Fault Tolerant Communication Topologies for Wireless Ad Hoc Networks

Bernd Thallner and Ulrich Schmid
Embedded Computing Systems Group E182/2
Institute of Computer Engineering
Vienna University of Technology
Treitlstrasse 3, A-1040 Vienna, Austria
Email: {thallner,s}@ecs.tuwien.ac.at
Tel: +43-1-58801-18203, FAX: +43-1-58801-18297

Abstract— **We present[1] a distributed algorithm which constructs a fixed node-degree overlay network for efficient fault-tolerant multi-hop communication in large distributed systems: Given a fully connected weighted graph, where the weight of an edge represents the cost of a connection, the algorithm builds and maintains a $\overline{\Delta}$-regular subgraph that is $\overline{\Delta}$-node connected, ensures failure locality, and has low total weight. The algorithm automatically adapts to a dynamically changing environment, is guaranteed to stabilize, and exhibits good average case performance as well. It is hence well suited for both establishing fault-tolerant communication topologies in CDMA-based wireless ad hoc networks and for constructing robust overlay graphs in peer-to-peer networks.**

Keywords: FT Communication, FT Mobile Computing/Networking, Distributed Algorithms, Redundant Paths, Overlay Networks

I. Introduction

Ad hoc wireless networks and peer-to-peer systems are two examples of distributed applications that cannot reasonably be modeled as fully connected systems. The huge number of participants and/or resource limitations do not allow a node to communicate with—or even to know—every other node in the system. Rather, every node can communicate directly with a small number of neighbors only. Typical CDMA-based ad hoc networks [1], for example, which are our major target, need some dedicated hardware (e.g. a correlator) on both sender and receiver side of every point-to-point connection. Consequently, given the limited resource/power/... budgets of wireless devices, only a relatively small number (about, 2...20) of connections can be maintained simultaneously. Note that such connections are bidirectional and typically used for multiple messages, since spread spectrum communication involves an expensive communication setup phase (due to synchronization etc.).

In addition, an individual cost figure (*weight*) is usually associated with a connection. It measures how expensive or difficult it is to communicate with a specific peer. Distance, required transmition power, interference level or any combination of such quantities are legitimate cost figures. Note that such connection weights are usually time-dependent: The transmit power required for communicating with a certain node in a CDMA wireless system, for example, does 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 connection to a receiver that temporarily suffers from low battery or heavy processor load may be considered more costly than usual.

From a system's point of view, the connections between direct neighbors form a (sparse) *overlay graph*, which can be used for simulating a fully connected network via multi-hop routing. In order to guarantee efficient communication between any two nodes, the overlay graph must of course be spanning and should have low total weight. Moreover, to guarantee unimpaired communication even in presence of node and link failures,

[1]This research is part of the W2F-project, which targets a wireline/wireless fieldbus based upon CDMA communications, see *http://www.auto.tuwien.ac.at/Projects/W2F/* for details. W2F is supported by the Austrian START programme Y41-MAT.

the overlay network must be robust: It should (1) provide multiple disjoint paths between any pair of nodes to guarantee timely delivery of every single message, despite of failures, and (2) adapt quickly to more persistent changes in the environment. Suitable overlay graphs must hence be more general than spanning trees and have to 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.

Accomplishments: We present a method for constructing a $\overline{\Delta}$-regular overlay graph for efficient fault-tolerant multi-hop communication in networked distributed systems. It is based upon recursively grouping $\overline{\Delta}$ nodes into groups which are treated like single nodes subsequently, thereby building up a group structure corresponding to a $\overline{\Delta}$-ary tree. By choosing groups in a way that minimizes the total weight of all group-internal connections, the (unique) minimum overlay graph (providing the lowest total weight achievable by our method) can be constructed. We prove that the resulting overlay graph is $\overline{\Delta}$ node-connected, which is optimal, and inherently ensures failure-locality. Hence, even excessively many failures in some part of the system do not impair fault-tolerant communication in other parts.

We also give a distributed algorithm which constructs and continuously maintains the minimum overlay graph in a dynamic setting. The algorithm automatically adapts to a changing environment (i.e. node crashes, node joins, varying connection weights) and is guaranteed to stabilize when the environment is stable for sufficiently long. We also provide an extention of our basic algorithm which adapts the overlay graph only in the neighborhood of node failures or node joins. Our analysis[2] of the achieved properties is complemented by a few results of our simulation experiments, which show that the average number of groupings grows linearly with the number of nodes in the system. We are hence convinced that our algorithm is well suited for both establishing fault-tolerant communication topologies in CDMA-based wireless ad hoc networks and for constructing robust overlay graphs in peer-to-peer networks.

[2]Lack of space does not allow us to include a complete performance analysis in this paper.

Related Work: Most work in this area either deals with nodes that are position-aware [2] or is based on the *unit disk graph* (UDG) model [3] where a link between two nodes is only possible if their Euclidean distance is at most one. On top of this model, a number of solutions [4], [5], [6], [7], [8], [9], [10], [11], [12] for constructing spanners have been proposed. A spanner is an overlay graph, which guarantees that any two nodes have a distance of at most a constant factor times the minimum distance in the UDG graph. Distributed algorithms for constructing spanners have been presented [4], [5], [9], [11], [12] as well. Still, none of those solutions ensures redundant paths for fault-tolerant communication and maintains a fixed and small node degree. In [13] an algorithm for transforming an arbitrary spanner into a fault-tolerant spanner is presented; it is not distributed, however. Finally, there are probabilistic algorithms for the related problem of constructing fault-tolerant overlay networks for peer-to-peer systems [14].

The remainder of this paper is organized as follows: In Section II we define our basic notation. Section III introduces our group construction method and proves the properties of the resulting overlay graph. We present a distributed algorithm that constructs the overlay graph in a dynamic setting in Section IV. Section V provides some extensions of our method, and Section VI presents a few results of our simulation experiments. A brief summary of our accomplishments and directions of further work in Section VII concludes the paper.

II. Definitions

We consider simple undirected weighted graphs $G = (\Pi, \Lambda)$ consisting of a set of *nodes* $\Pi = \{1, ..., n\}$ and a set of weighted *edges* $\Lambda \subseteq \Pi \times \Pi \times \mathbb{R}$ with $(x, x, w) \notin \Lambda$ for all $x \in \Pi$. A graph is called *κ-node-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 said to be *regular* of degree r if all nodes have degree r.

Our network is modeled as a specific simple undirected weighted graph G, called communication graph, which is assumed to be fully connected.[3] The set of nodes and the weight of edges is time dependent but we assume the graph to be stable sufficiently long for our algorithm to stabilize. The presented algorithm finds a low weight *overlay graph* $G' = (\Pi, C)$ with $C \subseteq \Lambda$ for $n \geq \overline{\Delta}^2$ that is $\overline{\Delta}$-regular and $\overline{\Delta}$-node-connected if $n \cdot \overline{\Delta}$ is even or $\overline{\Delta} - 1$-node-connected if odd. In order to easily distinguish the communication graph G and the overlay graph G', the edges of the latter will be called *connections*.

