

LSWTC: A Local Small-World Topology Control Algorithm for Backbone-Assisted Mobile Ad hoc Networks

Matthias R. Brust*, Carlos H.C. Ribeiro*, Damla Turgut[†], Steffen Rothkugel[‡]

*Computer Science Division

Technological Institute of Aeronautics, Brazil

Email: {matthias.brust,carlos}@ita.br

[†]School of EECS, University of Central Florida

4000 Central Florida Avenue, Orlando FL 32816

Email: turgut@eecs.ucf.edu

[‡]Faculty of Science, Technology and Communication (FSTC)

University of Luxembourg, Luxembourg

Email: steffen.rothkugel@uni.lu

Abstract—The prevalence of the small-world phenomenon in numerous efficient networks, such as social networks, Internet, nervous systems, implies that small-worlds are an evolutionary solution for locally growing networks. These networks require short communication distances between their nodes in spite of their potentially large network diameter but at the same time are robust against randomly occurring failures. Integrating these properties into human-made communication networks drastically increases their efficiency and performance but represents an enormous challenge in the case of mobile ad hoc networks since their communication graphs rely on local construction principles.

In this paper, the focus is on backbone-assisted ad hoc networks. In such networks devices connect arbitrarily to other devices within their transmission range, and dedicated devices are able to connect to a backbone network. This construction principle leads to a geometric graph that has a small average path length, but that is sensitive to random attacks or failures. The challenge in evoking small-world properties in backbone-assisted mobile ad hoc networks is to build a topology control algorithm that works with localized data (i.e. using neighboring information) despite knowledge of average path length and clustering coefficient requiring global information. Such a topology control algorithm is described here. Empirical results show that depending on network density, small-world properties can be efficiently evoked.

I. INTRODUCTION

Small-worlds are characterized by a small average path length and a high clustering coefficient. The average path length *APL* is the value of the shortest path length expected between any pair of nodes in a network. The clustering coefficient *CC* measures the degree of how locally clustered nodes are in a network [1]. The advantage of a short average path length is to have few hops between any pair of nodes. Networks with short chains of communication are able to spread information more efficiently. In the area of mobile ad hoc networks, a small path length is important for routing mechanisms and the overall communication performance of the entire network [2].

While the impact of a short average path length on the network performance is an intelligible advantage, a high clustering coefficient reveals its hidden benefits only through the looking glass. One aspect becomes clear when considering the problem of disease spreading in small-world networks. Let us consider a population (nodes) of which only a certain fraction *q* is susceptible. When the disease starts spreading, it can only be transmitted and only affects susceptible nodes. How large *q* must be for the epidemic takes place? It has been shown that $q_{small-world} \gg q_{random}$ [3]. Since a small-world network is locally highly clustered, the disease can only spread within the connected cluster of susceptible nodes in which it first starts, making a small-world network a robust network against the spreading of disease and failure. Additionally, in a high clustered network, a randomly occurring failure is unlikely to increase the average path length, thus keeping the average path length not only low but also more stable. This makes a small-world network with its high clustering coefficient more tolerant against sudden failures or randomly occurring errors. Thus, removing certain links i.e. communication resources, can increase the network robustness [4].

A small but also difficult to increase average path length under randomly occurring failures are two attractive properties for self-organizing communication networks. However, in spite of being present in so many real-world networks, small-world properties cannot be taken for granted.

Evoking these properties in human-made communication networks such as backbone-assisted ad hoc networks [5] or wireless sensor and actor networks (WSANs) [6] drastically increases their efficiency and performance. However, it also represents an enormous challenge for these wireless networks since the communication graph relies on local construction principles and mostly unpredictable network connections due to mobility. Devices connect spontaneously to devices within

their limited transmission range. Special devices, however, are able to connect to a backbone network. Backbone networks are characterized by their capability to provide QoS. From theory perspective, this construction principle builds geometric graphs that have a small average path length, but are sensitive to random attacks or failures. With the backbone network, the graph-theoretical model is extended by special shortcuts that can connect long-distant devices. These links, however, are limited and imply additional costs and higher energy consumption. A further complicating factor for using these shortcuts is that the backbone is assumed to know only a small fraction of the nodes in the network.

Because of the option to dynamically control such links, the network topology basically can be modified as needed, opening the doors for compensating the disadvantages caused by the local construction model and the dynamics of spontaneous link failures and device mobility.

In this paper, two locally performing topology control algorithms are combined for adaptively manipulating the topology, maintaining small-world properties in a dynamically changing, mobile and unpredictable wireless network. Reckful Roaming (RR) [7] increases the clustering coefficient and a k -hop clustering algorithm (KHOPCA) [8] aims to indirectly decrease the average path length. The combination of RR and KHOPCA results into one synergetic acting topology control algorithm called LSWTC which simultaneously optimizes clustering coefficient and average path length with local information.

The results show that depending on the density of the destination network, small-world properties can be evoked and maintained even in presence of nodes mobility and the dynamics of the network topology.

The remainder of this paper is organized as follows. The subsequent section reviews background on small-world and topology control in wireless networks related to this work. Section 3 describes the system model used in this paper. In Section 4, the LSWTC is introduced. Properties and results are presented in Section 5 and Section 6 concludes the paper.

