Fault-tolerant Base Station Planning of Wireless Mesh Networks in Dynamic Industrial Environments

Svilen Ivanov
rt-solutions.de GmbH
Oberlnder Ufer 190a
50968 Cologne, Germany
ivanov@rt-solutions.de

Edgar Nett
University of Magdeburg
Institute for Distributed Systems
Universittsplatz 2, 39106 Magdeburg, Germany
edgar.nett@ovgu.de

Ralf Schumann
Economical University of Applied Sciences
Competence Center Information Security
Hauptstrasse 2, 51465 Bergisch Gladbach, Germany
ralf.schumann@fhdw.de

Abstract

Wireless Mesh Networks are infrastructure networks with a wireless multi-hop backbone. In this paper we present an innovative algorithm for fault-tolerant base station planning in wireless mesh networks. The algorithm determines a close-to-minimal number and the positions of base stations to be installed such that the radio coverage is correct in the presence of faults (base station crash or link outage). The algorithm considers both the connectivity of the multi-hop backbone and the connectivity of the mobile stations. The presented algorithm produces correct results, in limited number of iterations under realistic network size and in acceptable time. The provided fault-tolerance is sufficient in most practical situations.

1 Introduction

Wireless networks in automation open possibilities for adaptable manufacturing processes. Wireless mesh networks are a promising technology, since they do not require a wired backbone. However, some problems hinder their usage in automation scenarios. They are mostly related to non-functional communication properties [17, 22]. One of the basic issues that needs to be solved is to guarantee *radio coverage*, which includes last-mile and backbone coverage (section 2.1).

In our previous paper [11] we have developed a general framework for achieving high availability of radio coverage under the presence of faults (link failure and base station failure). It includes online detection of faults, leading to degradation of radio coverage, and network reconfiguration for restoring the normal state (redundancy) of radio coverage. For the online detection of faults we use our previously developed methods [10]. In this paper we focus on the network deployment and network reconfiguration aspect. In particular, we present a new algorithm for base station planning, which finds an optimal number and placement of base stations, leading to redundant radio coverage. Our algorithm extends established methods from infrastructure wireless networks to base station planning of wireless mesh networks. The innovation is that the algorithm considers a different network topology (wireless mesh networks) and provides a network configuration that has correct radio coverage in the presence of faults.

The paper is organized as follows: section 2 provides an overview of radio coverage in wireless mesh networks and defines the problem of base station planning. Section 3 provides related work and motivates the need of a new algorithm. In section 4 we present the main contribution of this paper: an algorithm for base station planning. In section 5 we provide evaluation of the algorithm and section 6 provides conclusion and future work.

2 Context and problem definition

2.1 Wireless mesh networks and radio coverage

Wireless Mesh Networks (WMN) are infrastructure networks with a multi-hop wireless backbone (see example on figure 1). The physical structure is similar to that of the classic infrastructure wireless networks (e.g. IEEE 802.11). Common is that it includes base stations (called also mesh points, or mesh routers), mobile stations, and a distribution system. Similar to infrastructure networks, the base stations are static wireless nodes, providing wireless network connectivity to the mobile stations. The mobile stations communicate to other stations