III. Topology Construction

We first present a method for constructing a topology that induces an overlay graph G' with the desired properties. The idea behind the topology construction is to build structures that appear like nodes such that they can be treated like nodes subsequently. We will then give a distributed algorithm that constructs this topology and hence the overlay graph G'.

Figure 1 gives an example where (c) is the fully connected communication graph G, (b) depicts the constructed overlay graph G' for $\overline{\Delta} = 3$, and (a) gives a tree representation corresponding to the constructed topology. From (b) it is apparent that nodes $(1, 2, 3)$, $(5, 6, 7)$ and $(8, 9, 10)$ are combined to 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. $5 - 6$, $5 - 7$, $6 - 7$) is minimal. Each of the $\overline{\Delta}$ members of a group is connected to all of the $\overline{\Delta} - 1$ other members (internal connections) and has exactly one connection left (external connection). Since there are $\overline{\Delta}$ members in a group, the group has—like a node—$\overline{\Delta}$ external connections left, which are available in higher level groups. For example, group C again consists of three members: A single node 4 and two groups A and B, which are connected via their external connections. Again all members of C agree upon minimality of the weight sum of its internal connections. Group R finishes the topology and consists of group C and D; note that it is incomplete in that it has less than $\overline{\Delta}$ members. The tree representation (a) depicts the group structure via a $\overline{\Delta}$-ary tree (except for the root, which corresponds to the group R and may hence have $< \overline{\Delta}$ children). The edges of the tree represent the membership relation among the groups.

We now give a formal description of our topology. We introduce *groups* which consist of an identifier g_i[4] 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 $\overline{\Delta}$ nodes and groups: $members(g_i) \subseteq (\mathbb{G} \cup \Pi)$. A node or group can only be member of a single group[5] $\forall g_a, g_b \in \mathbb{G}, g_a \neq g_b : x \in members(g_a) \wedge y \in members(g_b) \Rightarrow x \neq y$. For every $g_i \in \mathbb{G}$ we define the nodes of a group as $nodes(g_i) = \bigcup_{k=0}^{\infty} members^k(g_i) \cap \Pi$ where $members^0(g_i) = members(g_i)$ and $members^j(g_i) = \bigcup_{k \in members^{j-1}(g_i) \wedge k \in \mathbb{G}} members(k)$. For every node $p \in \Pi$ we define $members(p) = \{p\}$ and $nodes(p) = \{p\}$.

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 $g_a, g_b \in members(g_i)$. If there is a connection $(p, q, w) \in C$ from 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: $\forall a, b \in members(g_i), a \neq b, p \in nodes(g_a), q \in nodes(g_b) \Rightarrow \exists (p, q, w) \in C$.

Hence, every group member has $\overline{\Delta} - 1$ connections to other group members and therefore exactly one connection left. Since there are $\overline{\Delta}$ members in a group, the group has—like a node—$\overline{\Delta}$ connections left. We call the $\overline{\Delta}$ 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 $p \in \Pi$ and say that every node has $\overline{\Delta}$-node-disjoint paths to itself. The terminal node of a group with the smallest id is called *leader* of the group.

The weight of a group $\omega(g_i)$ is a tuple (A, B) where A is the sum of all group g_i internal connection weights and B is the ID of the group leader. A group g_i has smaller weight than g_j, formally $\omega(g_i) < \omega(g_j)$, if the sum of the internal connection weights A_i is smaller than A_j or, if equal, the leader ID B_i is lower than B_j.

Definition 1: An overlay graph G' is called *admissible* if its corresponding topology has a

[3]We drop the fully connected graph assumption when we introduce the extension of our algorithm in Section V.

[4]In an implementation the set of terminal nodes can be used as group identifier (explained later).

[5]Note that the member function is not transitive: $a \in members(g_i) \wedge g_i \in members(g_j) \Rightarrow a \notin members(g_j)$.

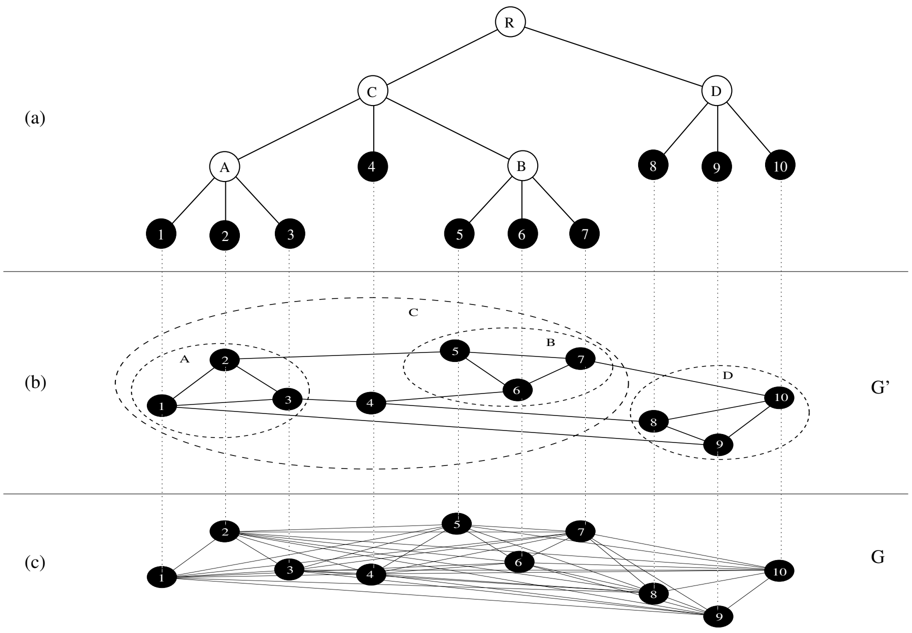


Fig. 1. Communication graph, network graph and topology

single *root group* g_{root}[6] with $members(g_{root}) \subseteq \mathbb{G}$ and $2 \leq |members(g_{root})| \leq \overline{\Delta}$. The root group is constructed as a ring connecting $\left\lfloor \frac{\overline{\Delta}}{2} \right\rfloor + \left\lceil \frac{\overline{\Delta}}{2} \right\rceil$ terminal nodes of every subgroup with its neighboring subgroups.

Note that single-node members in the root group are forbidden[7] in an admissible overlay graph. Moreover, we actually allow overlay graphs that violate $\overline{\Delta}$-regularity in the root group: In case of $n \cdot \overline{\Delta}$ being odd, a single terminal node of a root member group may have an unused connection left.

Theorem 1: For every graph G with $n \geq \overline{\Delta}^2$, there exists[8] an admissible overlay graph G'.

