Topology Control for Fault-Tolerant Communication in Highly Dynamic Wireless Networks*

Bernd Thallner, Heinrich Moser

[1]Embedded Computing Systems Group E182/2
Technische Universität Wien, Treitlstraße 3, A-1040 Vienna, Austria
{thallner,moser}@ecs.tuwien.ac.at

Abstract — *Energy efficiency and fault-tolerance are the most important issues in the development of next-generation wireless ad hoc networks and sensor networks.* Topology control *as a low level service (typically below the traditional layer structure) governs communication among all nodes and is hence the primary target for increasing connectivity and saving energy. In this paper, we present an improvement of our topology control algorithm for very dynamic networks and low power devices (e.g. sensor nodes). The algorithm constructs a fault-tolerant topology for energy-efficient and fault-tolerant multi-hop communication in a two-tier network consisting of a large number of wireless nodes and a few gateway nodes (e.g. base stations responsible for exchanging data with other networks). Using only local information, like distance/channel attenuation to neighbors, our fully distributed algorithm efficiently constructs and continuously maintains a k-regular overlay graph that guarantees low total transmission power, is k-node-connected and ensures failure locality. It automatically adapts to a dynamically changing environment, is guaranteed to converge, builds a hierarchy of clusters that reflects the node density and exhibits good performance as well.*

1 Introduction

During the recent years, wireless multi-hop networks (wireless sensor networks, mobile networks, ad hoc networks, ...) experienced a steady increase of both the number of applications and system size. The miniaturization of network nodes, which are primarily battery-powered nowadays, raised an increasing concern for power consumption and hence power efficiency. In addition, due to evolving critical application domains and the increasing number of failures that are likely to occur in systems of that size, there is also a growing demand for security and fault-tolerance.

*This research is supported by the FWF-project *Theta* (project no. 17757-N04)

A particularly critical component with respect to energy-efficiency and fault-tolerance in wireless networks is topology control. By selecting the particular neighbors a node may directly communicate with, topology control maintains a (sparse) *overlay graph* that can be used for multi-hop communication between any two nodes in the network. From a practical point of view, the number of neighbors per node should be bounded by a (small) constant here: Most wireless networks require some dedicated communication hardware for every link. For example, CDMA needs a correlator on both sender and receiver side of every point-to-point connection. The same is true if peer-to-peer hardware encryption/decryption is used on every link. As usual chipsets provide a fixed number of such devices, the overlay graph should ideally be regular (which could also improve resource and power utilization).

The problem of constructing/maintaining such an overlay graph is further exacerbated by the fact that link performance can degrade, that both links and nodes can crash and recover, and that nodes can move. In order to guarantee unimpaired communication under such circumstances, the overlay network must be robust: It should provide fault-tolerance and adapt quickly to changes in the environment. Suitable overlay graphs should hence be maintained in a cost-optimal and self-healing way: If connections or nodes become expensive or go down for some time, the overlay graph must be adapted in order not to use them further.

Still, traditional topology control solutions (see Section 2) cannot cope with those requirements, since fault-tolerance is usually sacrificed for power efficiency. In order to be power efficient, topology control algorithms try to reduce the number of links and thereby reduce the redundancy available for tolerating node and link failures.

The topology construction approach utilized in this paper avoids this problem, by means of a suitable separation of concerns: Some specified fault-tolerance, namely, a k-(vertex/ 0node)-connected network, is guaranteed by choosing a suitable (but provably minimal) number of links to be added to the overlay graph. Power efficiency is introduced in the construction algorithm by selecting the most efficient links among the set of available ones. Every (potential) link in the network has associated an arbitrary *weight* for this purpose, that is, we assume that every node can probe or estimate how expensive or difficult it is to communicate with a specific peer. Distance, required transmission power, interference level or any combination of such quantities are legitimate weights here. Note that the assigned weights need <u>not</u> satisfy the triangle inequality and that we do <u>not</u> require homogeneous nodes or uniform transmission ranges. Hence our algorithm does not require position information or assume the communication graph to be a Unit Disk Graph. Rather we use a general weighted communication graph. In addition, weights may be time-dependent: For example, a communication peer could be a moving node or the required transmit power might not only depend upon the distance but also upon the instantaneous level of the internal (thermal) noise, multiuser interference, signal attenuation, multi-path fading, and many more. Likewise, a link to a receiver that temporarily suffers from low battery or heavy processor load may be considered more costly than usual.

