Dynamic Multi-Hop Clustering for Mobile Hybrid Wireless Networks

Matthias R. Brust  
University of Luxembourg  
Faculty of Science, Technology and Communication  
Luxembourg  
matthias.brust@uni.lu

Hannes Frey  
University of Southern Denmark  
Department of Mathematics and Computer Science  
Denmark  
frey@imada.sdu.dk

Steffen Rothkugel  
University of Luxembourg  
Faculty of Science, Technology and Communication  
Luxembourg  
steffen.rothkugel@uni.lu

ABSTRACT
In mobile wireless networks communication is often improved by sending messages along a stable backbone of more reliable communication paths. Building such a backbone requires efficient clustering algorithms which aggregate network nodes into logical groups, each group being managed by a clusterhead and any two neighboring clusters being interconnected by at least one gateway node or gateway path. In this concept k-hop clustering refers to cluster structures where cluster members are at most k hops away from their clusterhead. While the dynamicity of mobile wireless network is often considered as a challenge, in this work we explicitly exploit node mobility in order to support cluster formation and maintenance of k-hop clusters. The described KHOPCA algorithm consists of a set of easy to implement rules which form and maintain k-hop sized clusters in a purely localized way. In a static network cluster formation is limited to a constant number of messages exchanges among neighboring nodes. In dynamic networks the localized nature of the described rules promise a fast cluster convergence and low communication complexity in case of mobility triggered cluster reconfiguration.

1. INTRODUCTION
Multi-hop ad-hoc networks are composed of a collection of devices that communicate with each other over a wireless medium [1]. Such a network can be formed spontaneously whenever devices are in transmission range. Joining and leaving of nodes occur dynamically, particularly when dealing with mobility in ad-hoc networks. Potential applications of such networks can be found in traffic scenarios, environmental observations, ubiquitous Internet access, and in search and rescue scenarios as described in detail in [2].

Multi-hop ad hoc networks are composed of a collection of devices that communicate with each other over a wireless medium [1]. Such a network can be formed spontaneously whenever devices are in transmission range. Joining and leaving of nodes occurs dynamically, particularly when dealing with mobility in ad hoc networks. Potential applications of such networks can be found in traffic scenarios, environmental observations, ubiquitous Internet access, and in search and rescue scenarios as described in detail in [2].

In ad hoc 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. In mobile network, however, where nodes join and leave continuously cluster formation/maintenance and clusterhead election appears as a challenge [8].

A k-hop clustering may be used as a resource efficient backbone when joining two different wireless technologies into a single hybrid wireless network structure, e.g. in a hybrid ad hoc/UMTS network. Ad hoc network links are free of charge while some "expensive" UMTS links can be used in order to reliably interconnect the cluster heads. This will reduce the overall cost compared to purely UMTS based communication, while on the other hand short path interconnects between any two
communication end points will improve reliability compared to purely ad hoc network based communication.

In this context, our contribution in this paper is to describe KHOPCA (K-HOP Clustering Algorithm), a k-hop clustering algorithm where k represents the maximal number of hops from the clusterhead to the cluster border. The KHOPCA is highly adaptive in respect to mobility and works with local 1-hop neighborhood information only, asynchronously, and fully distributed. In summary, this paper introduces KHOPCA, a novel k-hop clustering algorithm. Furthermore, we analyze its correctness as well as its empiric performance.

2. BIBLIOGRAPHIC NOTES

Clustering algorithms can be based on criteria such as energy level of nodes, their position, degree, speed and moving direction. Centralized and distributed approaches can be distinguished, as well as probabilistic and deterministic ones. In general, clustering algorithms are evaluated on the average number of clusterheads they produce. In mobile environments, the stability of clusterheads is of particular interest. Fully local (localized) ones are preferred when taking the message complexity into consideration. The second main aspect of clustering is how efficient clusters support inter- and intra-cluster communication, e.g. to apply routing algorithms on a clustered network. In this ongoing work, our focus is on the cluster structure, message complexity, and—in particular—on the number of clusterheads. The following gives an overview about distributed and weight-based clustering algorithms.

In [9] a weighted application aware clustering algorithm is introduced (WACA). WACA is designed to operate in a hybrid wireless environment. WACA fosters efficient information dissemination within the ad hoc neighborhood as well as it limits the use of uplinks to the backbone network. The algorithm uses weights that are calculated using a heuristic weight function. The weights of WACA are designed to be explicitly application driven. A corresponding weight function is presented and discussed. According to the weight function, clusterheads are devices with highest weight in their local neighborhood. Through the local approach of WACA, situations appear where a node is considered as clusterhead by some local neighbors, but the node itself considers a different node as clusterhead because of a higher weight. These intermediate nodes are called sub-heads. The simulation studies conducted are based on static as well as mobile network topologies.