Proof. Every admissible topology consists of $A \cdot \overline{\Delta} + B \cdot (\overline{\Delta} - 1)$ leaf nodes, where A with $2 \leq A \leq \overline{\Delta}$ corresponds to $|members(g_{root})|$ and $B \geq 0$ corresponds to the total number of all other subgroups $|\mathbb{G} \setminus members(g_{root})|$. Every of the A subgroups of g_{root} has $\overline{\Delta}$ nodes, and every of the B other subgroups is attached to the subtree rooted at any member of g_{root}. Attaching is done by replacing one node with a group (thereby reducing the number of leaf nodes by 1) consisting of $\overline{\Delta}$ new nodes, which amounts to a total of $\overline{\Delta} - 1$ new nodes. The proof is by induction. For $n_0 = \overline{\Delta}^2$ the claim follows trivially, since $\overline{\Delta}^2 = \overline{\Delta} \cdot \overline{\Delta} + 0 \cdot (\overline{\Delta} - 1)$ implies that there is a topology with $\overline{\Delta} - 1$ members in g_{root} and one subgroup of a root member.

For each $n_i = n_{i-1} + 1$, $i \geq 1$, there are two cases:

1) If the number of members in g_{root} in the topology for n_{i-1} is $2 \leq |members(g_{root})| \leq \overline{\Delta} - 1$, then $n_{i-1} = A \cdot \overline{\Delta} + B \cdot (\overline{\Delta} - 1)$ with $B \geq 1$ since $A \leq \overline{\Delta}$ and $n \geq \overline{\Delta}^2$. Consequently, for n_i, a topology $n_i = (A+1) \cdot \overline{\Delta} + (B-1) \cdot (\overline{\Delta} - 1)$ exists. For example, any subgroup $g_i \notin members(g_{root})$ can be replaced by the additional node, whereas g_i becomes member of the root group.
2) If the topology for n_{i-1} satisfies $|members(g_{root})| = \overline{\Delta}$, then $n_{i-1} = \overline{\Delta} \cdot \overline{\Delta} + B \cdot (\overline{\Delta} - 1)$, and a topology $n_i = 2 \cdot \overline{\Delta} + (B + \overline{\Delta} - 1) \cdot (\overline{\Delta} - 1)$ exists for n_i. For example, we could just remove $\overline{\Delta} - 2$ groups from the root group and replace some arbitrary node by a group consisting of the $\overline{\Delta} - 2$ removed groups, the replaced node, and the single additional

[6]Note that $g_{root} \notin \mathbb{G}$.

[7]Root groups with $members(g_{root}) = \{g_i, p\}$, $g_i \in \mathbb{G}$ and $p \in \Pi$ would in fact be feasible; we do not deal with this special case for simplicity, however.

[8]Note that $n \geq \overline{\Delta}^2$ is not necessarily a tight bound.

node.

Hence, the claimed existence of at least one topology for every graph G with $n \geq \overline{\Delta}^2$ follows. $\square$

By introducing a suitable minimum criterion, we can even stipulate the existence of a unique *minimal* admissible overlay graph. Although it is not necessarily the global minimum-weight overlay graph, it is nevertheless the minimum one w.r.t. all alternative admissible topology constructions. Informally, the minimum criterion for choosing group members requires that the selected group has minimal total weight of all internal connections, and that all members agree upon this fact. It may hence be the case that there is a lower-weight choice for some members, but no one that is better for all of them. Note that choosing a group according to this criterion does not lead to a violation of the minimality of any already existing group in the final topology.

Definition 2: An admissible overlay graph G' is *minimal* if, for all members $x \in members(g_i)$ of any group $g_i \in \mathbb{G}$ in the corresponding topology tree, no other group g' can be built with $x \in members(g')$, $\omega(g') < \omega(g_i)$ and $\forall y \in members(g') : \omega(g') < \omega(g_y)$, where g_y is the (unique) group in the topology tree with $y \in members(g_y)$.

Theorem 2: For every graph G with $n \geq \overline{\Delta}^2$, there exists a unique minimal admissible overlay graph G'.

Proof. By contradiction. 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 2. $\square$

In the following, we will establish a few additional properties of our overlay graphs.

Theorem 3: Each admissible overlay graph G' is $\overline{\Delta}$-regular if $n \cdot \overline{\Delta}$ is even.

Proof. Follows immediately from the topology construction. $\square$

Property 1: Each admissible graph G' has $\left\lfloor \frac{n-1}{\overline{\Delta}-1} \right\rfloor$ groups, including the root group, such that $|\mathbb{G} \cup \{g_{root}\}| = \left\lfloor \frac{n-1}{\overline{\Delta}-1} \right\rfloor$ different group identifiers are sufficient.

Proof. W.l.o.g. we fill up the root group with $0 \leq m \leq \overline{\Delta} - 2$ dummy nodes until g_{root} has $\overline{\Delta}$ members, which does not change the tree structure below the root group. The resulting topology tree is a $\overline{\Delta}$-ary tree with $n' = n + m$ leaf-nodes, which has $\frac{n'-1}{\overline{\Delta}-1}$ internal-nodes according to [15, Ex. 2.3.4.5.6]. We prove that $X = \left\lfloor \frac{n-1}{\overline{\Delta}-1} \right\rfloor$ is the only integer number $X \in \mathbb{N}$ for $0 \leq m \leq \overline{\Delta} - 2$ that equals $\frac{n'-1}{\overline{\Delta}-1}$. For $m = 0$ this follows trivially. For $m = \overline{\Delta} - 2$, we only have to show that no greater X fulfills $X \leq \frac{n+(\overline{\Delta}-2)-1}{\overline{\Delta}-1}$. Since $X+1 = \left\lfloor \frac{n-1}{\overline{\Delta}-1} \right\rfloor + 1 > \frac{n+(\overline{\Delta}-2)-1}{\overline{\Delta}-1} = \frac{n-1}{\overline{\Delta}-1} + \frac{\overline{\Delta}-2}{\overline{\Delta}-1}$, the claim follows. $\square$

Now we are ready for showing that our overlay graph is $\overline{\Delta}$-connected. Still, we need a few preliminary lemmas for this purpose.

Lemma 1: Let g be a group and $T_g \subseteq nodes(g)$ with $|T_g| = \overline{\Delta}$ be its set of terminal nodes. Then, every $p \in nodes(g), g \in (\mathbb{G} \cup \Pi)$, has $\overline{\Delta}$-node-disjoint paths to the nodes in T_g.

Proof. Induction on depth d in the topology tree. For $d = 0$, g is a node; the claim hence follows trivially from our convention. For some $d > 0$, suppose that the claim holds for every sub-group $g_i \in members(g)$. Clearly, $p \in nodes(g_i)$ for some specific i and has hence $\overline{\Delta}$-node-disjoint paths to T_{g_i}. By the construction of g, every $q \in T_{g_i} \setminus T_g$ is connected to one terminal node q_j in every T_{g_j} with $g_j \in members(g) \wedge g_i \neq g_j$. Applying the induction hypothesis again, q_j is connected to every node in T_{g_j} and hence also to the one in $T_{g_j} \cap T_g$. For $q \in T_{g_i} \cap T_g$, a connection to some node in T_g is obvious. $\square$

Lemma 2: Let $S \subseteq T_{g_i}$ with $|S| \leq \frac{\overline{\Delta}}{2}$ be a subset of group g_i's terminal nodes, where $g_i \in \mathbb{G}$. Then, any $p \in S$ is connected to any terminal in $T_{g_i} \setminus S$ via $|S|$ node-disjoint paths.

