Topology-based Clusterhead Candidate Selection in Wireless Ad-hoc and Sensor Networks

Matthias R. Brust, Adrian Andronache, Steffen Rothkugel
Faculty of Science, Technology and Communication (FSTC)
University of Luxembourg
Luxembourg
{matthias.brust, adrian.andronache, steffen.rothkugel}@uni.lu

Zinaida Benenson
Department of Information Technology
Uppsala University
Uppsala, Sweden
zina@it.uu.se

***Abstract*—Clustering techniques create hierarchal network structures, called clusters, on an otherwise flat network. Neighboring devices elect one appropriate device as clusterhead. Due to the dynamic environment, clusterhead selection becomes an important issue. We consider the problem of appropriate clusterhead selection in wireless ad-hoc networks and sensor networks. This work presents topological criteria for robust clusterhead candidate selection, resilient to sporadic node mobility and failure as well as for efficient information dissemination. One of the main ideas of our approach is to avoid selecting nodes located close to the network partition border as such nodes are more likely to move out of the partition, thus causing a clusterhead re-election. We conducted experiments both for static topologies as well as for cases in the presence of node mobility. Our results showed that the frequency of clusterhead re-election and average shortest path length from the clusterhead decrease when considering topological criteria. Additionally, the clusters tend to be robust to clusterhead failure. The presented mechanisms rely on local topological information only and do not require geographical data.**

Keywords-clustering, clusterhead selection; ad-hoc networks; sensor networks; topology control

I. INTRODUCTION

Multi-hop ad-hoc networks as well as wireless sensor networks are composed of a collection of devices that communicate with each other over a wireless medium [1]. Such a network forms spontaneously whenever devices are in transmission range. Joining and leaving of nodes occurs dynamically, particularly when dealing with mobility. Potential applications of ad-hoc networks can be found in traffic scenarios, ubiquitous Internet access, collaborative work, and many more. Wireless sensor networks gather and process environmental data. They consist of small devices communicating via radio. Normally, data processing occurs local and decentralized. Application areas are environmental observations as well as search and rescue scenarios as described in detail in [2].

In both ad-hoc networks and sensor networks, clustering is one of the most popular techniques for locality-preserving network organization [3]. Cluster-based architectures effectively reduce energy consumption, and enable efficient realization of MAC and routing protocols [4], data aggregation [5, 6], and security mechanisms [7].

A cluster is a group of interconnected nodes with a dedicated node called clusterhead. Clusterheads are responsible for cluster management, such as scheduling of the medium access, dissemination of control messages, or data aggregation. Therefore, the role of the clusterhead is critical for the proper network operation. Failure of a clusterhead results in expensive clusterhead re-election and re-clustering operations.

In static networks, the role of the clusterhead may be assigned to any node in the cluster in a self-organized way. Often, this role is assigned in turn to the nodes in order to ensure fairness, as a clusterhead consumes more energy than a regular sensor or ad-hoc node. An important criterion for the clusterhead selection is the remaining energy level of the node. However, for fault-tolerant clusterhead selection in dynamic networks, some additional criteria for choosing a clusterhead apply. For example, considering node mobility, if a clusterhead is close to the network partition border, it may vanish from the cluster earlier than a more centrally located node. On the other hand, a centrally located node should not be selected as a clusterhead if its failure leads to cluster partitioning.

We overcome these limitations by classifying nodes into different types with certain attributes that serve as indicators for proper clusterhead selection. The algorithms introduced are using two-hop topological information only, rather than relying on geographical positions.

In this context, we make three contributions in this paper:

(1) Heuristic algorithms are introduced with the objective to find clusterhead candidates on local topological information that are efficient in terms of information dissemination, more stable to mobility, and failures shall not cause partitioning.

(2) Clusterhead candidates fulfill these three criteria when using certain environmental settings and that there is a potential to further improvements.

(3) A topology owns of nodes that are more stable topologically in a dynamic environment than other nodes. The characteristics of these nodes are not inherent to that specific topology, but of the whole topology family, i.e. topologies generated with same settings.

The remainder of this paper is organized as follows. In the following section, related work is discussed. Section 3 contains criteria for choosing clusterheads. Section 4 introduces heuristic algorithms for classifying nodes according to these criteria. We present a simulation study of our approach in Section 5, and conclude with Section 6.

II. Related Work

Algorithms for the clusterhead selection can be based on such criteria as energy level of the node, node position, degree, speed and direction. Centralized and distributed approaches can be distinguished, as well as probabilistic and deterministic ones.

One of the first and most influential cluster-based protocols for sensor networks is LEACH [8]. It uses a distributed probabilistic approach. Each node elects itself as a clusterhead with a certain probability based on the desired percentage of clusterheads in the network, and the last round where it served as a clusterhead. Thus, the role of the clusterhead is determined probabilistically, which allows to conserve a large amount of energy. In [4], a centralized clusterhead election algorithm is presented, where the base stations assign the clusterhead roles based on the energy level and the geographical position of the nodes.