In the remainder of this paper, we present a fully distributed algorithm for constructing and continuously maintaining a k-regular and k-connected overlay graph for fault-tolerant multi-hop communication in large-scale wireless networks, which is based upon a clustering scheme introduced in [1][1]. It recursively forms groups consisting of k nodes that are treated like single nodes subsequently.

According to the above separation of concerns, the algorithm actually consists of two reasonably independent parts (that also allow to use our scheme in very different wireless networks):

1. The generic construction algorithm (presented in Section 5), which builds up and continuously maintains the k-regular and k-connected overlay graph. It does so by processing proposals for links to be added to the overlay graph supplied by the specific propose module (see next item). Furthermore Section 6 presents an extension which keeps topology changes caused e.g. by a node failure in the vicinity of the failed node. This novel extension is particularly beneficial in very dynamic environments and low power devices (e.g. sensor networks).
2. The propose module (cp. Definition 4), which tries to find minimal-weight links to be added to the overlay graph. The propose module is network-specific and allows to trade construction complexity for minimality of the weight-sum of the overlay graph (and hence overall power efficiency, for example).

The overlay graph is k-connected, which is optimal, and thus ensures that k node-disjoint paths exist between any pair of nodes. It has low total weight and inherently provides failure-locality as well: Even excessively many failures in some part of the system do not impair fault-tolerant communication in other parts. As a by-product, the algorithm produces a hierarchy of clusters represented by a k-ary tree that reflects the node "density". This property can be used in higher level services, like data aggregation in sensor networks, routing, naming, as well as geo- and multicasting.

Organization of our paper: After a short survey of related work in Section 2 and some definitions in Section 3, we introduce our basic method for constructing a fault-tolerant communication topology in Section 4. Section 5 presents our fault-tolerant algorithm, which implements this method in a fully distributed way. Section 6 introduces our novel extension for very dynamic networks and for low power devices. Some conclusions and directions of future research are provided in Section 7.

2 Related Work

Several non fault-tolerant topology control algorithms have been proposed in literature (refer to [2] for an overview). Most of them rely on the homogeneous network assumption with equal transmission range which may not hold in practice [2].

A few fault-tolerant topology control algorithms [3], [4] and [5] have also been presented in recent years (see [3] and [5] for a summary): Hajiaghayi [3] presented approximation algorithms for minimum weight k-connected subgraphs based on the minimum spanning tree. However, [5] contains a counter-example which shows that the topology does not assure k-connectivity. In [4], Bahramgiri et al. extended their CBTC algorithm to construct a fault-tolerant topology: It is proved to be k-connected but requires a homogeneous network. Li and Hou [5] first presented a fault-tolerant extension for their greedy

[1]Note that [1] introduced the basic idea of our clustering scheme. This paper is devoted to an extension of this scheme, which poses a number of unique problems.

algorithm of [2], and then derived a localized algorithm. Still, none of those solutions ensures fault-tolerance with bounded or fixed node degree and therefore a low or minimum number of connections.

3 Definitions

We consider a simple undirected weighted graph $G = (\Pi, \Lambda)$ consisting of a set of n *nodes* $\Pi = \{1, ..., n\}$ and a set of weighted *edges* $\Lambda \subset \Pi \times \Pi \times \mathbb{R}$. The network is modeled as a *communication graph* G and contains the set of potential edges. It is assumed to be fully connected, in the sense that $w < \infty$ for any edge $(x, y, w) \in \Lambda$. We will drop the fully connected graph assumption, however, i.e., allow $\omega = \infty$, when we introduce the extension of our construction method in Section 6. Note carefully that this assumption does not mean that any node actually communicates with every other node, but only that it could communicate with every (reasonable) peer. Both the set of alive nodes and the weight of the edges in the communication graph G may be time-variant.

Our algorithm constructs a low weight *overlay graph* G' that is k-regular and k-connected, for some given k. Recall that a graph is κ*-connected* (also referred to as κ-node-connected or κ-vertex-connected) if the removal of any subset of $\kappa - 1$ nodes leaves the graph connected while there exists a subset of κ nodes whose removal disconnects the graph. A graph is *regular* of degree r if all nodes have degree r. In order to easily distinguish the communication graph G and the overlay graph G', the edges of the latter will be called *connections*.