Proof. From Lemma 1, it follows that any terminal $q \in T_{g_i}$ is connected to every terminal of its subgroup. Hence, w.r.t. connectivity, groups and single nodes are equivalent. Since the subgroups in g_i are fully connected, the lemma follows trivially. □

Lemma 3: Let g_{root} be the root group, consisting of $2 \le x \le \overline{\Delta}$ members and $n \ge \overline{\Delta}^2$. Then the terminal nodes of any two sub-groups g_a and $g_b \ne g_a$ are connected via $\overline{\Delta}$ node-disjoint paths in case of $n \cdot \overline{\Delta}$ is even, or $\overline{\Delta} - 1$ node-disjoint paths otherwise.

Proof. Each group has $\overline{\Delta} \cdot (\overline{\Delta} - 1)$ group internal connections, which is always even. Since the number of groups is an integer, the number of connections in the root group is even if $n \cdot \overline{\Delta}$ is even or odd if $n \cdot \overline{\Delta}$ is odd. Therefore a ring connecting $\left\lfloor \frac{\overline{\Delta}}{2} \right\rfloor + \left\lceil \frac{\overline{\Delta}}{2} \right\rceil$ terminal nodes of every subgroup with its neighboring subgroups can be built if $n \cdot \overline{\Delta}$ is even. The same ring with one connection missing can be built if $n \cdot \overline{\Delta}$ is odd. The claimed connectivity follows from Lemma 2. □

Theorem 4: Each admissible graph G' with $n \ge \overline{\Delta}^2$ is $\overline{\Delta}$-node-connected if $n \cdot \overline{\Delta}$ is even, or $\overline{\Delta} - 1$-node-connected if $n \cdot \overline{\Delta}$ is odd.

Proof. Let p and q be two arbitrary nodes and $g \in \mathbb{G}$ be the lowest-level group such that p and $q \in nodes(g)$. We first show that p, q are connected via $\overline{\Delta}$ respectively $\overline{\Delta} - 1$ node-disjoint paths, which implies the statement of our theorem via Menger's Theorem [16].

(1) If g_i is a regular group, $p \in nodes(g_a)$ and $q \in nodes(g_b)$ with $g_a, g_b \in members(g)$ and $g_a \ne g_b$, we claim that there are $\overline{\Delta}$ node-disjoint paths between the two sets of terminals T_{g_a} and T_{g_b}. By the construction of a group, $\overline{\Delta} - 1$ pairs of nodes of $T_{g_a} \times T_{g_b}$ are connected by disjoint paths routed over at most one sub-group. According to Lemma 2, there is always a connection between any pair of terminals of this subgroup. Hence, we only have to show that the external connections of T_{g_a} and T_{g_b} are connected outside of g as well. Consider the parent group g' of g.
   a) If the terminal nodes $p' \in T_{g_a}$ and $q' \in T_{g_b}$ of group g are non-terminals in g', they are connected in g' via at most one other subgroup of g', according to Lemma 2.
   b) If w.l.o.g. $p' \in T_{g'}$ but $q' \notin T_{g'}$, then q' is connected to any terminal $q^* \in T_{g'}$ via at most one other subgroup of g' by Lemma 2.
   c) If both p' and q' (or q^*) are terminals in g', go up to the parent group of g' until either (a) applies or the root group is reached, where Lemma 3 guarantees connectivity between p' and q'.

(2) g is the root group g_{root}, with $p \in nodes(g_a)$, $q \in nodes(g_b)$, $g_a, g_b \in members(g)$ and $g_a \ne g_b$. The required connectivity follows immediately from Lemma 3.

Applying Menger's Theorem, the claimed connectivity follows. □

The tree structure of our topology reveals another benefit of our overlay graph with respect to fault-tolerant communication. Failures that hit some part of the tree do not severely—if at all—impair other parts. The proof of Theorem 4 reveals that failures that completely wipe out all members of some group g do not impair $\overline{\Delta}$ node-connectivity of the remaining tree that results from purging the subtree rooted at g. Moreover, all nodes within a subtree rooted at some member of g are still $\overline{\Delta} - 1$ node-connected with each other, since only the single external connection is cut.

In addition, if the weights in the communication graph reflect physical distance, our minimal topology construction groups nodes according to their spatial density. Nodes that are close to each other will be near the leafs of the topology tree, which leads to a nice fault-locality property of our overlay graph: Catastrophic failures in a spatially localized area will affect communication only in the immediate neighborhood. Combined with additional measures for self-healing, by continuous adaption of the overlay graph, this leads to excellent fault-tolerance properties.

IV. The Algorithm

In this section, we present a distributed algorithm that builds up and continuously maintains the minimal admissible overlay graph in a dynamic environment. We assume a fully connected distributed system of nodes that can crash and recover here, where links are reliable (dropped later) and connection weights are possibly time-variant. Moreover, the system must be sufficiently synchronous to allow the implementation of unreliable failure detectors, like the one of [17].

Our algorithm works by letting every group-leader of an existing group concurrently search for the minimal-weight next-level group to join. Every node repeatedly generates proposals for this purpose, consisting of the group members, the group weight, the group internal connections and the group's terminal nodes, which are sent to (the leaders of) the proposed members for confirming minimality. In analogy to unreliable failure detectors [18], in particular, the leader-based Ω, we encapsulate the functionality of generating proposals in an abstract *propose module*, which is simply assumed to eventually generate the group proposal with minimum weight w.r.t. the topology generated so far. For example, an implementation could acquire information about weights and existing groups via explore messages from its surrounding and calculate group proposals based on the received information. Generating a new proposal may be triggered periodically, to adapt to changed weights, and upon detection of a node crash or join. Lacking space does not allows us to elaborate on the wide variety of possible implementations of the propose module; a forthcoming paper will be devoted to those issues.

The pseudo-code of our algorithm for topology construction is given in Figure 2. ID is the unique id of a node; the id of a group is the set of its terminal node ids. The variable $Group(i)$ at every member of group i stores the group relevant information: The group members $Group(i).Members$, the group weight $Group(i).Weight$, the group internal connections $P.Connections$ and the terminal nodes $P.Terminals$. Note that the block for received PROPOSAL messages is presented without handling of root groups (extended later). The simple blocks for received BUILD_CONNECTION and CANCEL_CONNECTION messages are omitted for brevity.

The algorithm consists of the following concurrently executed atomic blocks:
The main loop (`line 3-12`) is executed when a proposal P for a new group is provided by the propose module. The proposal is sent to all group members first, and if everyone accepts the new group, it is built via JOIN_GROUP messages. If any member rejects the proposal is discarded and an UNLOCK message is sent.

If a member receives a PROPOSAL message (`line 13-24`), it decides whether the proposed group is better than its current one and replies with an ACCEPT or REJECT message. If a member accepts the proposed group, it locks out further proposal messages until the new group is built or an UNLOCK message is received.