II. BACKGROUND

A. Small-Worlds

The local clustering coefficient CC_i of a node i with k_i neighbors is $CC_i = \frac{|E(\Gamma_v)|}{k_i(k_i-1)}$ where $|E(\Gamma_v)|$ is the number of links in the neighborhood of i and $k_i(k_i-1)$ is the total number of possible links in the neighborhood of v .

The (global) clustering coefficient CC of a graph $G = (V, E)$ is then the average of all local clustering coefficients in the network denoted as $CC = \frac{1}{n} \sum_i CC_i$ where n is the number of nodes in G denoted as $n = |V|$. A high CC means that the network consists of a high number of nodes that are locally clustered, i.e. CC reflects the probability that a randomly chosen pair of nodes $v_1, v_2 \in V$ that are connected $(v_1, v_2) \in E$ have a mutual neighbor $v_3 \in V$ with $(v_1, v_3) \in E$ and $(v_2, v_3) \in E$. The clustering coefficient measure can be understood as the localization of the measure edge density that

divides the number of existing links in a network by the number of possible links denoted as $d = \frac{|E|}{n(n-1)}$.

The average path length APL is the mean of the means of all the shortest path lengths connecting each node to all other nodes. That is, $\forall v_1, v_2 \in V$ find $d(v_1, v_2)$, i.e. the shortest distance between v_1 and v_2 . Then, calculate the average path length APL as $\frac{1}{n(n-1)} \sum_{i,j} d(v_i, v_j)$. The clustering coefficient reflects local characteristics of a network while the average path length measures a global characteristic.

The communication graph model of backbone-assisted mobile ad hoc networks and WSNs rely on spatial connection constraints where only devices allowed to be connected within transmission range. One could argue that the backbone network can establish long-distance links. However, the backbone network is not aware of all the nodes in the network, but only of the devices able to connect to the backbone or the devices within the transmission range. A small-world network must show a specific correlation between average path length APL and clustering coefficient CC . A small-world graph is a graph with n vertices and an average degree k that exhibits an average path length $APL \approx APL_{Random}(n, k)$ but a clustering coefficient $CC \gg CC_{Random}(n, k) \approx \frac{n}{k}$ with $APL_{Random}(n, k)$ and $CC_{Random}(n, k)$ are the average path length and clustering coefficient of a random network with parameters n and k [1].

B. Topology Control and Clustering Algorithms

The area of Topology Control (TC) aims to tune the topology in order to increase efficiency in the network operation performance according to the optimization criteria such as battery lifetime, interference, noise. For these optimization criteria, a variety of different TC protocols have been proposed. Some of these protocols are related to this work and reviewed here.

The LMST (Local Minimum Spanning Tree) protocol [9] aims to find the minimum transmission range where all devices in the network are still connected. It approximates a minimum spanning tree with local knowledge using Prim's algorithm. The LMST protocol can be computed in a fully distributed and localized manner and it preserves connectivity in the worst case. The LMST protocol has a message complexity of $O(n)$.

The Cone-Based Topology Control (CBTC) protocol [10] uses as optimization criteria preserving connectivity and minimizing energy consumption by removing energy inefficient links. The CBTC protocol divides the sensed environment in cones of angle α which are technologically sensed by directed antennas denoted as $CBTC(\alpha)$. The CBTC protocol is fully distributed which is based on local directional information only and preserves network connectivity. CBTC has been extended to provide fault tolerance in means of guaranteeing k -connectivity [11].

The previously described topology control protocols LMST, CBTC and augmented CBTC consider connectivity and k -connectivity as criteria for fault-tolerance. Network robustness and fault-tolerance by keeping the average path length sensitivity and the susceptibility of the network low is not considered by any of the existing TC protocols.

Both, topology control and clustering algorithm build a more efficient communication topology. A short review about clustering algorithms relevant to this work follows.

Clustering algorithms can be based on criteria such as energy level of nodes, their position, degree, speed and moving direction.

The problem considered in this work requires an efficient control of the number of clusterheads. The dominating set theory is often applied for this kind of problems. A k -dominating set D_k is a subset of connected nodes that guarantees a maximal path length of k to any node in the network, while minimizing the number of nodes in D_k [12], [13]. The construction of the network backbone by D_k with shortcuts, however, is not feasible if neither the backbone nor the nodes themselves are aware of the available nodes in the network. For that reason, the dominating set theory must be rejected for tackling the problem of controlling the number of clusterheads regardless of the number of total nodes in the network. However, multi-hop clustering algorithms fulfill these criteria and are of particular interest in the context of this work.

In [14] a weighted application aware clustering algorithm is introduced (*WACA*). *WACA* is designed to operate in backbone-assisted mobile ad hoc networks and fosters efficient information dissemination within the ad hoc neighborhood as well as it limits the use of uplinks to the backbone network. However, the multi-hop structure is not controllable and limited with high probability to size 2.

In [15] a k -hop clustering algorithms for mobile ad hoc networks is introduced. A priority weight based approach is used whereby proposing to use connectivity and lower ID as clusterhead election criteria.

In [16] the problem of k -hop clustering (or k -clustering) is formally defined. The minimum k -clustering problem is described as: given a 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, \dots, V_l and $\text{diam}(G(V_i)) \leq k$ for $i = 1 \dots l$. k -clustering is *NP*-complete for simple undirected graphs. However, the algorithm has polynomial time and message complexity.