Chatterjee et al [9] propose a distributed deterministic clusterhead selection algorithm, namely WCA, based on transmission power, degree, mobility, and battery power of the node. They also give an overview of earlier work in this area. In [10], a centralized algorithm based on fuzzy logic is proposed. The nodes are selected as clusterheads by the base station based on their distances to each other, energy level, and the concentration of the nodes in the region. Tan et al. [11] present a distributed clusterhead selection algorithm where each node computes its priority based on its ID, current communication round, energy level and speed, and exchanges this information with its two-hop neighbors. The nodes with highest priority become clusterheads.

Andronache et al [12] follow a similar approach with the proposed WACA algorithm that is optimized for hybrid wireless networks. A heuristic weighting function is introduced that uses battery power, signal strength, stability to neighbor, degree, local clustering as well as appropriate weighting factors as parameters to calculate the weight of each device. Following, multi-hop clusters are created in accordance of the weight of the nodes unlike Chatterjee et al [9] which uses single-hop clusters. It is shown that the WACA algorithm produces less clusterheads than WCA.

All aforementioned algorithms do not give guarantees on the resulting network structure, e.g., on the size of the resulting set of clusterheads. Their effectiveness is evaluated by simulation. In this sense, these algorithms represent heuristics. Our approach also falls into this category. Theoretical approaches to choosing clusterheads in turn concentrate on computing minimum dominating sets (MDS) and maximal independent sets (MIS) for unit disc graphs [3], and their approximations. These algorithms typically provide bounds on running time and approximation ratio. For an overview and recent results, see e.g. [13, 14].

III. System Model

We define a wireless network to consist of a set of computers or sensors connected by wireless network links. We consider a wireless network with homogeneous nodes placed according to a network model, such as random geometric graph [15] and a network growth model [16].

All nodes have the same communication range with bidirectional communication links. This appears as a valid consideration, because wireless protocols such as 802.11 MAC layer require links to be bidirectional for unicasts, too. We furthermore abstract away the details of the MAC and network layer.

Nodes might be static or mobile, but sometimes they may fail, disappear from the network (e.g., due to some obstacles), or move arbitrarily to a place relatively close to their previous position. Thus, the neighborhood of a node changes over time. For evaluation reasons, nodes are moving according to the random waypoint mobility model.

We assume that every node knows its current one-hop neighbors as well as the list of two-hop neighbors. Geographical positions of the nodes are not used.

IV. Criteria for Clusterhead Selection

We present criteria for choosing potential appropriate clusterhead candidates. The algorithms presented can be used in combination with any other clusterhead selection algorithm, e.g. [17] or [12].

Our approach supports clustering in unstable networks, i.e. ones obeying a certain mobility rate. The goals are to facilitate more efficient information dissemination in clusters, to reduce the frequency of clusterhead re-elections due to mobility, and to ensure connectivity between cluster members in case of clusterhead failure. In order to achieve these goals, we identified the following criteria for an appropriate clusterhead:

[C1] *Efficient information flow*: choosing nodes which can disseminate information in a particular region more efficiently, e.g., with a minimal number of hops or messages. One such criterion usually considered in the literature is the number of neighbors (degree). However, in Section 5.1, we argue that a better criterion possibly is the clustering coefficient.

[C2] *Insensitivity to mobility*: the clusterhead should be placed "centrally" in the region, such that its movement would not cause immediate loss of connections with the nodes in the region. In other words, placing clusterheads on the border of a densely populated network region should be avoided.

[C3] *Robustness to clusterhead failures*: it cannot be prevented that a network node fails occasionally. The failure of a clusterhead, however, should at least not cause the partitioning of the cluster.

In the following, we present heuristics algorithms for identifying nodes that satisfy criteria C1-C3.

V. CLUSTERHEAD CANDIDATE SELECTION

As we strive for not relying on geographical positions, we use the connectivity information provided by the two-hop neighborhood. At first, we classify the nodes into *weak* and *strong nodes*. Based on this classification, we further identify *bridge nodes* and *border nodes* among the strong nodes. We argue that weak, bridge, and border nodes do not fulfill criteria C1-C3. For each class of nodes, we present a heuristic algorithm that work properly in most cases. The overall objective is not to look at each criterion individually, but to focus on a balanced combination of all three instead.

A. Strong and Weak Nodes

To accomplish criterion C1, a node should have a high degree of connectivity. This way, more information can be disseminated or gathered using one-hop communication, which avoids multi-hop communication, thus unloading the network. However, to fulfill the requirement C3, the node should not represent a potential single point of failure in the information flow. For example, in Figure 1a, node *v* does not have a high degree of connectivity, thus not fulfilling C1. On the other hand, the node is located centrally in the region, which makes its position less sensitive to movements. Thus, if this node fails, its neighbors remain connected. Therefore, the node fulfills criteria C2 and C3.