In order to avoid the special top-level group of the overlay graph, employed in [1], we introduce *gateway nodes*[2] and assume that a small number of them ($n'' \geq 2k - 2$ are sufficient, cf. Theorem 1) are present in the network. In addition to the wireless communication links to/from regular nodes, which have to be set up by our algorithm, all gateway nodes are assumed to be fully interconnected with all other gateway nodes via a dedicated backbone network. The set of nodes Π hence consists of n' *regular nodes* Π' and n'' *gateway nodes* Π'' with $n = n' + n''$ and $\Pi = \Pi' \cup \Pi''$. In order to ensure that gateway nodes are only used after all regular nodes have been exhausted, it suffices to assume that the edge weight between regular and gateway resp. gateway and gateway nodes are chosen according to $\forall x \in \Pi', y \in \Pi'' \Rightarrow (x, y, k^2K) \in \Lambda$ resp. $\forall x \in \Pi'', y \in \Pi'' \Rightarrow (x, y, 2k^2K) \in \Lambda$, where K is the maximum edge weight between regular nodes. Our algorithm will then construct a low weight overlay graph $G' = (\Pi' \cup B, C)$ with $B \subseteq \Pi''$ and $C \subseteq \Lambda$. Note that possibly not all gateway nodes are used in the overlay graph and that gateway nodes in G' may have degree $k - 1$ due to the additional backbone interconnection, whereas regular nodes are always used up and always have degree k.

4 Topology Construction Method

In this section, we provide an overview[3] of the clustering scheme introduced in [1], which induces an overlay graph G' with the desired properties. The idea is to build groups of nodes that appear like single nodes, such that they can be treated like those subsequently.

[2] For a single tier topology without gateway nodes refer to [1].

[3] The results provided in this section actually differ from [1] in that we changed the handling of root groups.

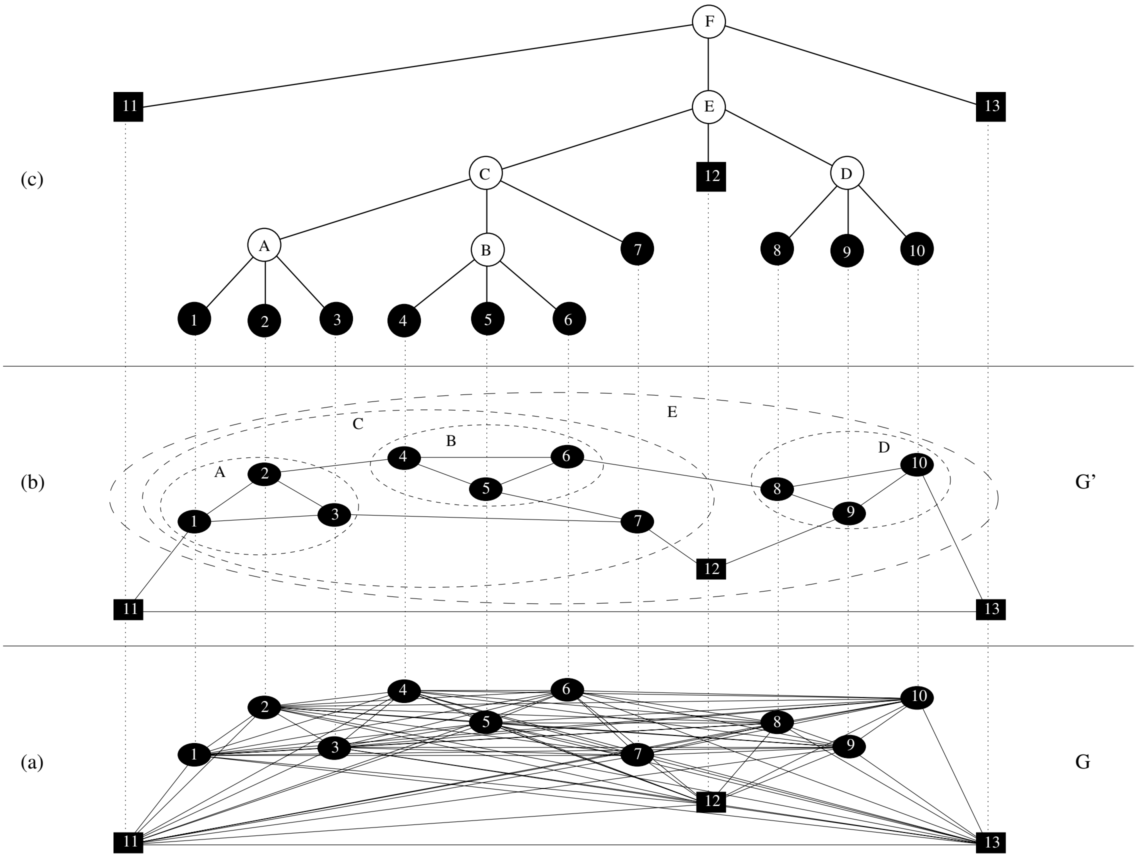

Figure 1: Communication Graph (a), Network Graph (b) and Topology (c)

