop neighborhood graph LN_v. $\hookleftarrow$

STEP 2. Then, it sorts its neighbors in descending value of their $\mathcal{NI}_v(\cdot)$ index. $\hookleftarrow$

STEP 3. Indicate which neighbors are covered by the retransmission of v itself. $\hookleftarrow$

If node v does not have links to all the other nodes of the sensor network, then there exists at least on node u, such that $u \in N_{12}(v)$, but $u \notin N_1(v)$. Therefore, broadcast by v does not cover its 2-hop neighborhood. If v is the message source, it executes STEP 4a, whereas if v was designated to broadcast, it executes STEP 4b.

STEP 4a. While its 2-hop neighborhood is not covered, examine one-by-one the members of the list obtained in STEP 2. If the currently examined 1-hop neighbor u covers at least one (not covered yet) 2-hop neighbor, then designate the 1-hop neighbor as a forwarding node. Keep examining the next 1-hop neighbor of the list, till the neighborhood is covered. $\hookleftarrow$

STEP 4b. If there are any 1-hop neighbors which have already broadcast the message, then find which part of the 2-hop neighborhood is not covered yet. While this part of the 2-hop heighborhood is not covered, examine one-by-one the members of the list obtained in STEP 2 (skipping any nodes that have already broadcast). If the currently examined 1-hop neighbor u covers at least one (not covered yet) 2-hop neighbor, then designate the 1-hop neighbor as a forwarding node. Keep examining the next 1-hop neighbor of the list, till the neighborhood is covered. $\hookleftarrow$

STEP 5. Retransmit the message, augmented by the list of neighbors designated as forwarding nodes. $\hookleftarrow$

It is relatively easy to prove that:

Proposition 1: The broadcasting nodes form a CDS.

3) Simulation results: We performed simulation experiments for the protocols using the size of the generated CDS as a measure of the communication complexity. We believe this metric is representative of the latency metric, since a smaller dominating set implies less broadcasting nodes, thus smaller probability of collisions, shorter message routes and smaller processing and communication times. Besides this metric has been used in earlier studies as well, e.g., [14], [15], [16]. We assume that we are able to determine an assignment of time slots to the sensor nodes such that no interference occurs, i.e., no two nodes transmit in the same time slot. Such a scheme can be found using the D2-coloring algorithm from [17].

We created network topologies, modelling features such as the existence and "strength" of clusters, density of nodes etc. We observed that the topologies generated with procedures like that in [13], or with procedures that distribute nodes randomly in the plane with random velocities and speeds, are alike the Random Graph Model of Erdös-Rényi. Although this model is quite useful, we argue that it is not suitable for ad hoc network graphs, because these graphs are not formed uniformly at random, but present a *group/cluster-based behaviour*. Thus, we had to resort to richer graph models that model the existence of clusters, like that of Pennock [18].

The parameters of the network topology generator are:

- gn: the number of network nodes (default value: 100).
- gc: the number of network clusters (default value: 7).
- gd (*density*): a float depicting the fraction of edges relative to the edges of a complete graph with gn nodes; small values of gd simulate a small transmission radius. gd controls the average node degree (default value: 10).
- $ga \in [0.5 \ldots 0.99]$ (*assortativity*): a float depicting the fraction of edges which exist inside the clusters, relative to the total number of edges present in the graph (default value: 70%). Large values (> 85%) simulate clusters with very dense linkage inside them and only a few links toward other clusters, whereas values around 0.50 completely "blur" the existence of clusters.

As competing methods, we implemented two baseline schemes [14], i.e., the basic scheme without the two rules (Rule 1 and Rule 2) indicated as *WL* and a scheme incorporating these rules (*WL_1+2*). We also implemented the MultiPoint Relaying method [15] (*MPR*), and its superior extension [16] (*AHBP*). Finally, we implemented a high performance broadcasting algorithm [13] (*SSZ*), and the localized minimum spanning tree broadcasting algorithm [4] (*LMST*). All protocols use 2-hop information, except from *SSZ* and *LMST*.

Impact of the number of nodes. We can easily figure out (Figure 3) the linear dependence of the CDS size on the network size and the efficiency of the $\mathcal{NIDD}$ protocol, which always performs from 4% to 10% better than the second best performing algorithm no matter what the scale of the network is (in terms of number of nodes). Moreover, the performance gap between $\mathcal{NIDD}$ and its competitors widens as the network size grows.