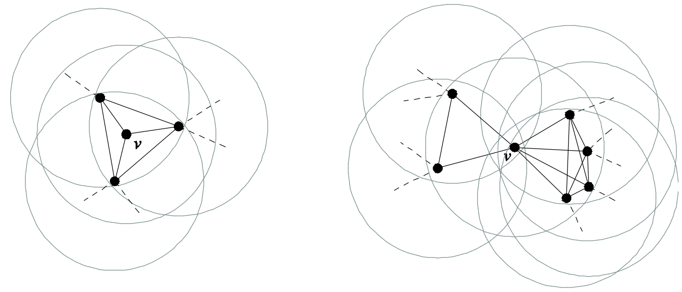
To facilitate the selection of nodes that fulfill all three criteria, we propose considering not only node degree, but also the connectivity between the neighbors of the node. For this purpose, we apply the local clustering coefficient as introduced in [18]. The local clustering coefficient CC of a node *v* with k_v neighbors is the number of links between the neighbors of *v* divided by the number of all possible links which is!" (!" −1)/2 .

We classify all nodes that have less than three neighbors and also all nodes with a clustering coefficient less than a certain threshold T_C, as *weak nodes* otherwise as *strong nodes*. In Figure 2, the pseudo code for detection of weak and strong nodes is given. Informally speaking, weak and strong nodes also differentiate in some way between sparse and dense regions of a network partition. Our experiments show that $T_C = 0.4$ represents a reasonable threshold value. Depending on the type of applied topology, a threshold higher than 0.4 results in classifying many beneficial nodes as weak nodes.

B. Bridge Nodes

In the previous section, we pointed out that weak nodes mark sparse regions. Strong nodes obey a more complex nature, because strong nodes can have a high degree of connectivity as well as higher CC-values. Nevertheless, they may represent a potential single-point-of-failure at the same time. Figure 1b shows a setting where CC for node *v* is above the chosen threshold of 0.4 and does not fulfill to C2-C3. Experiments showed that increasing the threshold results in classifying many appropriate nodes along the lines of C1-C3 as weak nodes.

These problematic situations appear when a node *v* has a high number of neighbors and the neighbors are *grouped*. To



(a) (b)

Figure 1. Using the node degree as an indicator of a good clusterhead is not appropriate: (a) node *v* with the low degree can nevertheless be a redundant node, thus fulfilling C2 and C3; (b) a high node degree does not necessarily mean that a node is not a single-point-of-failure (for node v, C1 is fulfilled while C2 and C3 are not).

```
Algorithm Weak Node Detection (for node " )
Input :    Neighborhood # (" ) of "
           Neighborhoods # ($) of $ ∈ # (" )
Output :   true if weak node otherwise false
1 :   % ← ∅
2:    if |# (")| < 3 then
3 :       return &$(
4:    for each $ ∈ # (" ) do
5 :       for each *  ∈ # ($) do
6 :           % ← %∪{($,* ) | *  ∈ # (")}
7 :   ) ) ← 2·|%| / ((|# (")|−1)·|# (")|)
8 :   if ) ) ≤ +) then
9 :       return &$(
10 :  return ,- ./(
```

Figure 2. Pseudo code for detection of weak and strong nodes. Threshold T_C is set at line 8. It was shown that $T_C = 0.4$ represents a good threshold value. A higher threshold classifies too many beneficial nodes as weak nodes.

```
Algorithm Bridge Node Detection (for node " )
Input :    Neighborhood # (" ) of "
           Neighborhoods # ($) of $ ∈ # (" )
Output :   true if bridge node otherwise false
1 :   % ← ∅
2:    for each $ ∈ # (" ) do
3 :       for each *  ∈ # ($) do
4 :           % ← %∪{($,* )|*  ∈ # (")}
5 :   0 ' 1$2($) ← /&' &3456 1 int $
6 :   for |# (" ) | −1 do
7 :       if ∃($,* ) ∈ % where $ ∈ 0 ' 1$2($) then
8 :           0 ' 1$2($) ← {* |*  ∈ # (")}
9 :   if |# (") | ≠ |0 ' 1$2($) | then
10 :      return &$(
11:   else
12 :      return ,- ./(
```

Figure 3. Pseudo code for detecting bridge nodes in bidirectional graphs nodes as weak nodes.