Figure 1 shows an example, where 1a is the fully connected communication graph G, 1b depicts the constructed overlay graph G' for $k = 3$, and 1c provides the tree representation corresponding to the constructed topology. From 1b it is apparent that the regular nodes $(1, 2, 3)$, $(4, 5, 6)$ and $(8, 9, 10)$ are combined into groups with id A, B, and D, respectively. Such a group is formed if all members agree upon the fact that the sum of the weights of their internal connections (e.g. $4 - 5$, $4 - 6$, $5 - 6$) is minimal over all alternative group constructions. Each of the k members of a group is connected to all of the $k - 1$ other members (internal connections) and has exactly one connection left (external connection). Since there are k members in a group, any group has k external connections left, which are available in higher level groups. From the point of view of higher-level groups, groups hence look like nodes.

For example, group C again consists of three members: A single node 7 and two groups A and B, which are connected via their external connections. Again, all members of C agree upon minimality of the sum of their internal connections' weights. Groups E and F finish the topology and include gateway nodes ($11, 12$ and 13), which may have degree $k - 1$ due to the additional backbone connectivity. Figure 1c reveals that the resulting group structure is a k-ary tree. The edges of the tree represent the membership relation among the groups.

We now give a formal description of our topology: A *group* consists of an identifier g_i and a set of $members(g_i)$. The set of all group identifiers is $\mathbb{G}$. The set of $members(g_i)$ consists of exactly k nodes and groups: $members(g_i) \subseteq (\mathbb{G} \cup \Pi)$, $|members(g_i)| = k$. A node or group can only be member of a single group:[4] $\forall g_a, g_b \in \mathbb{G}, g_a \neq g_b : x \in members(g_a) \wedge y \in members(g_b) \Rightarrow x \neq y$. For every $g_i \in \mathbb{G}$ we define the nodes of a group as $nodes(g_i) = \bigcup_{l=0}^{\infty} members^l(g_i) \cap \Pi$ where $members^0(g_i) = members(g_i)$ and $members^j(g_i) = \bigcup_{r \in members^{j-1}(g_i) \wedge r \in \mathbb{G}} members(r)$. For every node $p \in \Pi$ we define $nodes(p) = \{p\}$.

A connection $(p, q, w) \in C$ from node p to node q with weight w is a *group g_i internal connection* if $p \in nodes(g_a)$ and $q \in nodes(g_b)$ with $g_a, g_b \in members(g_i)$ and $g_a \neq g_b$. If there is a connection $(p, q, w) \in C$ between node $p \in nodes(g_a)$ and node $q \in nodes(g_b)$ we call the groups g_a, g_b connected. The members of a group are fully connected among them: $\forall g_a, g_b \in members(g_i), g_a \neq g_b \Rightarrow \exists p \in nodes(g_a), q \in nodes(g_b), (p, q, w) \in C$.

Hence, every group member has $k-1$ connections to other group members and therefore exactly one connection left. Since there are k members in a group, the group has—like a node—k connections left. We call the k nodes of a group g_i with one connection left the *terminal nodes* $T_{g_i} \subseteq \Pi$ of a group. By convention, we define that $T_p = p$ for a single node $p \in \Pi$.

Definition 1. *The* weight of a group $\omega(g_i)$ *is a triple* $(A_i, members(g_i)$,*internal connections of group* g_i*), where* A_i *is the maximum of the sum of all group* g_i *internal connection weights and the maximum of all group members' weights plus an arbitrary small constant* ϵ*, formally:* $A_i = max(\sum$*internal connection weights*$, max($*members' group weights*$) + \epsilon)$*. A group* g_i *has smaller weight than* g_j*, formally* $\omega(g_i) < \omega(g_j)$*, if* A_i *is smaller than* A_j *or, if equal,* $members(g_i) < members(g_j)$ *in lexical order or, if equal, the internal connections of group* g_i *have less weight than the internal connections of group* g_j *in lexical order.*

Note that this definition implies that the weight of a parent group is always higher than the weight of any of its members. This property is required for ensuring that the minimum admissible overlay graph introduced in Definition 3 is well defined, and that our distributed algorithm converges.