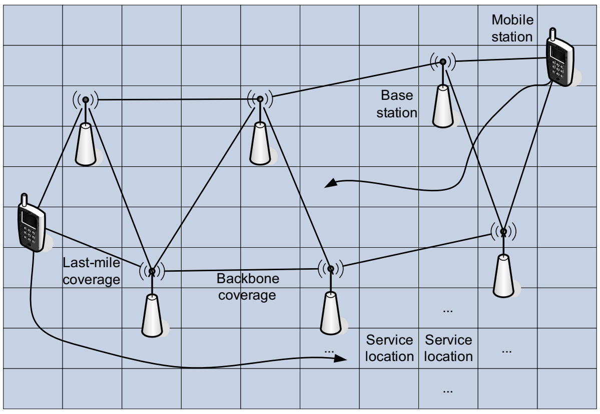

Figure 1. Wireless mesh networks and radio coverage

via the base stations. The distribution system (backbone) provides connectivity among the base stations. The first major difference is the backbone: in classic infrastructure networks it is wired and in WMN it is wireless. Since not all base stations can communicate directly, the backbone is organized as a multi-hop ad-hoc network. This is the second major difference: the logical structure of the network is ad-hoc, meaning that every station can directly communicate with other reachable stations. So, the mobile stations connect to the wireless network through multiple communication *links*. A link exists when two wireless devices can communicate through the wireless medium with some quality parameters. As the mobile stations move, they gradually obtain links to new base stations and loose some of the links to base stations out of reach. In this way, the mobile stations remain always connected to the network and do not perform roaming as in the classic infrastructure networks. The base stations perform routing. *Provided that correct backbone coverage exists*, they automatically discover links to neighbors and the network topology, find routes to the destination, and re-route the traffic in case of base stations failure or link failure.

Radio coverage is a basic *service* of a wireless network. In WMN radio coverage includes two aspects: *last mile coverage* and *backbone coverage.* Last mile coverage provides network connectivity to the mobile stations, which move within some pre-defined *service area*. In particular, last mile coverage guarantees that on all potential *service locations* within the service area, a mobile station can have a link to at least one base station. Last mile coverage is equivalent to radio coverage in classic infrastructure networks, which has been extensively addressed in the literature, e.g [6, 7]. On the other hand, backbone coverage in WMN provides connectivity among the base stations within the distribution system. Backbone coverage guarantees that a path (sequence of links) among every two base stations exists. In classic infrastructure networks the backbone connectivity is not an issue: it is given by definition by the wired network. In WMN this is not the case. The backbone coverage is a result from the base stations placement, the propagation environment, and routing, therefore it requires new methods to be developed.

For computing the radio coverage we use the log-normal shadowing radio propagation model [18] (equation 1).

$$P(d) = P(d_0) - 10n\log_{10}(\frac{d}{d_0}) + X_\sigma \qquad (1)$$

This model calculates the received signal strength $P(d)$ as a sum of a distance-dependent mean and a shadowing factor X_σ. The path loss exponent n determines the rate of signal strength decrease with distance. X_σ is a zero-mean normally distributed random variable with standard deviation σ. X_σ is a statistical way to model the differences in the average signal strength, which occur over large amount of transmitter-receiver separations with the same distance but different obstacles along the path. In this way the parameter σ models the inhomogeneity of the propagation environment.

The advantage of this model is a low modelling effort, which is especially important in industrial automation scenarios. We compensate the lower inaccuracy of this model by automatic model calibration, pessimistic assessment of the radio coverage and by modelling environments of different type. In this way we have shown that this model can be used for reliable assessment of the radio coverage in dynamic industrial environments [11, 10, 21]. For determining the connectivity we use a threshold P_{min}: when the received power in both directions exceeds P_{min}, then connectivity is available (e.g. a base station covers a service location, or a link between two base stations is possible).

2.2 Problem definition

The following input information is given to the base station planning algorithm:

- Service location information. This is information about possible service locations, which have to be covered.
- Candidate sites information. This is information about possible locations of base stations.
- Radio connectivity information:
  - Last mile information: for every service location, the candidate sites, which cover this service location, if base stations were installed at all candidate sites,
  - Backbone information: for every candidate site, the candidate sites, which have a link in the backbone network, if base stations were installed at all candidate sites.

The radio connectivity information is given a form of radio connectivity model, which allows to predict the radio connectivity for any service locations and candidate sites.

- Optional: currently installed base stations and their positions

The base station planning algorithm has to determine the number and positions of base stations to be installed such that:

- The radio coverage (last mile and backbone) is correct
- The radio coverage is fault tolerant, i.e. it remains correct in the presence of faults. In this paper we consider the *minimum fault tolerance*, which means that the radio coverage remains correct in the case that one base station fails or one link fails.
- The number of base stations is as low as possible and the running time of the algorithm is reasonable

The fault-tolerance requirement can be formally represented as:

- Last mile coverage requirement: every service location is covered by at least two base stations. This tolerates either the failure of one base station or the failure of a link.
- Backbone coverage requirement: we model the multi-hop backbone network as a graph where the base stations are vertices and the links are edges. The requirement is that the backbone graph is *biconnected* (2-connected). This means that there is no graph element (vertex or edge), whose removal disconnects the graph (follows from [8], proposition 1.4.2).