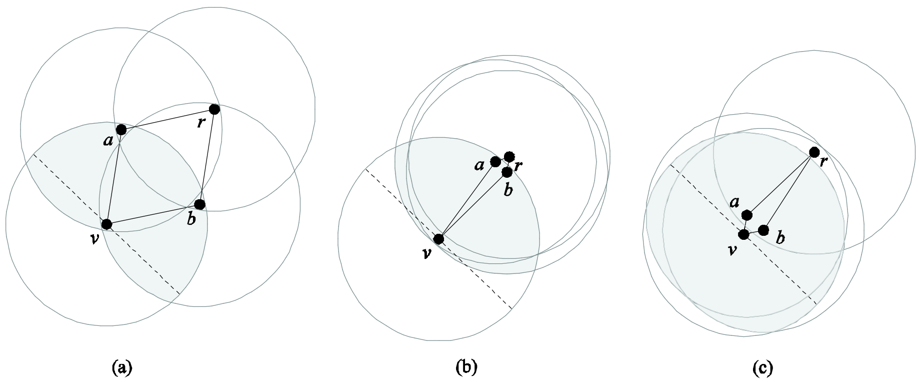


Figure 4. The basic algorithm for detecting border nodes utilizes a reference node *r* and two spanning nodes *a* and *b*.

illustrate this term, suppose0 = (# ("),%#) is given where %# the edges between the neighbors of v are. If it is not possible to reach each node from any neighbor node as starting point ignoring node *v* then we call the neighbors of *v* grouped. Consequently, *v* represents a potential single-point-of-failure (C3 is violated, cf. Figure 3). In that case, the node *v* is called a *bridge node*, because it provides the only possible local path between these groups.

C. Border Nodes

A more complex type, we identified as *border nodes*. These nodes are located geographically at a border of a partition or cluster. The challenge in detecting border nodes lies in trying to estimate geographical relations while relying on local topological information only. Our approach uses a reference node *r* for tackling this problem (cf. Figure 4).

Besides the reference node *r*, the algorithm also utilizes two spanning nodes *a* and *b*. Assume a pair *a* and *b* exists, both neighbors of *v*, and both having a common neighbor that is not neighbor of *v*. In this case, it will be checked if the neighborhood of *v* is completely covered by the neighborhood # (- ⋃7) of *a* and *b*. If ∀$ ∈ # (") with $ ∈ # (- ⋃7) then *v* is marked as border node (cf. Figure 4a). Ideally, the relationship between *r*, *a* and *b* is such that the resulting coverage (grey shaded area) covers around half of the coverage of *v* (cf. Figure 4a and Figure 4b). Since no geographical positions are available and nodes can be located anywhere, settings might exist that almost approximate the coverage of *v*. Figure 4c illustrates this worst-case scenario.

Figure 5 illustrates the pseudo code for detecting border nodes. Observe that the criterion for a border node is fulfilled if one set of nodes *a*, *b* and *r* exists that covers the neighbors of *v*. A more restrictive condition is that all possible sets of nodes *a*, *b* and *r* covers the neighbors of *v*. First experiments showed interesting results, but they are not pointed out in this paper.

VI. Simulation Study and Evaluation

The different node classification algorithms have been implemented and tested in *ConnyLab* [19]. We used following parameters for the experiments if not described in an opposite manner.

Topology. Topologies are generated using the geometric random graph (GRG) approach as well as using a network growth model (NGM) as it happens on shows or market places.

```
Algorithm Border Node Detection (for node " )
Input :   Neighborhood # (" ) of "
          Neighborhoods # ($) of $ ∈ # (" )
Output :  true if border node otherwise false
   for each ' ∈ # (- ⋃7) do
      if ' ∉ # (" ) then
         if ∀$ ∈ # (" )|$ ∈ # (- ⋃7) then
            return &$(
   return ,- ./(
```

Figure 5. Pseudo code for border detection.

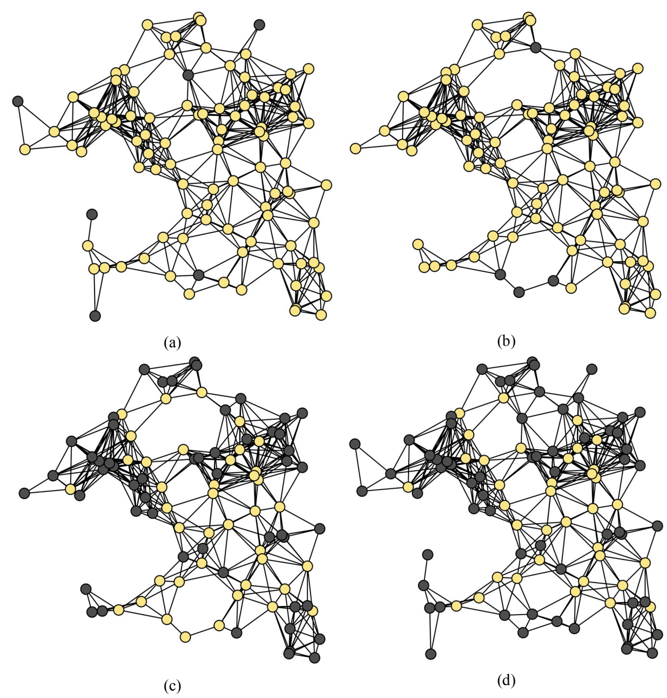


Figure 6. Weak nodes (a), bridge nodes (b) and border nodes (c) are marked in dark. In (d) all node types are included, indicated clusterhead appear clear. A NGM topology, a 300 m×300 m square area with 100 nodes is used, the transmission is set to 50 m.