Dow et al [10] present a comparison study of five types of clustering algorithm that are the highest connectivity algorithm (HD), lowest ID algorithm (LI), least cluster change algorithm (LCC), weighted highest degree algorithm (WHD), and the DMAC algorithm. The highest connectivity algorithm declares a node with the highest number of neighbors as clusterhead while the lowest ID algorithm uses the lowest ID as election criterion. The least cluster change algorithm can be compared with the king bonus of WACA. Both serve to stabilize clusterhead changes. The WHD algorithm deduces re-clustering when two clusterheads come into transmission range.

Nocetti et al [11] describes k-hop clustering algorithms for mobile ad hoc networks. They use a priority weight based approach whereby they propose to use connectivity and lower ID as clusterhead election criteria. The study reports on results for k = 1 and k = 2.

Fernandez et al [12] defines formally the problem of k-hop clustering (or k-clustering) with the objective to find optimal clusters. The minimum k-clustering problem is described as: Given an unit disk graph denoted by $G = (V, E)$ and a positive integer $k$, find the smallest value of $l$ such that there is a partition of $V$ into $l$ disjoint subsets $V_1, \ldots, V_l$ and $\text{diam}(G(V_i)) \leq k$ for $i = 1, \ldots, l$. Fernandez et al point out that k-clustering is NP-complete for simple undirected graphs. The presented algorithm deals with mobility and consists of two phases: Constructing a spanning tree and partitioning the spanning tree into k bounded sub-trees. However, the algorithm has polynomial time and message complexity.

The WCA algorithm [13, 14] uses geographical information and node speed for calculating the weight. Although locally working, WACA allows the formation of multi-hop clusters. Similar to WHD, KHOPCA forces a re-clustering when two clusterheads are in transmission range. In contrast to the algorithms described here (except [12] and [15]), KHOPCA allows more than 2-hops between two arbitrary nodes in a cluster. That is, because the objective of the KHOPCA is to support 2k-size clusters. The WACA algorithm allows clusterhead (or subhead) chains. However, the maximum cluster size is not well manageable, i.e. the size of the cluster and, thus, the number of clusterheads cannot be restricted as it is the intention of KHOPCA. The k-hop clustering algorithm presented in [15] is analyzed for $k = 1$ and $k = 2$, while KHOPCA is investigated for $k = 2, \ldots, n$.

3. KHOPCA: k-HOP CLUSTERING

Inspired by Conway’s “Game of Life” [16], we though on dealing with a spatially organized ad hoc network as a kind of cellular automata, not using a grid but a graph as underlying topology. In order to completely avoid multi-hop broadcasting for cluster formation, the idea was to create a set of rules that describes the transition between the state of a node solely depending on the current state of it neighbors.

Each node is continuously involved in the process of k-hop cluster formation and cluster maintenance. KHOPCA does not require any predetermined initial configuration. A node can potentially choose any role, i.e. weight $w_i$, in the beginning as well as at any time.

In the start configuration, a participating node knows the clusterhead weight, i.e. the weight when a node is assumed to be a clusterhead, and the minimum weight. These weights are named MIN and MAX. We assume that the only information available for a node $v$ is the information of the 1-hop neighbors $N(v)$ and the corresponding weights $W(N(v))$. In practice, the beaconing is most appropriate to provide this information. The KHOPCA clustering process explicitly supports joining and leaving of nodes. Provided information conforms the definition of a 1-local algorithm. Following rules describe the state transition for a node $v$ with weight $w_v$:

1. If there is a neighbor with a highest weight $w_u > w_v$ of all neighbors then node $v$ changes it weight to $w_v \leftarrow w_u - 1$.
2. If weight $w_v = \text{MIN}$ and $w_v = w_u$, $w_u \in W(N(v))$, i.e. no neighbor has a higher weight than $\text{MIN}$ then node $v$ declares itself as clusterhead, thus $w_v \leftarrow \text{MAX}$.
3. If \( w_v \neq \text{MAX} \) and \( w_v \geq w_w \) \( \forall w_w \in W(N(v)) \), i.e. there is no neighbor with a higher weight then \( w_v \leftarrow w_v - 1 \).

4. If \( w_v = \text{MAX} \) and \( 3w_v \in W(N(v)) \) with \( w_v = \text{MAX} \), i.e. two clusterheads appear in transmission range, then decrement the weight of one clusterhead to weight \( w = w - 1 \).