If a member receives a JOIN_GROUP message (`line 25-39`), it cancels its possibly existing current membership by sending a CANCEL_GROUP_PARENT message to its leader and stores the proposal P in its $Group$ variable. The new group id, which is the set of the terminal node ids $P.Terminals$, is stored in $ParentID$ (`line 34`). If the node is the leader of the new group, it sends BUILD_CONNECTION messages to all members for generating the internal connections. A node that receives a BUILD_CONNECTION message builds the corresponding connection.

If the leader of a group receives a CANCEL_GROUP_PARENT message (`line 40-46`), it also cancels its parent group with a CANCEL_GROUP_PARENT message and forces all group members with a CANCEL_GROUP message to leave the group. Finally, all internal connections are canceled via CANCEL_CONNECTION messages. A node that receives a CANCEL_CONNECTION message cancels the corresponding connection.

A member that receives a CANCEL_GROUP message (`line 47-49`) clears its group reference in $ParentID$ and a node that receives a UNLOCK message (`line 50-52`) removes its lock.

```
VAR Group(ID) ←⊥                              /* stores group information */
VAR LOCK ←⊥                                   /* lock between ACCEPT and JOIN_GROUP message */

do                                            /* Main loop, periodically executed on every node */
    wait for proposal P with P.Members, P.Weight, P.Connections and P.Terminals
    for all x ∈ P.Members do send (PROPOSAL,P,x) to leader of x
    receive (ACCEPT,P) from X
    if X = P.Members
        for all x ∈ X do send (JOIN_GROUP,P,x) to leader of x
    else
        for all x ∈ X do send (UNLOCK) to leader of x
    end
end

if received (PROPOSAL,P,i) from j             /* Block for PROPOSAL messages */
    if LOCK = P
        send (ACCEPT,P) to j
    else if LOCK ≠⊥
        send (REJECT,P) to j
    else if Group(i) =⊥ ∨ Group(i).ParentID ≠⊥ ∨
            P.Weight < Group(Group(i).ParentID).Weight ∨
            (P.Weight = Group(Group(i).ParentID).Weight ∧ min(P.Terminals) < min(i))
        LOCK ← P
        send (ACCEPT,P) to j
    else
        send (REJECT,P) to j
    end
end

if received (JOIN_GROUP,P,i)                  /* Block for JOIN_GROUP messages */
    if Group(i) ≠⊥ ∧ Group(Group(i).ParentID) = P
        exit block                            /* Ignore duplicate proposals */
    end
    if Group(i) ≠⊥ ∧ Group(Group(i).ParentID) ≠⊥
        send (CANCEL_GROUP_PARENT,Group(i).ParentID) to leader of Group(i).ParentID
    end
    wait until Group(i).ParentID =⊥
    Group(P.Terminals) ← P
    Group(i).ParentID ← P.Terminals
    LOCK ←⊥
    if min(P.Terminals) = ID
        for all c ∈ P.Connections do send (BUILD_CONNECTION,c) to endpoints of c
    end
end

if received (CANCEL_GROUP_PARENT,i)           /* Block for CANCEL_GROUP_PARENT messages */
    if Group(i) ∧ Group(i).ParentID ≠⊥
        send (CANCEL_GROUP_PARENT,Group(i).ParentID) to leader of Group(i).ParentID
    end
    for all x ∈ Group(i).Members do send (CANCEL_GROUP,x) to leader of x
    for all c ∈ Group(i).Connections do send (CANCEL_CONNECTION,c) to the endpoints of c
end

if received (CANCEL_GROUP,i)                  /* Block for CANCEL_GROUP messages */
    Group(i).ParentID ←⊥
end

if received (UNLOCK)                          /* Block for UNLOCK messages */
    LOCK ←⊥
end
```

Fig. 2. Topology construction algorithm for node ID

A. Root Group Handling

Building the final root group requires a modification of the receive PROPOSAL messages block and an extension of the specification of the propose module according to Definition 1: A root group proposal must satisfy $members(g_{root}) \subseteq \mathbb{G}$ and $2 \leq |members(g_{root})| \leq \overline{\Delta}$ and does always exist if $n \geq \overline{\Delta}^2$, see Theorem 1. Hence, if $2 \leq x \leq \overline{\Delta}$ groups without a parent group are left, the proposal module generates a root group proposal with id $P.Terminals = \emptyset \neq \perp$ that connects its members in a ring. Eventually the root group proposal must have minimum weight as usual.

In addition, the code in Figure 3 replaces `line 12-24` from Figure 2. A group leader now accepts a group proposal if it is not yet member in a group, if the proposal has less members, or the same number of members but less weight.

Due to space limitations, we cannot give a full correctness proof and performance analysis in this paper. The following Theorem 5 shows, however, that the algorithm eventually stabilizes and constructs the unique minimum admissible overlay graph.

Theorem 5: The algorithm of Figure 3 eventually stabilizes and constructs the unique minimum admissible overlay graph G' for every communication graph G with $n \geq \overline{\Delta}^2$, provided G remains stable sufficiently long.

Proof. By contradiction. Let us assume that, even at times after the stabilization time, there is always a cycle $C = \big((G')^1, (G')^2, \dots (G')^k, (G')^{k+1} = (G')^1, \dots\big)$ for some $k \geq 2$, where any $(G')^e$ is an incompletely built overlay graph that differs from $(G')^{e+1}$ in that a single additional group $g_i^{e+1} \in (G')^{e+1}$ has been built (and the former groups of the new members been destroyed, if any). By the properties of the propose module, every additional group must be the minimal one. For an arbitrary $1 \leq a \leq k$, consider some member x of a group $g \in (G')^a$ with weight $\omega(g)$. We claim that, in $(G')^{a+1}$, there is at least one group g' with weight $\omega(g') < \omega(g)$ (except in case (1), where $\omega(g') = \omega(g)$).

Actually, there are 5 exhaustive cases:

(1) x remains in the same group $g \in (G')^{a+1}$, hence $g' = g \in (G')^{a+1}$.

(2) x is moved into the additional group g_i^{a+1}, hence $g' = g_i^{a+1}$ since the minimum criterion of Definition 2 requires $\omega(g_i^{a+1}) < \omega(g)$ for moving x.

(3) x is freed by moving some other member y of g into the additional group g_i^{a+1}, hence $g' = g_i^{a+1}$ by the same argument applied to y.

(4) x is freed by destroying some other member y of g, by moving some of y's members (y must of course be a group here) into the additional group g_i^{a+1}. Since the proposal modul must generate the minimum group proposal, this implies that cases (1)–(3) would have led to a group g'' with weight $\omega(g'') \geq \omega(g)$ larger than $\omega(g_i^{a+1})$. Hence, $g' = g_i^{a+1}$ again.

(5) x is destroyed by moving some of x's members (x must of course be a group here) into the additional group g_i^{a+1}. By the same argument as in case (4), $\omega(g_i^{a+1}) < \omega(g)$.

Let x be the last member that drops into a lowest-weight group g' during the entire cycle, by one of the cases (2)–(5), so $\omega(g) > \omega(g')$. This member cannot escape from g', even at the end of the cycle $(G')^1$, hence cannot join the higher-weight group g again.