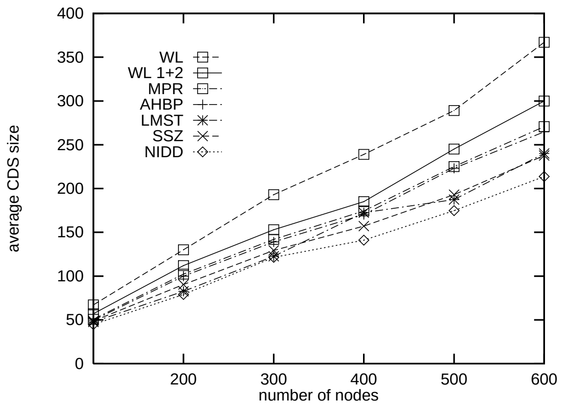

Fig. 3. Impact of the nodes' number on CDS size.

Impact of the number of clusters. The general trend (see Figure 4) is that the larger the number of clusters is the smaller the generated CDS is. This trend is followed by all methods and it is explained by the fact that a large number of cluster implies smaller clusters with more dense linkage between the nodes of the clusters (since the density and the assortativity is the same). $\mathcal{NIDD}$ is the most efficient algorithm and it is not affected significantly by the number of clusters.

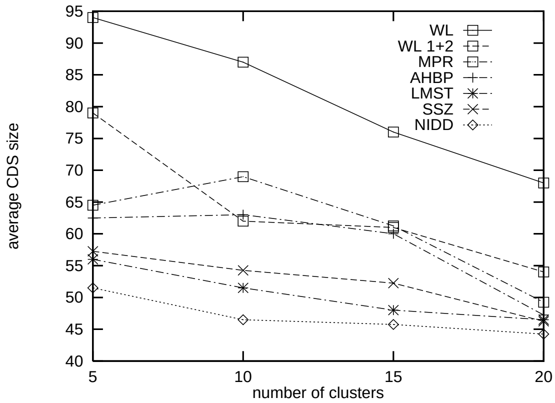

Fig. 4. Impact of the clusters' number on CDS size.

Impact of the strength of clusters. We evaluated the impact of the clusters' "strength" (assortativity) on the size of the CDS (Figure 5). $\mathcal{NIDD}$ exhibits an immunity on this parameter, which is a desirable feature for a broadcasting algorithm, since (ideally) we are interested in making locally optimal decisions, irrespectively of the existence or not of clusters. For the degenerate case $ga = 0.90$, $\mathcal{NIDD}$ as well as the rest of the protocols take advantage of the well-clustered network in creating a very small forward-node set.

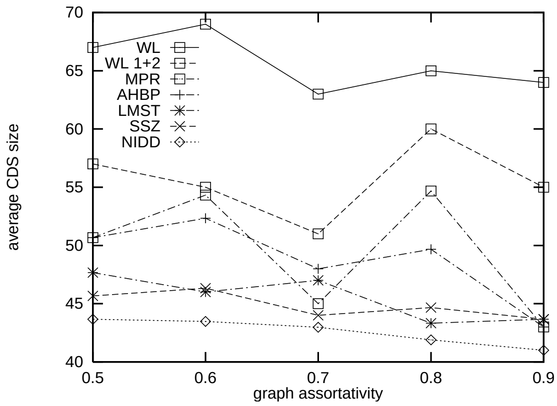

Fig. 5. Impact of the "strength" of clusters on CDS size.

C. Hierarchical networks by clustering

Apart from this family of algorithms, a second family provided mechanisms to address the energy consumption problem due to the repetitive communication by the same nodes, i.e., the cluster heads. This family of protocols essentially proposed ways to "rotate" the role of cluster head among nodes of clusters, e.g., the SPAN [19], the LEACH [5], and the HEED [6]. The proposed methods use the residual energy of each node in order to direct its decision about whether it will elect itself as a cluster head node or not. However, this family's methods ignore topological features of the nodes.

LEACH [5] is an energy efficient protocol designed for sensor networks with continuous data delivery mechanism and no mobility. Sensor nodes elect themselves as cluster heads with some probability and broadcast their decisions. The remaining nodes join a cluster, of which the cluster head is closest in terms of the communication energy cost. Then the role of cluster head is periodically rotated among the nodes to balance energy consumption, since cluster heads have the extra burden of performing a long-range transmission to a distant sink node. Thus, LEACH counteracts the problem of non-uniform energy drainage by role rotation. HEED [6] introduces a variable known as cluster radius which defines the transmission power to be used for intracluster broadcast. The initial probability for each node to become a tentative cluster head depends on its residual energy, and final cluster heads are selected according to the intracluster communication cost. HEED relies on the assumption that cluster heads can communicate with each other and form a connected graph; realizing this assumption in practical deployments could be tricky. In [20], the authors use LEACH-like clustering and multi-hop forwarding for both intracluster and intercluster communication. They provide also methods in order to compute the optimal values of the algorithm parameters a priori. Chang and Tassiulas [21] proposed methods in order to maximize overall network lifetime by distributing energy consumption fairly. In this protocol, nodes adjust their transmission power levels and select routes to optimize performance. In [22], a multilevel hierarchical structure is proposed where cluster heads are selected according to their residual energy. Buttyan et al. [23] propose a Position-based Aggregator Node Election (PANEL) in Wireless Sensor Networks. PANEL is an energy-efficient protocol that ensures load balancing in the sense that each node is elected aggregator (CH) nearly equally frequently. However, PANEL uses the geographical position information of the nodes to determine which of them should be the aggregators, which is a restriction in WSNs, since the geographical position is difficult to obtained without the use of GPS-like hardware or central coordination.