III. SYSTEM MODEL

A wireless network consists of a set of devices or sensors connected by wireless links. By use of technologies such as Bluetooth and Wi-Fi, these mobile devices are able to create communication links within their transmission range. Additionally, e.g. GSM- or UMTS-adapters can be employed to establish supplementary communication links between two devices by constructing a backbone network.

A. Ad hoc Communication Graph

The communication graph for ad hoc communication is constructed as follows.

Let $V \in \mathbb{R}^2$ be a set of nodes in the 2-dimensional bounded region with side length l . The nodes are deployed according to a deployment model D . The symmetric Euclidean

graph $G = (V, E)$, such that for any pair $u, v \in V$ of nodes $\text{dist}(u, v) \leq r \implies \{u, v\} \in E$ and $\text{dist}(u, v) > r \implies \{u, v\} \notin E$ is called a r -disk graph. The r -disk graph describes the graph for ad hoc communication between the devices.

The deployment model D follows a uniformly random distribution of the nodes in the bounded region.

Additionally, nodes are moving according to a uniformly assigned speed in a squared simulation area with length l .

B. Ad hoc-to-Backbone Communication Graph

Establishing backbone connections, additional cellular network links such as GSM/UMTS adaptors are available on some of the devices. Backbone links induce additional costs and higher energy consumption, connecting too many devices to the backbone network causes overload. Thus they must be minimized. Furthermore, these devices represent a bottleneck when a high number of ad hoc devices make use of them for accessing the backbone. A more complicating factor for optimizing these shortcuts is that the backbone is assumed to be aware only of a small fraction of the nodes in the network.

A backbone link is defined as a link (u, v) between nodes u and v whereby $u \notin E$. That is, a backbone link connects two nodes which are not in the same spatial neighborhood, i.e. within the same transmission range.

C. Backbone Communication Graph

Devices that have access to the backbone or are part of the backbone are considered to be fully connected. In context of the ad hoc communication of this paper, a backbone link is counted as one hop.

IV. THE LSWTC PROTOCOL

The *LSWTC* protocol consists of two parts: (a) the topology control algorithm, *Reckful Roaming (RR)*, that optimizes the clustering coefficient (*CC*) by removing links and (b) the clustering algorithm *KHOPCA* provides optimization for the number of access points to the backbone network, thus indirectly optimizing the average path length (*APL*). Both algorithms are discussed in this section.

A. Clustering Coefficient Optimization

The underlying system model of this work is conforming to the graph construction process of a r -disk graph. Hereby, the nodes detect their spatial neighboring nodes by their transmission signal with range r . To optimize the clustering coefficient, only operations allowed on the graph structure are either keeping or removing a link. The definition of the clustering coefficient intuitively induces that more links in each node's neighborhood result in a higher *CC*. For instance, in cases where a new link does not build a new triangle, the number of links in the neighborhood increases but the *CC* decreases. The introduction section contains a discussion how and why in certain circumstances a network with less communication resources and a higher *CC* has increased network robustness

and fault tolerance, so that the following procedure is a valid approach to the problem: a topology control algorithm called Reckful Roaming (*RR*) is introduced (Algorithm 1). Its main purpose is to verify locally the spatial-random link distribution in order to find dedicated links in which removal results in an increase of CC (see Fig. 1).

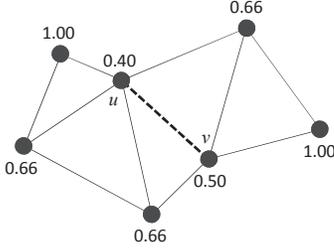


Figure 1: Negotiation between u and v , whether removing (u, v) increases the local clustering coefficient for both nodes.

After determining if the removal of a link $(u, v) \in E$ would result in an increase of CC , the link (u, v) is marked as removal candidate. It is not removed immediately because removing might not be in accordance with the criterion of both nodes. However, since a link removing affects the local clustering coefficients of the 2-hops neighborhood of $\{u, v\}$ an additional removal-confirmation phase must be performed.

In this confirmation phase, nodes exchange the removal-candidate with that corresponding neighbor. For example, if node v calculates that link (u, v) is a removal-candidate, it sends this result to node u . In case that node u has calculated the link (u, v) as removal-candidate as well, the final phase is executed; otherwise the link (u, v) remains. This agreement between two neighboring nodes is justified by the fact that removing a 1-hop link causes a considerably higher impact on the local CC -value than removing a 2-hops link. Since a link removal affects the CC -values of the 2-hops neighborhood of nodes $\{u, v\}$, the final phase calculates the 2-hops neighborhood of $\{u, v\}$. Both the current CC -values of the 2-hops neighborhood of (u, v) are calculated. Based on a comparison of those values, i.e. if the new CC -values are higher than before, the link (u, v) is removed, if fulfilling the condition that at least one neighbor of u is connected to one neighbor of v .

B. Average Path Length Optimization

The main question of optimizing the characteristic path length is how to limit the number of network-wide backbone links with local information only, in particular if the backbone is not aware of all available devices?

The proposed k -hop clustering algorithm (*KHOPCA*) where k (written as k_K to differentiate from the average node degree introduced above) represents the maximal number of hops from the clusterhead to the cluster border; recreates clusters with a certain size that includes an approximate number of devices, the network with n devices is partitioned in a number of clusters c with $c < n$. Since each cluster is assigned to one clusterhead,

and each clusterhead serves as access point to the backbone network, the number of backbone links is globally determined with local information only. The efficiency of *KHOPCA* is measured by its capacity to mainly reduce and to adapt the number of clusters according to the requirements.