Mobility Model. Random waypoint model is used with device speed between 0 m/s and 2 m/s. These velocities correspond to those of pedestrians or the velocity of sensors moving in some rivers. The pause time is set to 0, because we are interested in the startup sequence.

Simulation Area. The simulation area is a 300×300-square in meters.

Transmission Range. As described in the system model, all devices are supposed to have the same transmission range of 50 m establishing only bidirectional communication links.

The output topology of the weak node detection algorithm is used as input topology for the bridge and border detection algorithms (Figure 6).

A. Metrics and Performance Evaluation

[C1] *Efficient information flow*. In order to find out if clusterhead candidates tend to spread information more efficiently than any arbitrarily chosen node, we use a hop-count measure. We understand a shorter path from one node to another as a performance improvement of the information flow efficiency. The question of how to find this path is task of the routing layer that is not issue of this paper.

The all-pair shortest path of a network with the set of nodes *N* is the average shortest path from any node to another.

$$6 = \frac{1}{|\#|(|\#|-1)}\sum_{3\neq 8} 9(3,8) \quad \text{with } 3,8 \in \#\ ,$$

where $9!3'8\#$ is the length of the shortest path between nodes *i* and *j*. Assume the set of clusterhead candidates *C*, the clusterhead candidates-to-all shortest path is

$$6 = \frac{1}{)|()|-1)}\sum_{3\neq 8} 9(3,8) \quad \text{with } 3 \in)\ ,\ 8 \in \#\ .$$

Non-clusterhead candidates-to-all shortest path is calculated with the set *NC* of non-clusterhead candidates respectively.

For this, the all-pair shortest path value is calculated and compared with the average one-to-all shortest path initiated by clusterhead candidates as well as with the average one-to-all shortest path initiated by non-clusterhead candidates, i.e. weak nodes, bridge nodes, and border nodes.

Observe that during this measurement we assume connected topologies, i.e. one partition, because the shortest path between two nodes is defined for connected nodes only. Therefore, we modified the topology generation slightly for this purpose. Note also that there are different reasons why the network diameter represents an inappropriate metric here and the shortest path is a preferable metric. One reason is the average shortest path represents a characteristic for all nodes. Furthermore, we assume for theoretical aspects that the routing algorithm running on the topology is optimal, i.e. using the shortest path.

Figure 7 shows the hop counts for comparing clusterhead candidates (CH) node-to-all shortest path length with the nodes-to-all shortest path length of all non-clusterhead candidates. We measured also the average number of clusterhead candidates and average node degree. The algorithms classified steadily between 27.5% and 38.8% of the nodes as potential clusterhead candidates.

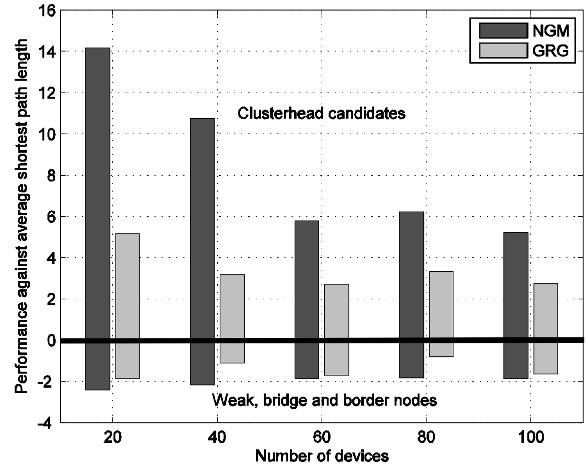


Figure 7. Hop performance, comparing clusterhead candidates node-to-all shortest path length with the nodes-to-all shortest path length of all non-clusterhead candidates (i.e. the set of weak, bridge, and border nodes). The baseline is the average shortest path for each network topology. The data shown are averages over 20 random realizations of each topology model.

The election of one clusterhead from the clusterhead candidates results in a hop improvement depending on the number of devices and the topology between 2.70% and 14.17%. Furthermore, non-clusterhead candidates tend to have a worse average shortest path in the network compared to the all-pair-shortest path that is between -2.43% and -0.82%.

[C2] *Insensitive to mobility*. Prolonged topological stability of clusterheads in mobile environments is an important issue [12, 17]. An important issue for most clustering algorithm is the update policy. Depending on the changes of the environment, e.g. data flow or topology, the clustering mechanism has to be re-initiated and new clusters and clusterheads must be elected. Due to the overall message complexity in establishing clusters, it is a crucial point how to design the update policy, i.e. the frequency of re-elections. Decreasing the frequency of re-elections by choosing preventively appropriate clusterheads candidates has an arbitrative impact on the performance of the whole network.