The challenge of the defined problem is the backbone coverage requirement. In last mile coverage, the requirement can be formally defined as a local property, which depends only on the considered entities (e.g. a base station covers a service location). In backbone coverage, the requirement is global. It includes all network paths among all pair of base stations. The existence of a path between two base stations depends not only on the considered base stations, but on the number and positions of all other base stations in the network. The fault-tolerance (biconnectivity) requirement and the minimality requirement increase the complexity of the problem. The contribution of this paper is an algorithm for fault-tolerant base station planning in wireless mesh networks, that satisfies these requirements.

3 Related work

Industrial automation networks have typically been isolated single-cell networks or classic infrastructure networks with multiple cells. This means that coverage planning is required only for the 'last mile', i.e. the connection between a base station and a mobile station, e.g. [7]. In the case of multi-hop wireless mesh networks, the coverage planning of the backbone network is a new aspect that needs to be considered. Another aspect is that most papers on radio network planning consider network throughput as a main planning goal, e.g. [6]. However, the most common requirement of industrial networks is availability. With the introduction of technologies for multi-hop communication in industrial environments (e.g. Zigbee, Wireless HART), the base station planning problem gets importance. Paper [19], for instance, presents the challenges for developing a planning tool for industrial wireless sensor networks and identifies that base station planning is an important requirement. However, to the best of our knowledge, no systematic approach exists for planning multi-hop wireless networks with respect to fault-tolerance requirements of industrial automation networks.

The existing algorithms for base station planning in wireless mesh networks [2, 20] have a different goal. It is to design a mesh network with minimum number of base stations such that the end-to-end throughput requirements of application flows are fulfilled. These requirements are typical for Internet access in areas with no alternative high-speed wired connection. The approach is to transform the planning problem into a linear optimization problem, which is a combination of a set covering problem and a network flow problem. As a result the backbone is a connected graph, but there is no fault-tolerance. Additional disadvantage is the intractability of the proposed approaches. For some inputs the algorithm takes too much time for the result to be useful. This is because the underlying linear optimization problem is a binary integer problem, which is well known for its NP-completeness. Paper [20] addresses this issue by a decomposition method, but still the algorithm takes about 22 hours for a scenario with 58 nodes. This is acceptable for the mentioned scenarios, but for network reconfiguration in automation scenarios a faster algorithm is required. Extending these algorithms to fault-tolerance would mean an additional increase in the complexity. Paper [13] addresses the problem of fault-tolerant deployment of wireless ad-hoc networks. The authors present a method for determining the probability that a backbone network graph is k-connected, based on the transmission range. However, a basic assumption of the method is that the network can be modeled as a union disk graph, where all nodes within a given transmission range are perfectly reachable and all nodes outside this range are not reachable at all. It has been shown that this network model in the general case does not comply with real networks [12].

Many papers address the problem of fault-tolerant communication in mobile multi-hop ad-hoc networks (MANET) and in wireless sensors networks. Papers considering fault-tolerant routing, for instance [9, 3], have as a prerequisite at least biconnected backbone network, but the base station planning problem is out of their scope. Some scientific works consider the topological properties of the networks, for instance biconnectivity testing and topology control. [15] proposes a distributed algorithm for testing a given wireless multi-hop network for biconnectivity under uncertainty caused by message losses. Topology management algorithms (e.g. [5, 4]) determine the sleep transitions and transmit power levels of the nodes, such that the network topology is connected and the total power consumption is minimum. Node placement algorithms generate multi-hop network topologies for the purpose of simulation. The goal of these algorithms is to resemble the real network properties as much as possible [16] and they generate topologies which are not biconnected.

However, the base station planning problem has been little addressed in the MANET and sensors networks research domains. This is because in these scenarios the number and position of the nodes is considered uncontrolled or hardly controlled: the nodes are typically autonomous or randomly deployed. In automation scenarios, however, the networks are typically planned to provide service in some predefined geographical area (e.g. production hall). This requires careful base station planning for ensuring high availability of the radio coverage. Our approach is to extend the existing methods from infrastructure network planning to planning multi-hop wireless mesh networks with fault-tolerance aspects. The proposed base station planning algorithm provides coverage in a predefined area by adjusting the number and the position of the base stations.