The worst case scenario is that each device tries to connect to the backbone directly. However, the minimum requirement for a fully connected network is to have exactly one backbone link per network partition. Since partitions can be very large in size, it is not practical to manage exactly one backbone link per partition. However, by using multi-hop clustering, it is possible to find an intermediary solution: forming clusters with devices that are k -hop away and allow only the clusterhead to connect to the backbone (see Fig. 2).

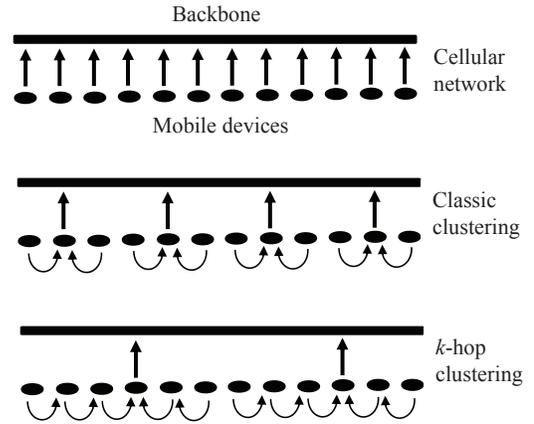


Figure 2: Reduction of the number of backbone links; k -hop clustering permits more control of backbone links than classic clustering with 1-hop neighbors.

Using this indirect optimization method, it is possible to control the number of clusterheads by augmenting the size of clusters. The larger the cluster size, the smaller the number of clusters and the number of clusterheads.

KHOPCA consists of a set of four rules that describes the transition between the states of a node solely depending on the current state of its neighbors (Algorithm 2).

Each node $i \in V$ chooses a weight w_i between 0 and k_K uniformly at random. In the start configuration, a node i is aware of (a) the weight of the clusterhead that is zero and (b) k_K that is the maximum number of hops that a node can reach its clusterhead. The only information available for a node v is the information of the 1-hop neighbors $N(v)$ and their weights $W(N(v))$ that can be provided by e.g. beaconing.

In the first rule a node copies the lowest neighbor weight added by one. This is important to create a hierarchical structure. The second rule deals with the situation where isolated nodes are without clusterheads on the maximum weight level k . In that case, a node declares itself as clusterhead by assigning itself the weight 0. The third rule covers situations where lower weight nodes that are not clusterhead attract surrounding nodes with more weight. In order to avoid fragmented clusters, these

nodes successively increase their weight with the objective to connect to an existent cluster. The fourth rule describes the situation where two clusterheads meet in neighborhood. In accordance of a criterion—usually the node ID or any other unique identification scheme—one clusterhead remains while the other clusterhead reassigns itself a weight of 1.

C. The LSWTC Protocol – an Interplay of RR and KHOPCA

The LSWTC protocol consists of two parts: (a) RR optimizes the clustering coefficient (CC) and (b) KHOPCA provides optimization for the average path length (APL) by adjusting the number of access points. A single interplay between both algorithms can be described formally as following:

$$\begin{aligned} G_{RR} &\leftarrow RR(G_{Init}); \\ G_{KHOPCA} &\leftarrow KHOPCA(G_{RR}); \\ G_{Final} &\leftarrow G_{RR}; \end{aligned}$$

RR removes links from the initial topology G_{Init} , resulting in G_{RR} . KHOPCA then creates an overlay network for clusterhead determination on top of G_{RR} . For multiple interplays, it is set $G_{Final} \leftarrow G_{RR}$. RR does not consider G_{KHOPCA} , indeed always acts upon G_{RR} . KHOPCA itself does not remove any links from the neighboring lists of the devices. It determines a structure for finding efficient paths to the closest access point. Changes by RR affect the results of KHOPCA, since RR removes neighboring devices from the neighboring list.

The description of the interplay between RR and KHOPCA is synchronous, while in fact the nodes are not synchronized since they are not aware if their neighboring nodes finished creating G_{RR} in the same round. Due to the influence of mobility, the topology changes permanently, resulting in a fundamentally asynchronous target environment.

Therefore the difference between synchronous and asynchronous method have been taken into account: (a) Awerbuch [17] showed that any synchronous algorithm (i.e. algorithm for synchronous network) can be transformed in its asynchronous counterpart by using synchronizers. (b) Results below show that running LSWTC without synchronization can indeed produce small-world properties. The synchronization problem of RR itself is discussed below.

An additional issue to consider is the connectivity on the backbone itself. Albeit, a backbone link counts a single hop in this paper, the underlying real-world network is far more complex. The question of how to optimize the backbone network communication is highly important, but is not focus of this work.

V. SIMULATION OF MOBILE SMALL-WORLD NETWORKS

An example illustration of LSWTC with small-world backbone links is shown in Fig. 3.