Informal, the first rule says that a node copies the highest neighbor weight subtracted by one. This is important to create a top-to-down hierarchical structure. The second rule deals with the situation where isolated nodes are clusterhead-less on the minimum weight level. In that case a node declares itself as clusterhead. The third rule covers situations where higher weight nodes that are not clusterhead attract surrounding nodes with less weight. In order to avoid a fragmented cluster, this node successively decreases its weight, finally connecting to an existent cluster. The fourth rule describes the situation where two clusterhead meet in 1-hop neighborhood. In accordance of a criterion one clusterhead survives while the other clusterhead dies. In case of rule 4, for deciding which node continues as clusterhead, a unique ID can be used in order to resolve this conflict. An exemplary sequence of weight transitions applying the KHOPCA-rules is illustrated in Fig. 1. Note that the figure depicts only one outcome of the different ones which are possible due to asynchronicity.

![Figure 1. A possible sequence from an initial starting weight of 3 whereby the k is set to 5. In the static case, this example terminates after 9 steps. In a mobile environment, however, the cluster structure will change continuously in the way new neighbors arrive or old neighbors leave.](image-url)
weight of only change by at most a decrement of 1. Moreover since $S_{i-1}$ is no longer changing the condition of rule 3 will never be satisfied, i.e., weight $w_u$ will never change again.

All neighbors of a given node $u \in T_i$ will either be in $S_i$ or in $T_{i-1} \setminus S_i$. Nodes in $S_i$ have a weight of $MAX - i$ and nodes in $T_{i-1} \setminus S_i$ have a weight less than $MAX - (i - 1)$. It follows that the weight values of any node $u$'s neighbor node $v$ will satisfy $w_v \leq MAX - i$. Thus, node $v$ will either set its weight to $MAX - i - 1 = MAX - (i + 1)$ by rule 1 or set its weight to a value less or equal than $MAX - (i + 1)$ due to rule 3. In both cases we have that the weight of a node $u$ in $T_i$ always satisfies $w_u < MAX - i$.

In summary, we have that after a finite number of algorithm execution steps $S_i$ and its nodes’ associated weights will no longer change. Moreover, since minimum hop distance is always limited by $n$, the number of network nodes, $S_i$ will be empty for any $i > n$. Thus, at some point in time $t$ all sets $S_i$ and their nodes’ associated weights will no longer change. Since $S_i$ is the set of all network nodes, there are no more weight changes after time $t$.

**Theorem 2**: The topology constructed after KHOPCA terminated always satisfies the following conditions:

- A node $u$ with weight $w_u = MAX$ has no neighbor $v$ with weight $w_v = MAX$.
- A node $u$ with weight $w_u < MAX$ has a neighbor $v$ with weight $w_v = w_u + 1$.
- The weight difference between any two neighbors $u$ and $v$ always satisfies $|w_u - w_v| \leq 1$.
- The weight $MAX - w_u$ of a node $u$ reflects the minimum number of hops to reach a clusterhead from $u$.

**Proof**: The first property is an immediate consequence of rule 4.

For proving the second property assume that a node $u$ has no neighbor with weight $w_v = w_u + 1$. If there is a neighbor $v$ with a highest weight $w_v > w_u + 1$ then due to rule 1 node $u$ will set its weight to $w_u + 1$ which is greater than the old weight value of $u$. If all neighbors $v$ have a weight $w_v \leq w_u$ then node $u$ will change its weight to $w_u + 1$. In both cases node $u$ will change its weight which contradicts the assumption that KHOPCA already terminated.

For proving the third property, assume that two neighboring nodes $u$ and $v$ satisfy $|w_u - w_v| > 1$. Without loss of generality let $w_u < w_v - 1$. It follows that node $u$ has a neighbor $v'$ with a highest weight $w_{v'} \geq w_v$. Due to rule 1 node $u$ will set its weight to $w_v - 1$ which is greater than the old weight value $w_u$ of $u$. This contradicts the assumption that KHOPCA already terminated.

For proving the fourth property assume that there exists a clusterhead $v$ which can be reached from $u$ with $l < MAX - w_u$ hops. Let $u = u_1 u_2 \ldots u_{l+1} = v$ be such an $l$-hop path connecting $u$ with $v$. Node $v$ being a clusterhead has weight $w_v = MAX$. Due to property 3 the weight from $u_1$ to $u_{l+1}$ can only change by at most a decrement of 1. It follows that the weight of $u$ must satisfy $w_u \geq MAX - l$. This contradicts $w_u < MAX - l$ which follows from the initial assumption $l < MAX - w_u$. 

Because of its weight, a node is aware of its hop-distance to the closest clusterhead. The KHOPCA clustering algorithm even guarantees that a node’s weight represents the optimal hop-distance to the next closest clusterhead. If there appeared a node with a weight $w_v + 2$ in the neighborhood of node $v$, node $v$ would change its weight to $w_v + 1$ in the next phase. Hence, a node changes to the optimal weight that represents the shortest path to the closest clusterhead.