Since the algorithm applies the minimum criterion of Definition 2, it constructs a minimum overlay graph, which is unique by Theorem 2. □

V. EXTENSION

A drawback of the presented topology is that it is not efficient for very dynamic environments. Each node that joins or leaves the network requires a restructuring of the topology. In worst case this can lead to a reconstruction of great parts of the topology. Assume the case where a node $p \in \Pi$ and $p \in T_{g_i}$ with $g_i \in members(g_{root})$ leaves the network. Then at least all groups g with $\{p\} \in T_g$ have to be rebuilt.

Another disadvantage, directly related to this problem concerns the effectiveness of the topology construction algorithm. Since group construction is done on each node and group leader, the topology is built fully concurrent. Therefore it can happen that a low level group which is included in many other groups of the minimal solution is built late and thereby many already constructed but suboptimal groups have to be reconstructed.

This section therefore present an extension to the topology introduced in section III that allows

```
13  if received (PROPOSAL,P,i) from j              /* Block for PROPOSAL messages */
14      if LOCK = P
15          send (ACCEPT,P) to j
16      else if LOCK ≠⊥
17          send (REJECT,P) to j
18      else if Group(i) =⊥ ∨ Group(i).ParentID ≠⊥ ∨
                P.Members > Group(Group(i).ParentID).Members ∨
                (P.Members = Group(Group(i).ParentID).Members ∧
                (P.Weight < Group(Group(i).ParentID).Weight ∨
                (P.Weight = Group(Group(i).ParentID).Weight ∧ min(P.Terminals) < min(i)))
19          LOCK ← true
20          send (ACCEPT,P) to j
21      else
22          send (REJECT,P) to j
23      end
24  end
```

Fig. 3. Code for PROPOSAL messages with root group handling; replaces (`line 12-24`)

nodes to join or leave a group without reconstruction of the topology and thereby keeping changes local. The extended topology induces a low weight overlay graph $G^* = (\Pi, C)$ with $C \subseteq \Lambda$ that is $\overline{\Delta}$-regular and $\overline{\Delta}$-node-connected if $n \geq \overline{\Delta}^2$ and $\overline{\Delta}$ is even or $\overline{\Delta} - 1$-node-connected if $\overline{\Delta}$ odd.

Figure 4 (a) gives an example of a group and its parent groups with $\overline{\Delta} = 4$, (b) and (c) depicts the restructured group if node 2 respectively 4 is removed. The groups A and B are merged together to a new group B' which has more than $\overline{\Delta}$ 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 the nodes 1 and 3 respectively 1 and 2 have no external connections.

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

Lemma 4: Let g^* be an extended group and $T_{g^*} \subseteq nodes(g^*)$ with $|T_{g^*}| = \overline{\Delta}$ be its set of terminal nodes. Then, every $p \in nodes(g^*), g^* \in \mathbb{G}$ has $\overline{\Delta}$-node-disjoint paths to the nodes in T_g if $\overline{\Delta}$ is even.

Proof. Induction on depth d in the topology tree. For $d = 0$, g^* is a node; the claim hence follows trivially. For some $d > 0$, suppose that the claim holds for every sub-group $g_i \in members(g^*)$. Clearly, $p \in nodes(g_i)$ for some specific i and has hence $\overline{\Delta}$-node-disjoint paths to T_{g_i}. There are two cases:

1) If g_i is an internal member of g^* then by the construction every $q \in T_{g_i}$ is connected to one terminal node q_j in every T_{g_j} with g_j terminal member of g^*.
2) If g_i is a terminal member of g^*. For $q \in T_{g_i} \cap T_g$ the connection is obvious. We now show that every $q \in T_{g_i} \setminus T_g$ has a connection to one terminal node q_j in every T_{g_j} with g_j one of the $\overline{\Delta} - 1$ remaining terminal member of g^*. By the construction $\overline{\Delta} - 1 - I_{g^*}$ terminal nodes of $q \in T_{g_i}$ are directly connected to one terminal node q_j in T_{g_j} of the terminal member g_j. The remaining I_{g^*} terminal nodes of $q \in T_{g_i}$ are each one indirectly connected via a separate internal member g_j' with one terminal node q_j in T_{g_j} of the remaining terminal members g_j. By the induction hypothesis each terminal node in $T_{g_j'}$ has a connection to all terminal nodes in $T_{g_j'}$ and by the construction a terminal node $T_{g_j'}$ of a internal member g_j' has a

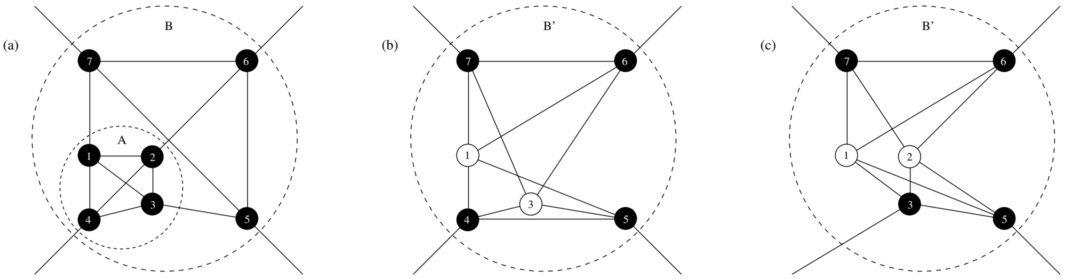


Fig. 4. Regular and extended group structure

connection to a terminal node q_j in T_{g_j} of the terminal member g_j.

Applying the induction hypothesis again, q_j is connected to every node in T_{g_j} and hence also to the one in $T_{g_j} \cap T_g$. □

Since Lemma 2 only relies on Lemma 1 which is equal to Lemma 4 for extended groups Lemma 2 follows also for extended groups.

Lemma 5: Let T_{g_a} and T_{g_b} be two sets of terminal nodes of the extended group g_i^* with $g_a, g_b \in members(g_i^*)$. There are α node-disjoint paths between the two sets and β terminal nodes of T_{g_a} respectively β of T_{g_b} each connected (node-disjoint to α paths) with a separate terminal node of $T_{g_i^*}$ so that $\alpha + \beta = \overline{\Delta}$ and $\beta \leq \frac{\overline{\Delta}}{2}$ if $\overline{\Delta}$ is even.