Algorithm 1 Reckful Roaming RR_c (for node v)

```

1: INPUT:  $N$ : Initial set of neighbor nodes
2: INPUT:  $N2$ : Neighbors of  $u \in N$  (to calc  $CC$ )
3: INPUT:  $RC$ : Set of remove candidates
4: OUTPUT:  $N_{RR}$ : Resulting neighbor nodes
5:  $CC(v) = \frac{|E(\Gamma_v)|}{k_i(k_i-1)}$ 
6:  $RC(v) \leftarrow \emptyset$ 
7: for all  $u \in N$  do
8:    $N(v) \leftarrow N(v) \setminus \{u\}$ 
9:    $CC_{RC}(v) = \frac{|E(\Gamma_v)|}{k_i(k_i-1)}$ 
10:  if  $CC_{RC}v > CC(v)$  then
11:     $RC(v) \leftarrow RC(v) \cup \{u\}$ 
12:  end if
13:   $N(v) \leftarrow N(v) \cup \{u\}$ 
14: end for
15:  $sendRC(v, RC(v))$ 
16:  $sendRC(u, RC(u))$ 
17: for all  $w \in (RC(v) \cap RC(u))$  do
18:    $CC(w) \leftarrow \sum_{x \in (N(v) \cup N(u))} CC(x)$ 
19:    $N(v) \leftarrow N(v) \setminus \{w\}$ 
20:    $CC_{rc}(w) \leftarrow \sum_{x \in (N(v) \cup N(u))} CC(x)$ 
21:   if  $CC_{rc}v > CC(w)$  then
22:     if  $N(v) \cap N(u) == \emptyset$  then
23:       if  $permit(v) == true$  then
24:          $N(v) \leftarrow N(v) \cup \{w\}$ 
25:       end if
26:     end if
27:   end if
28: end for
29:  $N_{RR}(u) \leftarrow N(v)$ 

```

A. Properties

Connectivity: KHOPCA does not remove links from the device’s neighboring list, it only creates an overlay network for data forwarding decisions. On the other side, RR removes links from the communication topology. The final removal decision, however, relies on a local connectivity criterion, i.e. a link $\{u, v\}$ is only removed if there is a link in the 2-hops neighborhood of u and v . Therefore, LSWTC preserves connectivity of the initial network topology.

Synchronization: KHOPCA is only collecting weight information of the direct neighbors and deciding its own weight based on them, thus does not require any synchronization. The outcome of RR, however, is far more sensitive to asynchronization, since it deals with agreements of neighboring nodes. An aggravating factor is the fact that the information is collected from the 2-hops neighborhood, since 2-hops topological information is required in order to plan the action. This local link removal affects the 2-hops neighbors. For reasons of simplicity the presented model has been implemented in a synchronous network, the asynchronization procedure is not detailed here. However, Awerbuch [17] showed that any synchronous algorithm can be transformed in its asynchronous counterpart.

Algorithm 2 *KHOPCA* (for node v)

```
1: INPUT:  $N$ : Set of neighbor nodes  $\{n_1, \dots, n_{|N|}\}$ 
2: INPUT:  $W$ : Set of neighbors' weights  $\{w_1, \dots, w_{|W|}\}$ 
3: INPUT:  $ID$ : A function of a unique ID for each node
4: INPUT:  $k_K$ : Maximum weight
5: OUTPUT:  $w_v$ : New weight for node  $v$ 
6: if  $\min(W) < w_v$  (Rule 1) then
7:    $w_v \leftarrow \min(W) + 1$ 
8: else if  $\min(W) = k \wedge w_v = k_K$  (Rule 2) then
9:    $w_v \leftarrow 0$ 
10: else if  $\min(W) > w_v \wedge w_v \neq 0$  (Rule 3) then
11:    $w_v \leftarrow w_v + 1$ 
12: else if  $\min(W) = 0 \wedge w_v = 0$  (Rule 4) then
13:   if  $ID(N(\min(W))) > ID(v)$  then
14:      $w_v \leftarrow w_v + 1$ 
15:   end if
16: end if
```

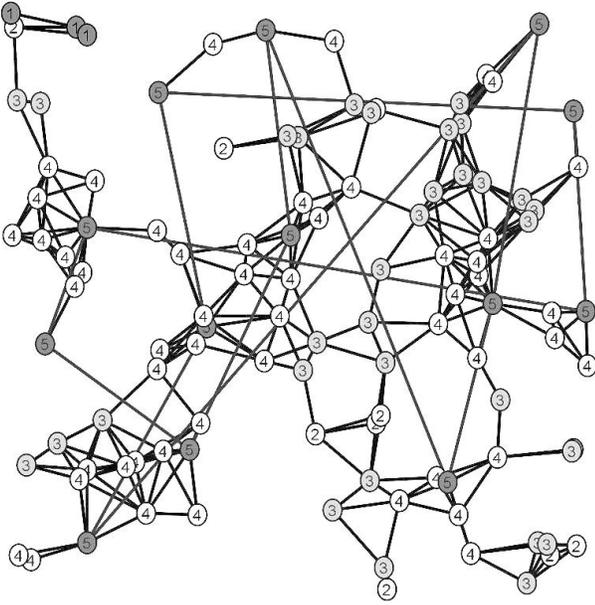


Figure 3: *LSWTC* with $k_K = 4$. Clusterheads have weight 5; nodes directly connected to a clusterhead have weight 4, etc. Shortcuts connect long-distant clusterheads.

Message complexity: The message complexity in a distributed system plays a crucial role, while time and space complexity are of less importance. It is defined as the number of messages sent in a system. In a system with wireless transmission, a message sent from node v to all or any of its 1-hop neighbors $Neigh(v)$ counts as 1. The message complexity of the *LSWTC* protocol consists of *RR* and *KHOPCA*. The message length is assumed to be less than $O(\log n)$ bits. This is sufficient to encode all the information exchanged.