Our metric for testing if the introduced heuristic has a positive effect in terms of prolonged topological stability is described following. The initial neighborhood N^0 of one node *v* is compared to its changes in time *t* whereby new neighbor nodes are ignored. The results are added and averaged over the total number of nodes *n*.

$$\&_4(\&) = \left(\sum_{3=1}^{4}\frac{\left|\#_3^0(") \setminus \#_3^{\&}(")\right|}{\left|\#_3^0(")\right|}\right)\cdot\frac{1}{4}$$

First, we assumed that each topology might have a different characteristic for an appropriate set of more stable nodes. Therefore one topology of 20 nodes has been created using GRG (cf. Figure 8 above), than using the random waypoint

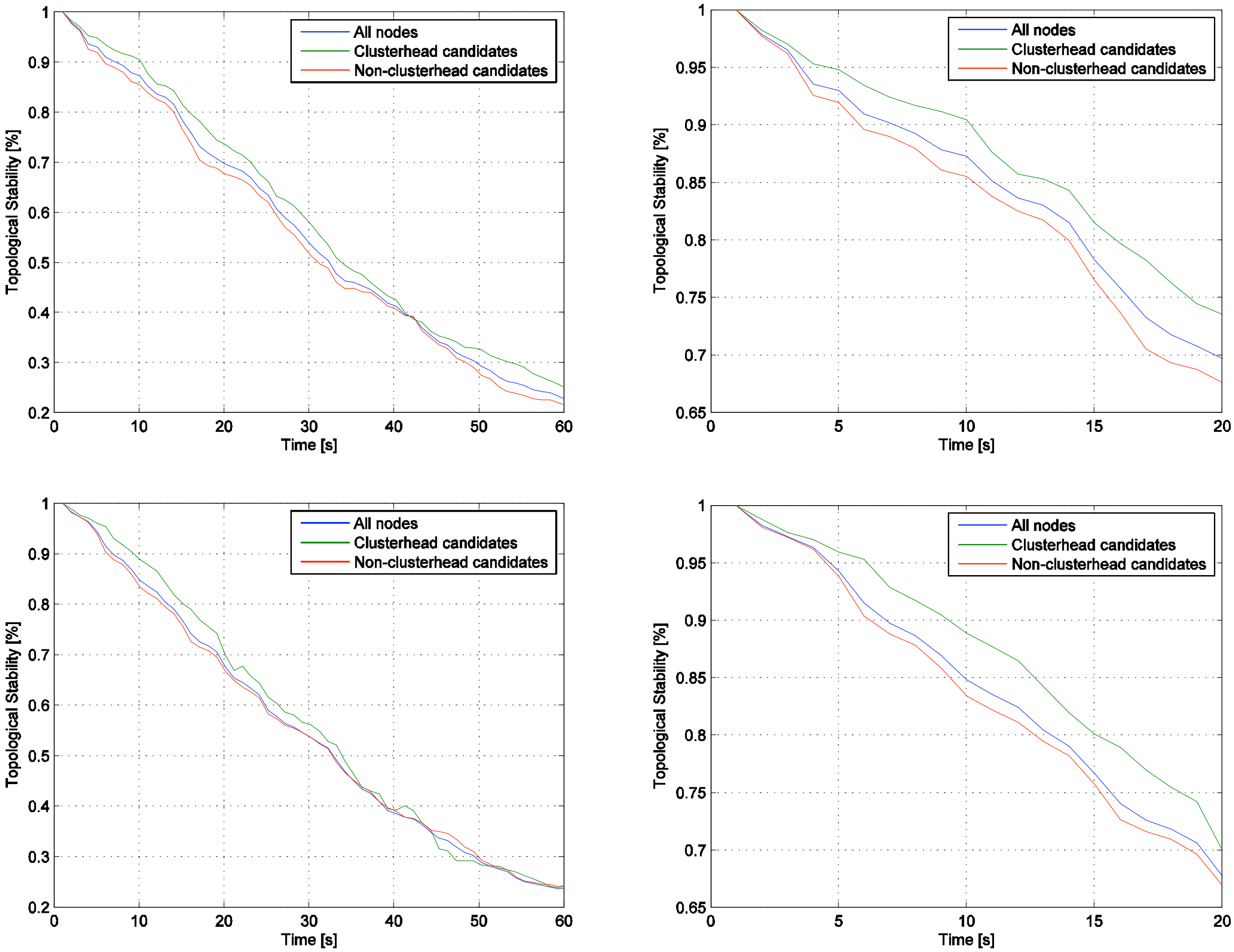


Figure 8. Measuring the topological stability of clusterhead candidates, non-clusterhead candidates and all node. One topology (NGM) with 20 devices is taken and 20 runs are realized (above). In further experiment, 20 different topologies (NGM, 20 devices) are used. 20 runs are realized (below).

model as described above. The results are averaged over 20 runs.

Figure 8 (above) show that the clusterhead candidates perform better than other nodes the first 40 seconds. Here the topological stability is prolonged between 1 and 3 seconds. As consequence of this experiment, we found that there are topologies with a set of nodes that tend to be more stable to mobility than others are.