4 Automatic base station planning algorithm

4.1 Overview of the algorithm

Our algorithm performs an optimization, fulfilling a simple local network property, which significantly affects the fulfillment of the global property (biconnectivity). This local property is the *minimum degree.* For a backbone multi-hop network, the *degree* of a base station is the number of links to other base stations. Minimum degree of the network is the least degree among all base stations. In graph theory, the minimum degree is a *necessary but not sufficient condition* for k-connectivity [8]. This means that a k-connected graph has a minimum degree of k, but a graph with minimum degree of k is not necessary k-connected. Formally, this rule applies to the backbone of wireless mesh networks. However, radio coverage in WMN has two aspects: backbone and last mile. Because of the last mile fault-tolerance requirement, every service location is covered by at least two base stations. Since the service locations are typically spread in some area (e.g. production hall), the probability that the necessary condition is also sufficient is significantly higher than in pure graph theory. Therefore, our algorithm fulfills the local necessary condition and checks whether the global sufficient condition is also fulfilled. If not, the algorithm performs an incremental correction. The advantage of this approach is that it fulfills the backbone coverage requirement without increasing the complexity of the underlying optimization problem.

The algorithm operates in three steps: optimization, connectivity testing, and graph consolidation (figure 2).

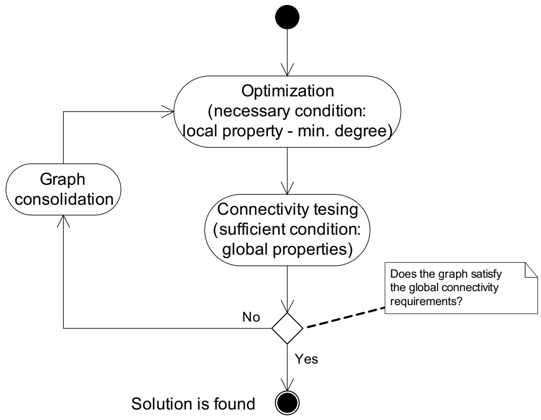

Figure 2. Base station planning algorithm

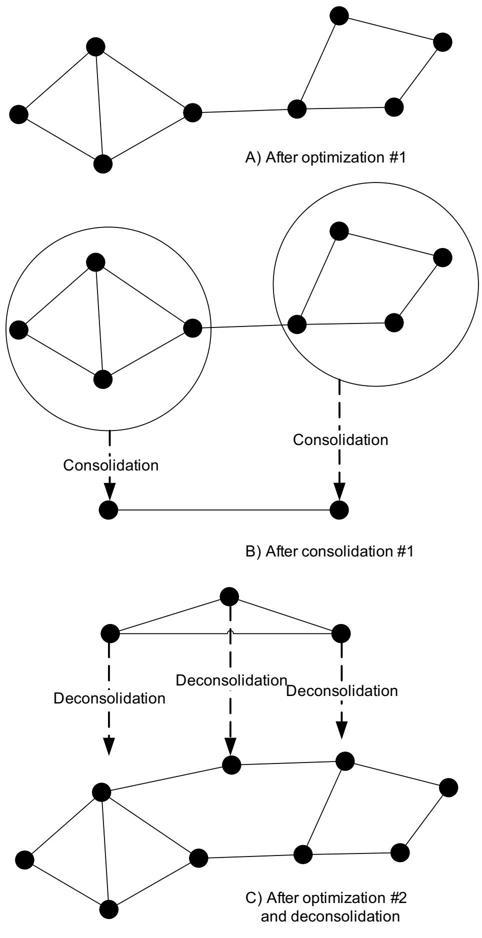

Figure 3. Example operation of the base station planning algorithm

The optimization step finds an optimal solution for the last mile coverage requirement and the *necessary condition for backbone coverage* (the local property min. degree). The connectivity testing step tests the resulted graph for biconnectivity (the sufficient condition). If the sufficient condition is true, the algorithm finishes. Otherwise the algorithm performs a graph consolidation step. The consolidation step maps parts of the graph, which are biconnected, to a single vertex. After the consolidation, the algorithm continues with the optimization step, which is done based on the consolidated graph. After few iterations the algorithm produces a solution that satisfies the coverage requirements. As we have seen in section 3, finding a globally optimal solution to the base station planning algorithm is intractable. For this reason our algorithm performs optimization based on the minimum degree, which can be solved in reasonable time and produces a biconnected graph with high probability. The incremental correction adds additional base station(s) if the graph is not biconnected. For this reason our algorithm finds a close-to-optimal solution.