KHOPCA does not require any predetermined initial configuration. A node can potentially choose weight between 0 and k as initial configuration. Thus, no message exchanges are required in this step. In the second step, each node collects solely

each neighbor's weight and compares it with its own, choosing the rule to apply. Weights are transmitted via beaconing. Thus, the message complexity of each appliance of *KHOPCA* results in $O(n)$ where n is number of nodes.

The optimization of *RR* relies on the clustering coefficient measure which is inherently demanding 2-hops neighboring information. The calculation of the 2-hops neighborhood requires a node v to send a beacon with its ID to its neighbors and in the second step to forward the neighboring list $Neigh(v)$ created by the received beacons to its neighbors to build the 2-hops neighboring list. The final step is the negotiation with nodes that are connected to the links that are removal candidates. The message complexity is $O(n)$. Thus, the message complexity of *LSWTC* is $O(n)$.

B. Simulation Results

The performance of *LSWTC* is evaluated by simulation. Simulations are conducted with different network settings. An r -disk graph describes the ad hoc communication graph whereby r is the transmission range and the deployment model D is a uniformly random distribution of the nodes in the bounded region with length l . Additionally, the number of nodes n equals to 130.

Since this primary simulation of *LSWTC* is concerned with topological properties where node affiliation characteristics do not have any role for the metrics used, changes of node speed does not affect the results. However, for the simulation results presented nodes are moving according to the random direction model. Node speed is set to an average pedestrian walking speed (1.5 m/s).

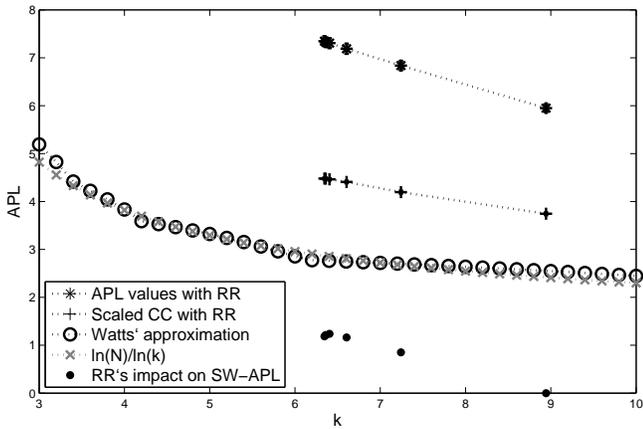
The lower limit for the network density is set to 6.11, since *RR* requires a connected initial topology. The network density is defined as $d = \frac{r^2 \pi n}{l^2}$ where n is the number of nodes and d corresponds to the average number of links per node. It is shown that 6 is a "magic" number where the network is with high probability still connected whereas for numbers lower than 6 the probability decreases drastically.

The upper limit for network density d is set to 9.95, covering in total an interval from 6 to 10. Additionally, for all simulations a confidence interval of 95% has been applied.

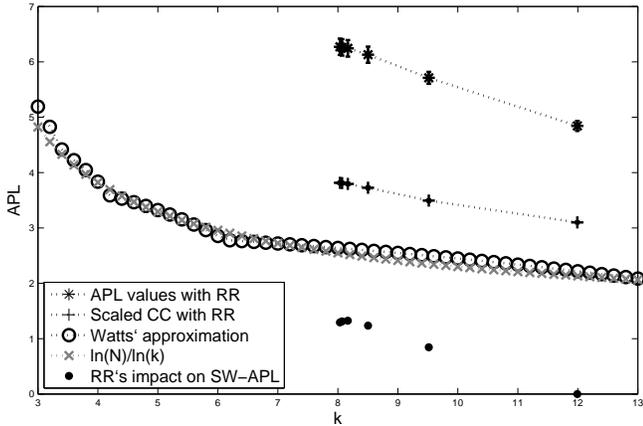
Effect of clustering coefficient optimization on average path length: *RR* aims to remove dedicated links in order to increase the clustering coefficient. Since, a removal of links on any network type increases the path length between the nodes, *RR* increases the average path length.

In Fig. 4 this effect of *RR* on the topology is shown. The Watt's approximation for the small-world average path length as well as the simplified approximation $\frac{\ln n}{\ln k}$ is given. Furthermore, the optimization realized by *RR* on the clustering coefficient as well as the increase of *APL* is shown. The real impact on the small-world property by the increase of the *APL* is illustrated as well. This impact is calculated as the difference between the *APL* small-world limit and the actual *APL* value.

As discussed above, the small-world *APL* depends on the average node degree k as described by the approximation $\frac{\ln n}{\ln k}$.



(a) $d = 6.11$



(b) $d = 9.96$

Figure 4: The effect of increasing the CC by removing links on the APL .

RR removes links from the topology, so that k decreases. That is why the small-world behavior of the initial topology and the RR -topology cannot be compared directly, rather the properties have to be scaled. Fig. 4 shows mainly two important points: (1) the decrease of the average node degree after applying RR and (2) the increase of the APL after applying RR relative to the small-world APL . Results show that by decreasing k the increase of APL is considerably low due to the natural increase of the small-world average path length.