![Figure 2. A KHOPCA starting up weight deployment applied to a strong hierarchical cluster topology is shown. After a stabilization phase the cluster size is grown and the number of clusterheads decreased (example for $k = 4$ and network density $k^* = 16.42$).](image)

![Figure 3. The percent of nodes that results in a clusterhead in respect to the network density and $k$-value.](image)

![Figure 4. Interpolation of the data from Figure 3.](image)
Table 1. The number of clusterheads stabilize with time, illustrated by sparklines. Clusterheads decrease considerably in networks with high densities and a higher cluster size, e.g. for $k = 4, 5, 6, 7$. The min and max values are given in percent.

<table>
<thead>
<tr>
<th>k</th>
<th>$k^*$ = 4.71</th>
<th>$k^*$ = 9.91</th>
<th>$k^*$ = 16.42</th>
<th>$k^*$ = 25.29</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>22</td>
<td>15</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>12</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>16</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>25</td>
<td>12</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>26</td>
<td>12</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>22</td>
<td>12</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

5. EVALUATION

5.1 Simulation Settings
Transmission range $r$, simulation area $A$, and number of devices $n$ are frequently used simulation parameters. In the following we utilize the notion of network density in order to join those parameters into a unique framework which identifies classes of network topologies with similar characteristics. Assuming a uniform node distribution the average network density $k^*$, i.e. expected number of nodes in a node's sending range including the node itself, can roughly be estimated as $k^* = \pi \cdot r^2 \cdot n/l^2$.

Experiments were conducted for values $k = 3, 4, 5, 6$ and 7, and for network densities between 4.71 and 25.29. Networks with densities around 4 are considered as sparse since full connectivity is very unlikely to happen. Networks with higher densities (6 and higher) are considered as dense. Thus, experiments covered a wide range of networks. Initially, weight $MIN$ was assigned to all nodes. For mobility simulations, the random direction mobility model was applied since nodes are distributed uniformly in the simulation area over time, avoiding known drawbacks of the random waypoint mobility model [17]. Node speed was chosen to be of pedestrian moving speed of 1.5 m/s. Exemplary KHOPCA topologies are shown in Fig. 2.

5.2 Simulation Results
In the KHOPCA implementation, the number of clusterheads is decreased when they appear in the direct neighborhood. In cases where the clusterheads are 2-hop away from each other, physically connected by an intermediate node, the clusterhead continue alive. For situation where $k$ corresponds a higher value (e.g. 4 or more) this circumstance works against the cluster $k$-growing process. Allowing that a clusterhead ID is submitted to cluster members, enables intermediate nodes as described, to discover conflicting situations and serve as relay nodes for clusterhead reduction. This method is local, but causes additional control messages. The ID has to be forwarded from the clusterhead to its cluster members. The weight calculation, however, works instantly.

The number of clusterheads stabilization process is reported in Table 1 using sparklines. Two tendencies can be observed. First, for low $k$-values as well as for low densities the warming up phase for KHOPCA is very low. Second, for high $k$-values in combination with high densities KHOPCA requires a perceivable warming up phase until the number of clusterheads is stabilizing. The duration depends on the node mobility, because clusterheads have to meet in order to decide which clusterhead will survive and to invoke re-clustering.

The number of clusterheads in respect to network density and $k$-value is illustrated in Figure 3 and Figure 4. Figure 3 reveals the clusterhead reduction potential when increasing $k$. Given a network density, increasing the cluster radius $k$, the number of clusterheads is asymptotically falling. Figure 4 generalizes the results of Figure 3 and interpolates from given network densities to network densities between 4.71 and 25.29. Figure 4 informs about the possible number of clusterheads when using a given network density and hop count. For example, if $k$ is set to 4, KHOPCA does not provide a network with density between 4.71 and 25.29 to have approx. 20% clusterheads.

6. CONCLUSIONS
A $k$-hop clustering algorithm (KHOPCA) is described where $k$ represents the maximal number of hops from the clusterhead to the cluster border. The KHOPCA algorithms is shown to be highly adaptive to mobility and works with local 1-hop neighborhood information only, asynchronous, and fully distributed.

The main contribution in this paper is the finding that clusters with a diameter of $2k$ can be efficiently created by local information only. It is shown that mobility essentially supports the self-organization of $k$-hop clusters. Experiments conducted prove a controllable clusterhead reduction. Finally, some fundamental KHOPCA topology and clustering properties have been proven.

7. REFERENCES