Definition 2. *An overlay graph* G' *is called* admissible *if its corresponding topology has a single* root group g_{root} *where all terminal nodes are gateway nodes:* $T_{g_{root}} \subseteq \Pi''$.

Recall that we assumed that all gateway nodes are fully connected among themselves (via some backbone). The following Theorem 1 shows that no more than $2k-2$ gateway nodes are necessary to construct an admissible overlay graph.

Theorem 1. *For every graph* G *with* $n' \geq 1$ *and* $n'' \geq 2k-2$*, there exists an admissible overlay graph* G'.

Proof. Consider the topology corresponding to an overlay graph G', where all regular nodes are used up for constructing a possibly incomplete subgroup X, like group E in

[4] Note that the member function is not transitive.

Figure 1b. As it is incomplete, up to $k-1$ gateway nodes are required to complete X. A root group g_{root} where all terminal nodes are gateway nodes, i.e., no regular node with an external connection is left, can be constructed by forming a group g_{root} consisting of X and $k-1$ additional gateway nodes. This construction hence requires at most $2k-2$ gateway nodes. □

By introducing a suitable minimum criterion, we can even stipulate the existence of a unique *minimal* admissible overlay graph. Note carefully that this is not necessarily the global minimum-weight overlay graph, but rather the minimal one w.r.t. all alternative admissible topology constructions subject to the particular minimum criterion. Informally, the minimum criterion defined below requires that, for any group member x of a minimal group, the weight-sum of all internal connections is minimal over all alternative minimal group constructions involving x. Every member of a minimal group must hence arrive at the conclusion that there is no better choice, i.e., they must agree on minimality. Still, the range of alternative choices is restricted by this requirement: Although it could of course be the case that a lower-weight alternative group existed for some member, this choice would lead to a violation of the minimum requirement for some other group and must hence abandoned.

Definition 3. *An admissible overlay graph G' is* minimal *if, for every member $x \in members(g)$ of any group $g \in \mathbb{G}$ in the corresponding topology tree, no alternative group g' can be built with $x \in members(g')$, $\omega(g') < \omega(g)$ and $\forall y \in members(g') : \omega(g') < \omega(g_y)$, where g_y is the (unique) group in the topology tree with $y \in members(g_y)$.*

The minimum admissible graph is well-defined: Choosing a new group according to this criterion does not lead to a violation of the minimality of any already existing group in the final topology, since Definition 1 ensures that a higher-level group has a higher weight than any of its members. Similarly, Theorem 1 holds also for the minimum admissible overlay graph, since the edge weights for gateway nodes have been chosen in a way that guarantees that regular nodes are always preferred by the minimum criterion. The following Theorem 2 proves that the minimum admissible overlay graph indeed exists and is unique.

Theorem 2. *For every graph G with $n' \geq 1$ and $n'' \geq 2k-2$, there is exactly one minimal admissible overlay graph G'.*

Proof. We first show that at least one minimal admissible overlay graph exists, by inductively constructing minimal groups one after the other and showing that those groups are stable, i.e., not destroyed during later construction steps. For $i \geq 1$, let g_i be the group added in the i-th step of this construction and G_{i-1} be the set of groups constructed in steps $1, \ldots, i-1$, with $G_0 = \Pi$ denoting the set of all nodes (by convention, we assume that $\omega(x) = 0$ for a single-node "group" $x \in \Pi$).

For $i = 1$, g_1 is the (unique) group with minimal $\omega(g_1)$ chosen among all "groups" (nodes) in G_0. Clearly, g_1 is minimal and trivially stable. For g_i, $i > 1$, we choose the group with minimal weight $\omega(g_i)$ formed from groups in G_{i-1}. Clearly, all members of g_i agree upon this as the minimal choice. It remains to be shown, however, that the choice

```
if a new proposal is provided by a propose module
   if ( all particpants agree that the proposal is better than their current parent group )
      all participants join the proposed new group [ atomically ]

periodically
   check for group consistency
   if group is consistent
      recalculate group weight
   else
      all participants leave the group
```

Figure 2: Construction Algorithm

to include some member $x \in G_{i-1}$ in g_i does not violate the minimum criterion for some earlier built group $g \in G_{i-1}$, which might already have x as a member.