In particular for $d = 6.11$, this result indicates that removing links affects more CC than APL and paves the way for applying dedicated shortcuts to compensate for deterioration of APL . However, for increasing values of d , this imbalance between the impact of removing links between CC and APL seems to balance (Fig. 4b), increasing the distance to the small-world average path length.

Mobile small-world wireless networks: The inter-connection on the backbone—not issue of this work—is simulated by a small-world network that can easily be created since global knowledge of all access points is available. These shortcuts have been included into the initial topology. To put in question,

is the impact of the shortcuts on locally maintained topology enough in order to evoke small-world properties?

Results for $d = 6.11$ and for $d = 9.96$ are shown in Fig. 5 and 6 respectively. It has been pointed out, that a small-world graph is a graph with n vertices and an average degree k that exhibits an average path length $APL \approx APL_{Random}(n, k)$ and a clustering coefficient $CC \gg CC_{Random}(n, k) \approx \frac{n}{k}$.

In both figures, the CC and APL of a random network with n nodes and an average node degree k are compared with the CC and APL before (CC_{old} and APL_{old}) and after (CC_{new} and APL_{new}) applying $LSWTC$ with $k_K = \{2, \dots, 7\}$.

In all cases, the CC increases up to 20% whereby it increases with higher k_K . This significantly fulfills the small-world property for the clustering coefficient.

The average path length is reduced about 35% to 50% by applying the shortcuts on the topology.

For $d = 6.11$ it is observed that the APL gets close to the small-world average path length while for $d = 9.96$ the APL is not in accordance of the small-world criteria. According to [1] the approximation observed in Fig. 5 can be classified as fulfilling the small-world property.

The enormous decrease of the APL by the shortcuts can be explained by the fact that Reckful Roaming generates a more sensitive network in a way that well-chosen interlinks recover and improve the APL values. Simulations show that in contrast to the initial topology, the resulting topology requires significantly less interlinks to reach the same small-world performance. This also explains why networks with a less network density are able to evoke small-world properties while dense networks are less sensitive to shortcut-placement to decrease the average path length and their high average node degree impede the evocation of small-world properties.

VI. CONCLUSIONS

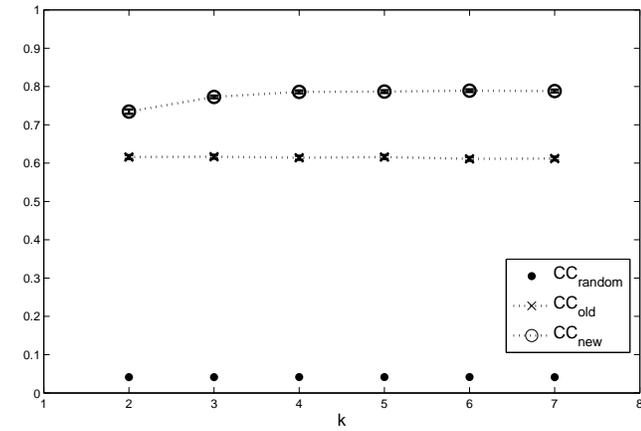
Small-world networks have a high clustering coefficient while still retaining a small average path length. These small-world properties induce several benefits. Whereas most approaches focus on augmenting the network by well-chosen shortcuts in order to evoke small-world properties, the introduced $LSWTC$ that relies on topology control and clustering techniques focuses on creating highly clustered regions by removing inefficient links.

The results show that small-world properties can be evoked and maintained in such highly dynamic networks as backbone-assisted mobile ad hoc and sensor networks. It has been found that a low network density is the fundamental parameter in order to fulfill the small-world criterion.

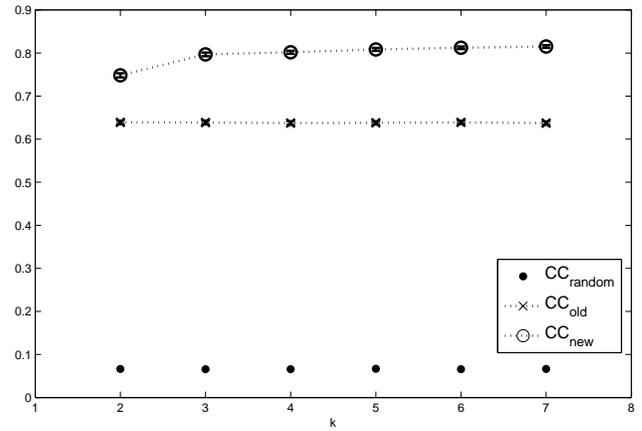
The $LSWTC$ algorithm is shown to be extremely adaptive to mobility and works locally and fully distributed.

ACKNOWLEDGMENT

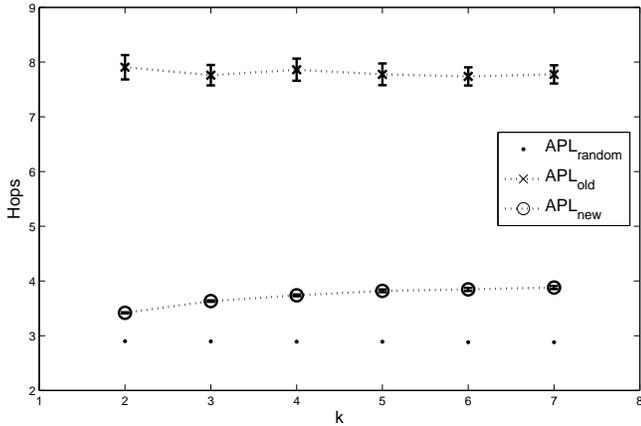
Matthias R. Brust is grateful to FAPESP (Grant no. 2008/52129-9). Carlos H.C. Ribeiro is grateful to CNPq (Grant no. 301228/97-3-NV) and FAPESP (Grant no. 2010/11334-9).