Proof. There are three cases:

1) If g_a and g_b are internal members of g_i^*, by the construction there are $\alpha = \overline{\Delta}$ node-disjoint paths between T_{g_a} and T_{g_b} routed over $\overline{\Delta}$ terminal members. The claim follows trivially.
2) If g_a and g_b are terminal members of g_i^* then we show that there are α' direct connections, α'' node-disjoint paths routed over internal members, and α''' node-disjoint paths (and node-disjoint to α'' paths) routed over a single terminal member and β terminal nodes of T_{g_a} respectively β of T_{g_b} each connected (node-disjoint to the α paths) with a separate terminal member of g_i^* and therefore with a terminal nodes of $T_{g_i^*}$. We show that $\alpha' + \alpha'' + \alpha''' + \beta = \overline{\Delta}$. Each extend group has $I_{g_i^*}$ internal members therefore $\alpha'' = I_{g_i^*}$ paths routed over internal members exist. For each terminal members $|T_{g_i^*} \cap T_{g_a}| = 1$ respectively $|T_{g_i^*} \cap T_{g_b}| = 1$, therefore $\beta = 1 + \beta'$. If $I_{g_i^*} = \overline{\Delta} - 2$ each terminal member has only a single connection to a other terminal member, therefore $\beta' = 1$, $\alpha' = \alpha''' = 0$ and the claim follows. For $I_{g_i^*} = \overline{\Delta} - 3$ the terminal members are connected in a ring with $\alpha''' = 2$ and $\alpha' = \beta' = 0$; the claim follows. If $0 \geq I_{g_i^*} \geq \overline{\Delta} - 4$ we assume two sets of terminal members A and B each connected to g_a respectively g_b. By the construction $|A| = |B| = \overline{\Delta} - 1 - I_{g_i^*}$. Therefore $\alpha' = |A \cap \{g_b\}| = |B \cap \{g_a\}|$, $\alpha''' = |A \cap B|$ and $\beta' = |A \setminus B| = |B \setminus A|$ the claim follows.
3) If g_a an internal member and g_b a terminal members of g_i^* w.l.o.g. we show that there are α' direct connections, α'' node-disjoint paths routed over terminal members, α''' node-disjoint paths (and node-disjoint to α'' paths) routed over terminal and internal members (2-member-hops) and β terminal nodes of T_{g_a} respectively β of T_{g_b} each connected (node-disjoint to the α paths) with a separate terminal member of g_i^* and therefore with a terminal nodes of $T_{g_i^*}$. We show that $\alpha' + \alpha'' + \alpha''' + \beta = \overline{\Delta}$. By the construction each internal member has a connection to each terminal member so $\alpha' = 1$. Each terminal member has $\overline{\Delta} - I_{g_i^*} - 1$ connections to other terminal members and each internal member has a connection to these terminal member therefore $\alpha'' = \overline{\Delta} - I_{g_i^*} - 1$. There are $I_{g_i^*} - 1$ internal members of g_i^* connected to each terminal member and $\overline{\Delta} - (I_{g_i^*} - 1)$ unused terminal members, therefore $I_{g_i^*} - 1$ node-disjoint paths of type ($g_a \rightarrow$ terminal member $\rightarrow$ internal member $\rightarrow g_b$) exist and $\alpha'' = I_{g_i^*} - 1$. Since there are $\overline{\Delta}$ terminal members, the α'' paths uses $\overline{\Delta} - I_{g_i^*} - 1$ terminal

members, α''' paths uses $I_{g_i^*} - 1$ there are 2 terminal members unused by $alpha$ paths. So $|T_{g_i^*} \cap T_{g_b}| = 1$ follows trivially and g_a has one connection to a single unused terminal member and therefore $\beta = 1$. The claim follows.

We apply Lemma 2 for connections routed over groups. □

Lemma 6: Let $A, B \subset T_{g_i^*}$ be two disjoint subsets of the terminal nodes of group $g_i^* \in \mathbb{G}$ with $|A| = |B| = \frac{\overline{\Delta}}{2}$. Then there are $\frac{\overline{\Delta}}{2}$ node-disjoint paths between the sets A and B if $n \cdot \overline{\Delta}$ is even.

Proof. From the construction each terminal node of $T_{g_i^*}$ is connected (possibly routed over other groups, refer Lemma 2) to a separate terminal node of $T_{g'}$ with $g' \in members(g_{root})$. From Lemma 2 and 3 follows that the terminal nodes in $T_{g'}$ are connected via $\overline{\Delta}$ node-disjoint paths, therefore $\frac{\overline{\Delta}}{2}$ node-disjoint paths between the subsets A and B exists. □

Theorem 6: The graph G' with extended groups is $\overline{\Delta}$-node-connected if $n \geq \overline{\Delta}^2$ and $\overline{\Delta}$ is even.

Proof. Let p, q be two arbitrary nodes and $g^* \in \mathbb{G}$ be the lowest-level extend group such that $p, q \in nodes(g^*)$. We first show that p, q are connected via $\overline{\Delta}$ node-disjoint paths, which implies the statement of our theorem via Menger's theorem. For $p \in nodes(g_a)$ and $q \in nodes(g_b)$ with $g_a, g_b \in members(g^*)$ and $g_a \neq g_b$. We claim that there are $\overline{\Delta}$ node-disjoint paths between the two sets of terminals T_{g_a} and T_{g_b}. By Lemma 5 there are α node-disjoint paths between T_{g_a} and T_{g_b} and two subsets (node disjoint to nodes included in α paths) $T'_{g^*}, T''_{g^*} \subseteq T_{g^*}$ with $|T'_{g^*}| = |T''_{g^*}| = \beta$, $|T'_{g^*} \cap T''_{g^*}| = 0$ and $\beta \leq \frac{\overline{\Delta}}{2}$, $\alpha + \beta = \overline{\Delta}$. Hence, we only have to show that there are β external node-disjoint connections between T'_{g^*} and T''_{g^*} outside of g^* as well. Since from Lemma 6 there are $\frac{\overline{\Delta}}{2}$ node-disjoint paths between two subsets the claim follows. Applying Menger's theorem, the $\overline{\Delta}$-node-connectivity follows. □

The lemmas, theorems and proof for $\overline{\Delta}$ is odd are analogical and omitted for brevity.

An extended group g_i^* is constructed as follow:

1) Each internal member of g_i^* has a connection to each terminal member. Since there are $\overline{\Delta}$ terminal member each internal member has $\overline{\Delta}$ connections.
2) Each terminal member of g_i^* has one external connection, $I_{g_i^*}$ connections to internal members and $\overline{\Delta} - 1 - I_{g_i^*}$ connections to other terminal members if $\overline{\Delta}$ is even. If $\overline{\Delta}$ is odd at most one terminal member has $\overline{\Delta} - 2 - I_{g_i^*}$ connections instead of $\overline{\Delta} - 1 - I_{g_i^*}$ to the other terminal members. The sum of the connection weights of the $\lfloor \frac{\overline{\Delta} \cdot (\overline{\Delta} - 1 - I_{g_i^*})}{2} \rfloor$ connections between terminal members is minimal over any other selection.

A. Algorithm

If a new node appears it is integrated into the nearest 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 $members(g_j^*) = (members(g_j) \backslash \{g_i\}) \cup (members(g_i) \backslash \{p\})$. The group g_i is removed. If $p \notin T_{g_j}$ then $T_{g_j^*} = T_{g_j}$ and 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 built a new connection to the new terminal node q (see figure 4 (c)).

If extended groups, groups and nodes are merged together and the number of members of the new extended group would be $members(g_j^*) \geq \overline{\Delta} + \overline{\Delta} - 1$, a new regular group with $\overline{\Delta}$ members and minimal weight is built and integrated as single member $members(g_j^*) = members(g_j^*) - \overline{\Delta} + 1$ into the new extended group.