So assume that it is the case that x is a member of both g_i and g with $\omega(g_i) < \omega(g)$, and let $j < i$ be the step where g has been built. Our definition of the group weight $\omega(.)$ ensures that any higher-level group must have a weight higher than the weight of every member. However, since g_i is also built, in steps $j, \ldots, i$, out of some groups in G_{j-1}, $\omega(g_i) < \omega(g)$ implies that g cannot have been the minimal agreed choice for x in step j. This provides the required contradiction.

Finally, since Theorem 1 holds also for this inductive construction, it is ensured that all regular nodes are used up before gateway nodes are considered. Hence, the root group can be built since at least $2k-2$ gateway nodes are available.

It still remains to be shown that the minimal overlay graph is unique. So let us assume that there exist two admissible overlay graphs G'_1 and G'_2, which are both minimal. Going up the topology tree of G'_1 and G'_2, at some depth the group structure must be different. More specifically, there must be a group member x in some group g_i in G'_1 which is member in some other (= not corresponding) group g_j in G'_2. Recall that a node or a group can only be member of at most one group. However, either g_i of G'_1 or g_j of G'_2 has lower weight, which implies that the admissible overlay graphs G'_1 and G'_2 cannot both be minimal according to Definition 3. □

The following theorems establish some important properties of our overlay graphs.

Theorem 3. *In every admissible overlay graph G' the node degree is bounded by k and all regular nodes have degree k.*

Proof. Obvious from the topology construction and Definition 2. □

Theorem 4. *Each admissible graph G' with $n'' \geq 2k-2$ is k-connected.*

See [6] for the Proof.

Our results reveal several interesting features and advantages of our approach. First of all, connection weights may be arbitrary; in particular, they need not to satisfy the triangle inequality. Moreover, by adding additional constraints to Definition 3, overlay graphs with specific additional properties can be built.

If the weights in the communication graph reflect physical distance, our minimal topology construction clusters nodes according to their spatial density.

5 The Distributed Construction Algorithm

In this section, we briefly present[5] a fully distributed algorithm that builds up and continuously maintains the minimal admissible overlay graph in dynamic environments.

For our system model, we assume a fully connected asynchronous system with reliable links, where nodes may crash and connection weights are possibly time-variant. Late joining of nodes is allowed, but no (undetected) crash and recovery. Our algorithm requires a weak non-blocking atomic commitment service like the one of [8], which is invoked by the candidate members of a to-be-formed group.

In our algorithm, every node and existing group (that is, their terminal nodes) concurrently searches for the minimal-weight next-level group to join. For this purpose every node repeatedly generates proposals P, consisting of the group members, the group weight, the group internal connections and the group's terminal nodes, which are sent to (the terminal nodes of) the proposed group members for confirming minimality. Generating a new proposal is typically triggered periodically, to facilitate adaption to changed connection weights, or upon detection of a node crash or join. Figure 2 give a high level pseudo-code description of our algorithm.

The structure of our algorithm allows to encapsulate the functionality of generating proposals in a dedicated *propose module*. Its most general specification is given by Definition 4.

Definition 4. *A* propose module *generates proposals for groups consisting of* k *members,* k *terminal nodes, group internal connections and the corresponding group weight. A propose module is* perfect *if it eventually generates proposals corresponding to groups in the final (unique) minimal overlay graph* G'.

Our analysis [6] reveals that the algorithm of Figure 2 guarantees convergence (to the unique minimal overlay graph) for every reasonable propose module. Speed of convergence as well as message and time complexity strongly depend on the propose module actually used. In fact, different propose modules presented in [9] allow to trade message complexity for convergence speed and minimality: The worst case total message complexity for constructing an admissible overlay graph with n nodes ranges from $O(n)$ to $O(n^{k+1})$, the worst case time complexity ranges between $O(n)$ and $O(n^2)$. Our simulation experiments (see [9] for details) reveal good average case complexity and small stretch factors for various propose modules.

6 Extension

A drawback of the topology construction method of Section 4 is that it is not particularly efficient for very dynamic environments. A node that joins or leaves the network could trigger a complete restructuring of the topology. Consider the case where a node $p \in T_{g_i}$ with $g_i \in members(g_{root})$ leaves the network, for example: All groups g with $p \in T_g$ have to be rebuilt on that occasion. In this section, we introduce an extension of our method

[5] See [7] for a detailed explanation of the algorithm and its complete correctness proof.