In [24], the authors propose a new energy efficient clustering approach (EECS) for single-hop wireless sensor networks, which is more suitable for the periodical data gathering applications. EECS extends LEACH algorithm by dynamic sizing of clusters based on cluster distance from the base station. In the cluster head election phase, unlike LEACH, the cluster head is elected by localized competition and its no iteration property makes it differ from HEED. This competition involves candidates broadcasting their residual energy to neighboring candidates. If a given node does not find a node with more residual energy, it becomes a cluster head. However, EECS protocol does not consider the structural characteristics of network topology and thus cluster heads elected based on residual energy. The strategy proposed in [25] is not scalable as it requires all nodes in the WSN to be in direct transmission range of the base station. The authors proposed a strategy to save energy in continuous data collection applications in WSN by exploiting the spatiotemporal correlation. Thus, the sink node partitions the sensor nodes with similar measured values into clusters and the sensor nodes within a cluster are scheduled to work alternatively in order to reduce energy dissipation. Youssef et al. [26] proposed MOCA, a randomized, distributed Multi-hop Overlapping Clustering Algorithm for organizing the sensors into overlapping clusters. However, the major goal of the clustering process is to ensure that each node is either a cluster head or within k hops from at least one cluster head, where k is a preset cluster radius.

V. SNA-BASED APPROACHES TO TOPOLOGY CONTROL

The area of Social Network Analysis is a broad, diverse and theoretically varied field, with a long and rich history. Informally, a *social network* is a collection of ‘actors’ (i.e., network nodes), a set of relational information on pairs of actors (i.e., wireless links), and possible attributes of the actors and/or of the links. The notion of a social network and the methods of social network analysis (SNA) is a quite old discipline and they have attracted significant interest initially from the social and behavioral communities, later from the data mining, and only recently from the networking community. This interest stems from the focus of SNA to relationships among entities and on the patterns and implications of these relationships. SNA could be viewed as another network measurement task, while the traditional tasks of network measurement deal with issues such as traffic monitoring, latency, bandwidth, congestion. The analysis of the ‘social’ aspects of a network is the study and exploitation of the structural information present in the network, such as existence and strength of communities, node centralities, network robustness to node removal, topology evolution over time, and so on.

A. Topology control with Edge Betweenness Centrality

During past years, *vertex betweenness* has been studied in the vest of a measure of the centrality and influence of nodes in networks [27], [28]. Given a node v_i, vertex betweenness is defined as the number of shortest paths between pairs of nodes that run through v_i. Vertex betweenness is a measure of the influence of a node over the information flow among nodes of the network, especially in scenarios such that information flowing over the target network primarily follows shortest available paths.

In order to compute betweenness centrality, Brandes [29] proposes an efficient *backwards algorithm* which starts from leaf nodes of a tree of shortest paths and progressively accumulates the leaf-nodes’ betweenness values moving bask towards the root node of the tree.

Girvan-Newman algorithm [30] extends the definition of betweenness centrality from network vertices to network edges, via introducing the concept of *Edge Betweenness* (EB). Let $G = \langle V, E \rangle$ be a connected undirected graph, and v_i and v_j two nodes in G, respectively. Let $\sigma_{v_i v_j}$ denote the number of shortest paths between nodes v_i and v_j. Let $\sigma_{v_i v_j}(e)$ denote the number of shortest paths between v_i and v_j which go through $e \in E$. Betweenness centrality of an edge $e \in V$, denoted by $EB(e)$, is defined as follows:

$$EB(e) = \sum_{v_i \epsilon V} \sum_{v_j \epsilon V} \frac{\sigma_{v_i v_j}(e)}{\sigma_{v_i v_j}} \quad (2)$$

In its original implementation [31], which focuses on unweighted, undirected networks, EB analysis makes use of algorithm *Breadth-First Search* (BFS). Girvan-Newman algorithm [30] works in the opposite way. Instead of trying to construct a measure that determines edges that are the "most central" for network communities, it focuses on edges that are the "least central" for network communities, i.e. edges that are "most between" for network communities. Communities are detected by progressively removing edges from the original graph, rather than by adding the strongest edges to an initially empty network. In our research, we do not use the centrality measure to find communities but instead to select the most important edges, energy-wise, to propagate messages.