B. Advantages

In the case where $\overline{\Delta}$ members of a merged extended group are hived off, the optimization is done in inverse direction, from root to bottom.

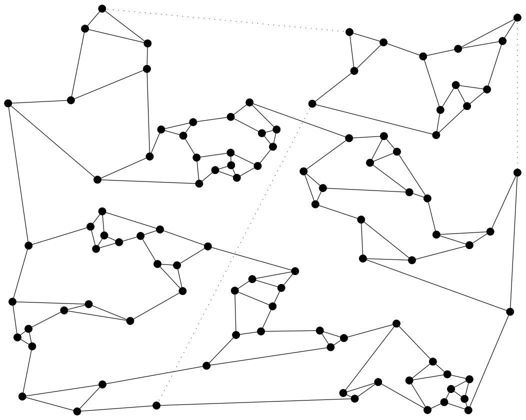
With the extension a node which is $\overline{\Delta}$ or $\overline{\Delta} - 1$-node-connected to the network keeps connected even if topology reconstruction is in progress.

Additionally the extension could be used if the communication graph G has some missing edges and is not fully connected. In that case a node or group with problems in building a new group can join an existing and possibly nearer group by building an extend group.

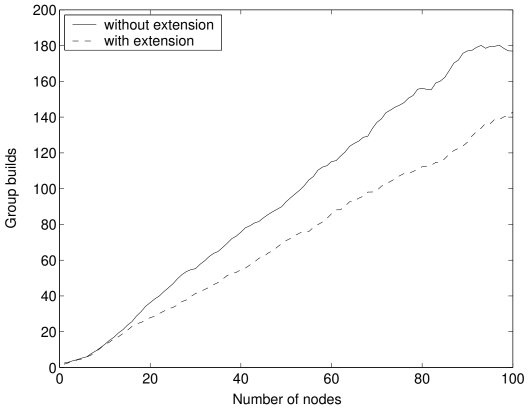
There might be cases where a combination of just regular group until a certain level of the topology tree and extended groups above that level are useful.

The definition of the group weight is a good method for optimizing the overlay graph to specific network requirements. E.g. in [1] a CDMA ad hoc network is suggested where connections of a node should have preferably equal weight and be above a minimum weight (minimum distance between nodes) or if a flat topology tree is required the group weight could include the weight of its members.

VI. Simulation



(a)



(b)

Fig. 5. Simulation results

Figure 5(a) depicts an example graph G' for $n = 100$ and $\overline{\Delta} = 3$. The dotted connections denote the g_{root} internal connections. In Figure 5(b) the average number of group builds in relation to different n for 10 random graphs with and without extension is shown. Since the number of groups for a given n is consistent, the difference between the number of groups and group builds are the groups which where destroyed again since they were not optimal.

VII. Conclusions

We presented a method for constructing overlay graphs in communication networks, which can be proved to be $\overline{\Delta}$-regular, $\overline{\Delta}$-node-connected, and has low weight, for any even $\overline{\Delta}$; in case of odd $\overline{\Delta}$, the overlay graph is $\overline{\Delta} - 1$ node-degree bounded and $\overline{\Delta} - 1$-node-connected. A distributed algorithm for constructing this overlay graph in a fully decentralized fashion was also given, including some simulation results that reveal its high performance. Our approach is hence well-suited for building up fault-tolerant communication topologies for CDMA-based ad hoc networks and peer-to-peer systems.

Future work will focus on additional theoretical results and a more elaborate performance evaluation using analytic modeling and simulation. There are also several variants and other applications of our algorithm, which will be investigated in detail.

References

[1] Bernd Thallner and Günther Gridling, "Simulation of a wireless CDMA ad hoc network," in *Proc. IASTED International Conference on Communications and Computer Networks (CCN'02)*, Boston, Massachusetts, USA, Nov. 4–6, 2002, pp. 354–359.

[2] H. Zhang and A. Arora, "GS^3: Scalable self-configuration and self-healing in wireless networks," in *21st ACM Symposium on Principles of Distributed Computing*, July 2002.

[3] Brent N. Clark, Charles J. Colbourn, and David S. Johnson, "Unit disk graphs," *Discrete Mathematics*, vol. 86, no. 1-3, pp. 165–177, 1990.

[4] Khaled Alzoubi, Xiang-Yang Li, Yu Wang, Peng-Jun Wan, and Ophir Frieder, "Geometric spanners for wireless ad hoc networks," *Proceedings of the IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 5, May 2003.

[5] Khaled Alzoubi, Peng-Jun Wan, and Ophir Frieder, "New distributed algorithm for connected dominating set in wireless ad hoc networks," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, 2002.

[6] Prosenjit K. Bose, Joachim Gudmundsson, and Michiel Smid, "Constructing plane spanners of bounded degree and low weight," in *Processings of the 10th European Symposium on Algorithms*, September 2002.

[7] Gautam Das, "The visibility graph contains a bounded-degree spanner," in *Proceedings of the 9th Canadian Conference on Computational Geometry*, August 1997.

[8] Bevan Das and Vaduvur Bharghavan, “Routing in ad-hoc networks using minimum connected dominating sets,” in *IEEE International Conference on Communications*, 1997.

[9] Jie Gao, Leonidas J. Guibas, John Hershberger, Li Zhang, and An Zhu, “Geometric spanner for routing in mobile networks,” *Proceedings of the second ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 45–55, October 2001.

[10] Samir Khuller, Balaji Raghavachari, and Neal Young, “Low-degree spanning trees of small weight,” *SIAM Journal on Computing*, vol. 25, no. 2, pp. 355–368, February 1996.

[11] Xiang-Yang Li, Gruia Calinescu, and Peng-Jun Wan, “Distributed construction of a planar spanner and routing for ad hoc wireless networks,” in *Proceedings of the 21th Annual Joint Conference of the IEEE Computer and Communications Societies*, June 2002.

[12] Yu Wang and Xiang-Yang Li, “Distributed spanner with bounded degree for wireless ad hoc networks,” in *International Parallel and Distributed Processing Symposium*, April 2002.

[13] Christos Levcopoulos, Giri Narasimhan, and Michiel H. M. Smid, “Improved algorithms for constructing fault-tolerant spanners,” *Algorithmica*, vol. 32, no. 1, pp. 144–156, 2002.

[14] Laurent Massoulie, Anne-Marie Kermarrec, and Ayalvadi Ganesh, “Network awareness and failure resilience in self-organising overlay networks,” in *Processings of the 22nd Symposium on Reliable Distributed Systems*, October 2003.

[15] Donald E. Knuth, *The Art of Computer Programming*, Addison Wesley, 1997.

[16] H. Whitney, “Congruent graphs and the connectivity of graphs,” *Journal of Mathematics*, vol. 54, pp. 150–168, 1932.

[17] Martin Hutle, “An efficient failure detector for sparsely connected networks,” in *Processings of the International Conference on Parallel and Distreibuted Computing and Networks*, February 2004.

[18] Tushar Deepak Chandra and Sam Toueg, “Unreliable failure detectors for reliable distributed systems,” *Journal of the ACM*, vol. 43, no. 2, pp. 225–267, March 1996.