Would this result also hold for a topology family, i.e. producing randomly topologies with same setting (e.g. 20 nodes, NGM, sending range 50 m)? Therefore, we produced randomly 20 topologies and used the random waypoint model as described and measured the topological stability.

Figure 8 (below) shows a similar functional behavior as Figure 8 (above). The conclusion is that the presented heuristic performs well not only for one selected topology with 20 nodes (NGM), but as well for the class of topologies with 20 nodes (NGM).

How do the initial topology and the number of devices influence this performance? In order to answer this question, we conducted experiments for all setting described in Figure 7. For the experiments, we used a different topology for each run and conducted each experiment 20 times. The clusterhead candidates are steadily performing better than the non-clusterhead candidates, but with a higher number of devices, the performance improvement is dramatically reduced for both GRG and NGM. Results for 60, 80 and 100 devices show the same tendency whereby the NGM still performs reasonable with 60 and 80 devices, there is no performance improvement recognizable when using GRG with 60, 80 or 100 devices.

[C3] *Robustness to clusterhead failures*. The algorithm detects bridge nodes. When failing, these nodes are responsible for network partitioning. The algorithm uses two-hop information for classifying nodes. That means, that the elements of the set of clusterhead candidates cannot cause partitioning when failure. Informally speaking, any arbitrary element of the set of clusterhead candidates can be removed from the topology without splitting it into partitions. Observe that this does not hold if removing more than one element. However, in our assumption, device failures occur occasionally and devices are recuperating (e.g. re-initiated by the user or sensor' batteries are recharging with solar or kinetic energy). Even though, first experiments show that there is a high tendency for the network not to partition by removing an increased number of clusterhead candidates. Observe that this is a work-in-progress report and that extended experiments have to be done.

B. Discussion

Our work represents a method to gather a set of efficient clusterhead candidates in respect to the criteria C1-C3. It is not part of this work to describe how to form clusters. Therefore, the metrics chosen are not based on clusters, but on the whole network. In a next step, we aim to implement different clustering mechanisms and compare the performance of the cluster as well as the clusterhead behavior.

As already mentioned, the thresholds used in pseudo code in Figure 2 are based upon tests we conducted. A reasonable threshold for the weak nodes classification is between 0.3 and 0.4 for the used types of topologies. If the threshold is set above 0.5, a high number of nodes that represent appropriate clusterhead candidates are classified as weak nodes. The best threshold value depends on the model used for topology generation.

Border nodes detection works well in cases where the node density is high enough to figure out an appropriate reference node. On the other side, if the agglomerate of nodes is geographically very dense, the distances are too short for an appropriate reference node (cf. Figure 4). The poor results for experiments with 60, 80 and 100 devices are also an expression of inappropriate border node detection that failed because of the used reference node. We affirm this because experiments with an area of 650 m×650 m, transmission range 80 m and 120 devices performed similarly well as results shown in Figure 8.

Note that the shortest path improvement as shown in Figure 7 is between 2.70 % and 14.17 %. In some cases where a bridge node is connected on its left and right to the same characteristic topology, the exclusion of bridge nodes causes a conflict with C1. In that case, a bridge node represents the best choice for information dissemination, but does not guarantee connectivity in case of failure (cf. C3).

We now discuss the effect of prolonged topological stability in a dynamic environment. Let us assume that an update policy is designed to invoke the cluster re-election process after the topological stability drops under 75 %. Practically we can say that using our approach in the case of 20 devices the update policy will be executed between 3 and 4 seconds later. Instead of invoking re-clustering after 15 seconds, it will be invoked after approx. 19 seconds. Summarizing, a stabilizing effect of 26 % was gained by choosing appropriate clusterheads. Thus, the update policy of a clustering algorithm can reduce the message complexity up to 26 % when taking the topological characteristics of one node into account. Observe that there are additional message exchanges caused by the algorithms as well as an additional use of computing resources that we expect to be considerably smaller than the decreased number of message exchanges caused by immoderate clusterhead re-elections. The algorithms' properties and message complexity have to be analyzed for this.

We showed there is a set of nodes that are more stable to mobility than other nodes. The results indicate that there is an inherent characteristic of these nodes independently of a fixed topology, but probably dependent on the type of topology. Two questions arise: (a) What are these characteristics exactly? (b) Assuming that we could apply the optimal policy (i.e. choosing the best possible selection of nodes), how big is the theoretical maximum?

Interestingly, the first series of experiments show better performance for the path length when using the NGM model. On the other side, when dealing with topological stability the performance is better using the GRG, in particular, when using more than 40 devices. Therefore, it is a consequence to ask for results for more topology models, e.g. continuum percolation [20] or occupancy theory.

We see a main drawback of the algorithms presented in the two-level approach. The approach may lack in a fully dynamical environment where node mobility is set to be very high and it might be difficult to execute each phase in a way that a consistence set of nodes is determined. Further experiments have to show if this really is a drawback.