Let the optimization step has produced a graph with minimum degree 2 (figure 3A) according to the necessary condition. This graph does not satisfy the biconnectivity requirements (one edge and two vertices exist whose removal disconnect the graph). The consolidation step identifies two subgraphs, which are biconnected, and maps them to vertices (figure 3B). Note that after the consolidation the minimum degree of the graph is 1. Then the optimization step places a new base station, such that the consolidated graph plus the new vertex result in a graph with minimum degree of 2 (figure 3C). Finally, the deconsolidated graph satisfies the biconnectivity requirements.

The idea of using minimum degree for achieving global connectivity has been used, for instance in algorithms for topology control based on transmission power [5]. These methods result in connectivity with *high probability* but *give no guarantees*. In contrast, our algorithm *guarantees* the biconnectivity of the resulting network graph. This is achieved by the graph consolidation and the iterative optimization.

4.2 Optimization

Minimization approach Our algorithm uses a minimization approach, based on binary search, for finding the minimum number of base stations (BS_{min}), which satisfy the coverage requirements. It searches iteratively the interval between a lower bound BS_{low} and an upper bound BS_{up}. At each iteration the algorithm chooses the middle of the interval as current value for BS and determines whether a solution is possible by solving an optimization problem. If the solution satisfies the coverage requirements, the algorithm decreases BS by searching the lower half of the interval, otherwise it increases BS by searching the upper half of the interval. Finally, the algorithm finds a minimum value for BS, which satisfies the coverage requirements.

Optimization problem formulation The optimization performed at each iteration can be defined by the following:

- Variables

  The optimization variables are the positions of the base stations $(X, Y, Z)_{BS}$. We consider a typical multi-hop network, operating in a single frequency, therefore the frequency assignment is a constant for all base stations.

- Bounds

  The variables have lower and upper bounds according to the candidate sites information, provided by the user. For instance, if the base stations are to be installed on the ceiling of a production hall with dimensions 200x300x6m, then the bounds are: $0 \leq X \leq 200$, $0 \leq Y \leq 300$,$Z = 6$. For the currently installed base stations, the lower and upper bounds are equal to the base stations' coordinates. In this way they are considered in the coverage, but they are not relocated by the algorithm.

- Service locations

  The service locations, represented by their coordinates, are stored in the set SL.

- Radio connectivity model

  From the values of the variables $(X, Y, Z)_{BS}$ the radio connectivity model provides the following information:

  - Last mile coverage: $RadioModel.LastMile((X, Y, Z)_{BS})$. This is a vector, which for every service location in the set SL contains the number of base stations that cover this service location.
  - Backbone coverage: $RadioModel.BSDegree((X, Y, Z)_{BS})$. This is a vector, which for every base station contains the number of links to other base stations.

- Objective function

  The objective function (Matlab pseudo code in algorithm 1) maximizes the average last mile coverage and the average degree in the backbone and influences the solution in a direction, which satisfies the coverage requirements. From the input coordinates and the radio connectivity model the function calculates the last mile coverage and the backbone coverage. For base stations, which have less than $N_{bb} = 2$ links to other base stations, the function calculates the backbone shortfall. This is the sum of the differences between the required and the current degree over all base stations. The shortfall is weighted by a backbone penalty factor and subtracted by the objective function. The penalty factor is a relatively large

Algorithm 1 Objective function of the optimization step

$function\,Objective(X, Y, Z)$
{
$PenaltyLM = 50;$
$PenaltyBB = 100;$
$LMCoverage = RadioModel.LastMile(X, Y, Z);$
$LMShortfall = sum(Nlm - LMCoverage(find(LMCoverage < Nlm)));$
$BBCoverage = RadioModel.BSDegree(X, Y, Z);$
$BBShortfall = sum(Nbb - BBCoverage(find(BBCoverage < Nbb)));$
$Objective = mean(LMCoverage) + mean(BBCoverage) - PenaltyLM * LMShortfall - PenaltyBB * BBShortfall;$
}

number, compared to the mean values, which influences the solution in a direction with a zero shortfall. The behavior for the last mile coverage links is similar. The objective function should be maximized.