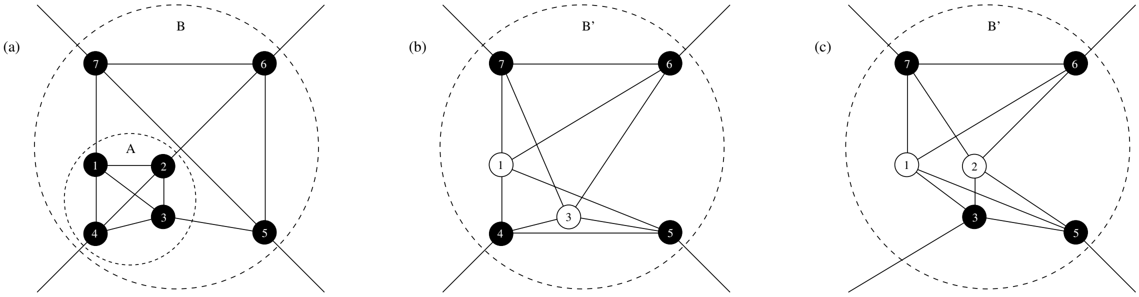

Figure 3: Regular and extended group structure

that allows nodes to join and leave a group without reconstructing the entire topology, by keeping changes local. The extended topology construction algorithm also induces a low weight—although not the minimal—overlay graph $G^* = (\Pi' \cup B, C)$ with $B \subseteq \Pi''$ and $C \subseteq \Lambda$ that is k-regular for regular nodes and k-connected if $n'' \geq 2k-2$ and k is even, or $k-1$-connected if k is odd.

Figure 3a shows an example of a group and its parent group with $k = 4$. Figure 3b and 3c depict the restructured group after successively removing nodes 2 and 4. The groups A and B are merged together to form a new group B' which has more than k members. The topology outside B and inside the members of A and B (except A) is not changed. Note that the number of terminal nodes of B' does not change, and that nodes 1 and 3 in 3b as well as 1 and 2 in 3c have no external connections.

We now give a formal description of our extended topology: *Extended groups* consist of an identifier $g_i^* \in \mathbb{G}$ and the set of $members(g_i^*) \subseteq (\mathbb{G} \cup \Pi)$ consisting of $k+1 \leq members(g_i^*) \leq 2k-2$ members. The set of terminal nodes of an extended group g_i^* consists of exactly k nodes. Since $|T_{g_i^*}| = k$, there is hence no difference between an extended and a regular group for an external node or group. The set of members $g \in members(g_i^*)$ of group g_i^* consists of terminal members with $\exists p \in nodes(g)$ and $p \in T_{g_i^*}$ and internal members with $\nexists p \in nodes(g)$ and $p \in T_{g_i^*}$. We denote the number of internal members for an extended group g_i^* by $I_{g_i^*} = |members(g_i^*)| - k \quad (1 \leq I_{g_i^*} \leq k-2)$. Note that a regular group g_i can be seen as an extended group with $I_{g_i} = 0$.

An extended group g_i^* is constructed as follows:

1. Each internal member of g_i^* has a connection to each terminal member. Since there are k terminal members, each internal member has k connections.
2. Each terminal member of g_i^* has one external connection, $I_{g_i^*}$ connections to internal members and $k-1-I_{g_i^*}$ connections to other terminal members if k is even. If k is odd, at most one terminal member has $k-2-I_{g_i^*}$ connections instead of $k-1-I_{g_i^*}$ to the other terminal members.

The group weight of an extended group $\omega(g_i^*)$ is defined analogously to Definition 1, except that the sum of the g_i^* group internal connections is normalized to the number of connections of a regular group: $\frac{k \cdot \text{sum of int. con.}}{k+I_{g_i^*}}$.

We have the following major result:

Theorem 5. *The graph G' with $n'' \geq 2k - 2$ and extended groups is k-connected if k is even.*

See Theorem 8 in [6] for the Proof.