VII. Conclusions and Future Work

This paper is a work-in-progress report on topological criteria for an appropriate clusterhead candidates selection in wireless networks. We present heuristic algorithms for these criteria that find candidates of potential clusterheads. Our approach is to classify nodes into weak, bridge, border nodes and clusterhead candidates. A clusterhead can be elected from the candidates set by a clustering algorithm that is not scope of this work.

The results showed that the introduced algorithms tend to comply with the following three criteria: (1) Fostering efficient information flow, (2) prolonged topological stability under the presence of mobility, (3) robustness to sudden clusterhead failures. The algorithms described work locally with two-hop neighbor information and without any use of geographical data.

An interesting result was the existence of a set of nodes in any topology that is more insensitive to mobility than the other nodes. Our experiments showed that the important characteristics of those favorable nodes are inherent to the topologies of our models. If this circumstance holds for any topology model is subject to further investigations and the results have direct impact on routing protocol design as well as on the design of clustering algorithms in ad-hoc and sensor networks.

A future challenge is to identify the characteristics that make a node more stabilizing in the presence of mobility than other does. In our model, we supposed that network nodes fail occasionally. We plan to conduct extended simulations to investigate the robustness of the network against clusterhead candidates increased failure rates, i.e. when an increased number of devices fail more often. Additionally, the algorithm properties and message complexity have to be analyzed. We plan to compare our algorithms especially with centralized and distributed clusterhead selection algorithms, which use geographical information and distances between the nodes.

REFERENCES

[1] O. Dousse, P. Thiran, and M. Hasler, "Connectivity in Ad hoc and Hybrid Networks," presented at INFOCOM'02, 2002.

[2] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*: Wiley, 2005.

[3] D. Peleg, *Distributed computing: a locality-sensitive approach*: Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2000.

[4] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, and C. Mit, "An application-specific protocol architecture for wireless microsensor networks," *Wireless Communications, IEEE Transactions on*, vol. 1, pp. 660-670, 2002.

[5] K. Dasgupta, K. Kalpakis, and P. Namjoshi, "An efficient clustering-based heuristic for data gathering and aggregation in sensor networks," *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 3, pp. 1948-1953, 2003.

[6] H. G. Luo, F. G. Ye, J. G. Cheng, S. G. Lu, and L. G. Zhang, "TTDD: Two-Tier Data Dissemination in Large-Scale Wireless Sensor Networks," *Wireless Networks*, vol. 11, pp. 161-175, 2005.

[7] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," *Proceedings of the 10th ACM conference on Computer and communication security*, pp. 62-72, 2003.

[8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pp. 10, 2000.

[9] M. Chatterjee, S. K. Das, and D. Turgut, "A Weight Based Distributed Clustering Algorithm for Mobile ad hoc Networks," *Proceedings of the 7th International Conference on High Performance Computing, LNCS 1970*, pp. 511-524, 2000.

[10] I. Gupta, D. Riordan, and S. Sampalli, "Cluster-head election using fuzzy logic for wireless sensor networks," *Communication Networks and Services Research Conference, 2005. Proceedings of the 3rd Annual*, pp. 255-260, 2005.

[11] H. Tan, W. Zeng, and L. Bao, "PATM: Priority-Based Adaptive Topology Management for Efficient Routing in Ad Hoc Networks," presented at International Conference on Computational Science, 2005.

[12] A. Andronache, M. R. Brust, and S. Rothkugel, "Multimedia Content Distribution in Hybrid Wireless using Weighted Clustering," presented at 2nd ACM Workshop on Wireless Multimedia Networking and Performance Modeling, Malaga, Spain, 2006.

[13] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "On the locality of bounded growth," in *PODC '05: Proceedings of the twenty-fourth annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing*: ACM Press, 2005, pp. 60--68.

[14] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "What cannot be computed locally!" in *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*: ACM Press, 2004, pp. 300--309.

[15] M. Penrose, *Random Geometric Graphs*: Oxford University Press, 2003.

[16] M. Kaiser and C. C. Hilgetag, "Spatial Growth of Real-World Networks," *Physical Review E*, 2004.

[17] M. C. M. Chatterjee, S. Das, and D. C. M. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks," *Cluster Computing*, vol. 5, pp. 193-204, 2002.

[18] D. J. Watts and H. Strogatz, "Collective Dynamics of 'Small World' Networks," *Nature*, vol. 393, pp. 440-442, 1998.

[19] S. Rothkugel, M. R. Brust, and C. H. C. Ribeiro, "Inquiring the Potential of Evoking Small-World Properties for Self-Organizing Communication Networks," presented at 5th International Conference on Networking (ICN 06), Mauritius, 2006.

[20] I. Glauche, W. Krause, R. Sollacher, and M. Greiner, "Continuum percolation of wireless ad hoc communication networks," *Arxiv preprint cond-mat/0304579*, 2003.