Specifically, steps that are used to compute the edge betweenness centrality index are the following:

1) compute shortest paths through the network by means of Dijkstra's algorithm [32];
2) for each edge, compute the edge betweenness centrality index like in [31].

VI. Discussion

DIMITRIS-THA TO PROSTHEESO EGO SIMERA.

References

[1] I. F. Akyldiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor netowrks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] G. Toussaint, "The Relative Neighborhood Graph of a finite planar set," *Pattern Recognition*, vol. 12, pp. 261–268, 1980.

[3] R. K. Gabriel and R. R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Zoology*, vol. 18, no. 3, pp. 259–278, September 1969.

[4] N. Li, J. C. Hou, and L. Sha, "Design and analysis of an MST-based topology control algorithm," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 1195–1206, 2005.

[5] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[6] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[7] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: Recent developments and deployment challenges," *IEEE Network*, vol. 20, no. 3, pp. 20–25, May/Jun. 2006.

[8] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh, "Max-min d-cluster formation in wireless ad hoc networks," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 1, 2000, pp. 32–41.

[9] S. Basagni, "Distributed clustering for ad hoc networks," in *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN)*, 1999, pp. 310–315.

[10] S. Basagni, M. Mastrogiovanni, A. Panconesi, and C. Petrioli, "Localized protocols for ad hoc clustering and backbone formation: A performance comparison," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 4, pp. 292–306, 2006.

[11] M. Chatterjee, S. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks," *ACM/Kluwer Cluster Computing*, vol. 5, no. 2, pp. 193–204, 2002.

[12] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," INRIA, Tech. Rep. 3898, March 2000.

[13] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 1, pp. 14–25, 2002.

[14] J. Wu and H. Li, "A dominating-set-based routing scheme in ad hoc wireless networks," *Telecommunication Systems*, vol. 18, no. 1–3, pp. 13–36, 2001.

[15] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," in *Proceedings of the IEEE Hawaii Conference on System Sciences (HICSS)*, 2002.

[16] B. Williams and T. Camp, "Comparison of broadcasting techniques for mobile ad hoc networks," in *Proceedings of MobiHoc*, 2002, pp. 194–205.

[17] R. Gandhi and S. Parthasarathy, "Fast distributed well connected dominating sets for ad hoc networks," University of Maryland at College Park, Tech. Rep. CS-TR-4559, Tech. Rep., 2004.

[18] D. Pennock, G. Flake, S. Lawrence, E. Glover, and C. Giles, "Winners don't take all: Characterizing the competition for links on the Web," *Proceedings of the National Academy of Sciences*, vol. 99, no. 8, pp. 5207–5211, 2002.

[19] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "SPAN: An energy efficient coordination algorithm for topology maintenance in ad hoc networks," *ACM/Kluwer Wireless Nerworks*, vol. 8, no. 5, pp. 481–494, 2002.

[20] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, 2003, pp. 1713–1723.

[21] J. Chang and L. Tassiulas, "Energy conserving routing in wireless ad hoc networks," in *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 1, 2000, pp. 22–31.

[22] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, 1999.

[23] L. Buttyan and P. Schaffer, "Panel: Position-based aggregator node election in wireless snsor networks," in *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, 2007, pp. 1–9.

[24] M. Ye, C. Li, G. Chen, and J. Wu, "An energy efficient clustering scheme in wireless sensor networks," *Ad Hoc and Sensor Wireless Networks*, vol. 3, pp. 99–119, 2006.

[25] C. Liu, K. Wu, and J. Pei, "An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 7, pp. 1010–1023, 2007.

[26] A. Youssef, M. Younis, M. Youssef, and A. Agrawala, "Distributed formation of overlapping multi-hop clusters in wireless sensor networks," in *Proceedings of the 49th Annual IEEE Global Communication Conference (Globecom)*, 2006, pp. 1–6.

[27] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, March 1977.

[28] ——, "Centrality in social networks: Conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1979.

[29] U. Brandes, "A faster algorithm for betweenness centrality," *Journal of Mathematical Sociology*, vol. 25, pp. 163–177, 2001.

[30] M. Girvan and M. E. Newman, "Community structure in social and biological networks." *Proc Natl Acad Sci U S A*, vol. 99, no. 12, pp. 7821–7826, June 2002.

[31] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," August 2003.

[32] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, December 1959. [Online]. Available: http://dx.doi.org/10.1007/BF01386390