Accommodating extended groups in the distributed algorithm of Section 5 requires only a few adaptations related to node joins and leaves:

- If a new node appears, it is integrated into some nearby group $g_i \neq g_{root}$ as internal member.
- If a node $p \in \Pi$ with $p \in members(g_i)$, $g_i \in members(g_j)$ and $g_j \neq g_{root}$ leaves the group, an extended group g_j^* is built with $members(g_j^*) = (members(g_j) \setminus \{g_i\}) \cup (members(g_i) \setminus \{p\})$. The group g_i is removed. If $p \notin T_{g_j}$ then $T_{g_j^*} = T_{g_j}$; if $p \in T_{g_j}$ then $T_{g_j^*} = (T_{g_j} \cup \{q\}) \setminus \{p\}$ with $q \in (T_{g_i} \setminus \{p\})$. In the latter case, a higher level group g_a where $p \notin T_{g_a}$ but $p \in T_{g_b}$ with $g_b \in members(g_a)$ has to build a new connection to the new terminal node q, cf. Figure 3c.
- If extended groups, regular groups and nodes are merged and the number of members of the new extended group would become $|members(g_j^*)| \geq k + k - 1$, a new regular group with k members and minimal weight is built and integrated as a single member into the new (extended) group. Note that restructuring proceeds in the opposite direction, from the root to the leaves of the tree, in this case.

Introducing extended groups provides a number of additional benefits. For example, a node that is k or $k - 1$-node-connected to the network preserves this property even when topology reconstruction is in progress. Most importantly, extended groups permit us to weaken the fully connected communication graph assumption: Non-existing edges $(x, y, \infty) \in \Lambda$ were disallowed in Section 3, since x and y must not become members of a common group—even if it is the only choice for them—if they cannot communicate with each other. With extended groups, this situation can be handled by just allowing x and y to join some nearby group as internal nodes. Particularly promising in this respect is a hybrid approach, which allows extended groups only above some particular level in the topology tree.

Furthermore extended groups can be used to construct a modular topology where the regular and more complex construction is used only below some particular tree level. The construction is therefore divided into independent parts and the number of nodes used for the regular topology construction algorithm is reduced according to the particular tree level and therefore independent of the actual number of nodes n in the system. This is particular beneficial for low power devices with a large number of nodes.

7 Conclusions and Future Research

We presented and analyzed a distributed fault-tolerant algorithm for constructing a topology (overlay graph) for fault-tolerant communication in wireless ad hoc networks. The constructed overlay graph is k-regular, k-connected, ensures failure locality and has low total weight. The algorithm adapts to a dynamically changing environment, is guaranteed to converge, and exhibits quite reasonable performance. The hierarchy of clusters reflects the spatial density of the nodes and might replace some additional under- and overlay clusters algorithms.

Part of our current/future research is devoted to some extensions of our approach. First of all, although our algorithm generates topologies with low total weight, the question of their sub-optimality with respect to the unique minimal k-connected and k-regular overlay graph arises. A related question concerns the achievable spanning factors.

References

[1] Bernd Thallner. Fault tolerant communication topologies for wireless ad hoc networks. In *1st Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS'04)*, Florence, Italy, June 2004. http://www.ecs.tuwien.ac.at/W2F/documents/diwans04.pdf.

[2] Ning Li and Jennifer C. Hou. Topology control in heterogeneous wireless networks: Problems and solutions. In *Proceedings of the IEEE INFOCOM'04*, 2004.

[3] Mohammad Taghi Hajiaghayi, Nicole Immorlica, and Vahab S. Mirrokni. Power optimization in fault-tolerant topology control algorithms for wireless multi-hop networks. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pages 300–312, 2003.

[4] M. Bahramgiri, M. Hajiaghayi, and V. Mirrokni. Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks. In *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN02)*, pages 392–397, 2002.

[5] Ning Li and Jennifer C. Hou. FLSS: A fault-tolerant topology control algorithm for wireless networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pages 275–286, 2004.

[6] Bernd Thallner and Ulrich Schmid. Distributed construction of fault-tolerant overlay networks. Research Report 35/2004, Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 1-3/182-2, 1040 Vienna, Austria, 2004. http://www.ecs.tuwien.ac.at/W2F/RR35-2004.pdf.

[7] Heinrich Moser and Bernd Thallner. Distributed construction of fault-tolerant overlay networks: Construction algorithm. Research Report 39/2004, Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 1-3/182-2, 1040 Vienna, Austria, 2004. http://www.ecs.tuwien.ac.at/W2F/RR39-2004.pdf.

[8] R. Guerraoui and A. Schiper. The decentralized non-blocking atomic commitment protocol. In *Proceedings of the 7th IEEE Symposium on Parallel and Distributed Processing (SPDP-7)*, pages 2–9, San Antonio, Texas, USA, 1995.

[9] Bernd Thallner. Distributed construction of fault-tolerant overlay networks: Propose modules. Research Report 38/2004, Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 1-3/182-2, 1040 Vienna, Austria, 2004. http://www.ecs.tuwien.ac.at/W2F/RR38-2004.pdf.