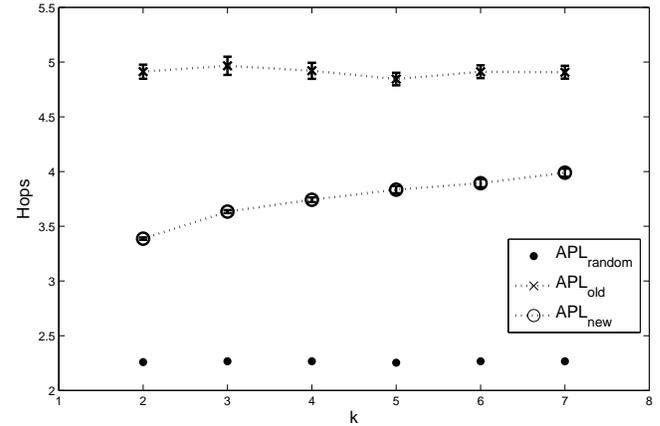
(a) CC



(a) CC



(b) APL



(b) APL

Figure 5: Analyzing small-world properties backbone-assisted mobile ad hoc networks for network density $d = 6.11$.

Figure 6: Analyzing small-world properties backbone-assisted mobile ad hoc networks for network density $d = 9.96$.

REFERENCES

- [1] D. J. Watts, *Small Worlds - The Dynamics of Networks between Order and Randomness*. Princeton University Press, 1999.
- [2] J. Kleinberg, "The small-world phenomenon," in *Proceedings of the Thirty-Second Annual ACM symposium on Theory of Computing (STOC '00)*, pp. 163–170, 2000.
- [3] M. Newman and D. Watts, "Scaling and percolation in the small-world network model," *Physical Review E*, vol. 60, pp. 7332–7342, Dec. 1999.
- [4] H. Youn, M. Gastner, and H. Jeong, "Price of Anarchy in Transportation Networks: Efficiency and Optimality Control," *Physical Review Letters*, vol. 101, pp. 1–4, Sept. 2008.
- [5] H. Frey and F. Ingelrest, *MFACE: A Multicast Backbone-Assisted Face Traversal Algorithm for Arbitrary Planar Ad Hoc and Sensor Network Topologies*, pp. 1–20, 2008.
- [6] I. F. Akyildiz and I. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Ad Hoc Networks*, vol. 2, pp. 351–367, Oct. 2004.
- [7] M. R. Brust and S. Rothkugel, "Small-Worlds: Strong Clustering in Wireless Networks," in *Proceedings of the First International Workshop on Localized Algorithms and Protocols for Wireless Sensor Networks (LOCALGOS 2007)*, pp. 78–86, 2007.
- [8] M. R. Brust, H. Frey, and S. Rothkugel, "Dynamic Multi-Hop Clustering for Mobile Hybrid Wireless Networks," in *Proceedings of the Second International Conference on Ubiquitous Information Management and Communication (ACM ICUIMC 2008)*, pp. 130–135, 2008.
- [9] J. Hou and L. Sha, "Design and analysis of an MST-based topology control algorithm," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 1195–1206, May 2005.
- [10] L. E. Li and P. Sinha, "Throughput and energy efficiency in topology-controlled multi-hop wireless sensor networks," in *Proceedings of the Second ACM International Conference on Wireless Sensor Networks and Applications*, pp. 132–140, 2003.
- [11] M. Bahramgiri, M. Hajiaghayi, and V. S. Mirrokni, "Fault-Tolerant and 3-Dimensional Distributed Topology Control Algorithms in Wireless Multi-hop Networks," *Wireless Networks*, vol. 12, pp. 179–188, Dec. 2005.
- [12] S. Basagni, D. Turgut, and S. K. Das, "Mobility-adaptive protocols for managing large ad hoc networks," in *Proceedings of IEEE International Conference on Communications (ICC 2001)*, pp. 1539–1543, 2001.
- [13] F. Dai and J. Wu, "On constructing k-connected k-dominating set in wireless ad hoc and sensor networks," *Journal of Parallel and Distributed Computing*, vol. 66, no. 7, pp. 947–958, 2006.
- [14] M. R. Brust, A. Andronache, and S. Rothkugel, "WACA: A Hierarchical Weighted Clustering Algorithm optimized for Mobile Hybrid Networks," in *Proceeding of the Third International Conference on Wireless and Mobile Communications (ICWMC 2007)*, pp. 23–30, 2007.
- [15] F. Nocetti, J. Gonzalez, and I. Stojmenovic, "Connectivity based k-hop clustering in wireless networks," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, vol. 22, pp. 2450–2459, Jan. 2003.
- [16] Y. Fernandess and D. Malkhi, "K-clustering in wireless ad hoc networks," in *Proceedings of the ACM Workshop On Principles Of Mobile Computing*, pp. 31–37, 2002.
- [17] B. Awerbuch, "Complexity of Network Synchronization," *Journal of the ACM (JACM)*, vol. 32, no. 4, pp. 804–823, 1985.