Optimization problem solving In order to solve this optimization problem we apply an optimization method. Special for this problem is that the objective function can not be differentiated. This is because it contains the radio connectivity model, which can not be represented as an algebraic function of the base stations' coordinates. Several algorithms exist for solving this type of problems (pattern search, genetic algorithm, simulated annealing). We select pattern search because it has a proven convergence and supports any type of constraints [14].

4.3 Connectivity testing

For graph k-connectivity testing in a graph with n vertices we use classic algorithms from the graph theory [8] with complexity of $O(k * n^3)$. In the context of this paper the usage of classic algorithms for k-connectivity testing is acceptable, since the base station planning and the connectivity testing step are performed at a central instance based on global information about the graph. If in some other context the base station planning is performed in a distributed way, distributed connectivity testing algorithms can be used [15].

4.4 Graph consolidation

In this step the algorithm finds subgraphs, satisfying the connectivity requirements, and transforms each subgraph into a single vertex.

The formal specification of the graph consolidation step is described in pseudo code in algorithm 2, which is explained as follows:

1. Given a graph G, identify all biconnected components G_c containing at least 3 vertices and store them in a set BC.

Algorithm 2 Graph consolidation algorithm

$BC = biconnected_components(G, |G_c| \geq 3)$
$V_{sap} = articulation.points(G, among(G_c \in BC))$
for all $v \in V_{sap} \cup (V(G) - V(BC))$ **do**
  $v \rightarrow v^{'}$
  $E(v^{'}) = E(v)$
end for
for all $G_c \in BC$ **do**
  $G_c = G_c - V_{sap}$
  $G_c \rightarrow v^{'}$
  $E(v^{'}) = ExternalEdges(G_c)$
  $remove.duplicate.edges(v^{'})$
end for

2. Identify the *special articulation points*, which are articulation points, shared between the biconnected components in the set BC. Articulation point is a vertex whose removal disconnects a graph. For identifying biconnected components and articulation points, classic graph search algorithms are used [8].
3. Every vertex, which is either a special articulation point or other vertex, not belonging to a biconnected component in BC, is directly transformed into a vertex in the consolidated graph. The consolidated vertex inherits all edges of the original vertex.
4. For every biconnected component in the set BC:
   (a) If it contains special articulation points, they are removed from the component.
   (b) All vertices from the component are transformed into a single vertex in the consolidated graph.
   (c) The consolidated vertex inherits all edges of the original vertices to other vertices in the graph. Other vertices are vertices not belonging to the same biconnected component.
   (d) Duplicated edges in the consolidated graph are removed.

5 Experimental evaluation

5.1 Evaluation approach

We evaluate the algorithm according to the following important evaluation criteria:

- Fault-tolerance: this shows the algorithm's ability to generate a network configuration that satisfies the fault-tolerance coverage requirements.
- Termination: this shows the number of iterations the algorithm needs to complete and the running time.
- Minimality: this shows the ability of the algorithm to use minimum number of base stations.

Parameter	Values
Transmit power P_{tx} [dBm]	20
Required receive power P_{min} [dBm]	-78
Path loss exponent	3
Area size (X/Y) [meters]	(50/50),(100/100) ,...,(300/300)
Shadowing deviation σ [dB]	5,6,7,8,9,10

Table 1. Evaluation parameters

The base station planning algorithm has been implemented in Matlab (about 600 lines of code). We evaluate the algorithm in various scenarios, typical for industrial automation (see table 1 for the parameter values). As input to the algorithm we use a production hall with various size, where the service locations comprise the whole floor and the candidate sites comprise the whole ceiling. We vary the homogenity of the propagation environment by modifying the shadowing deviation. The algorithm has been tested on all combinations of input parameters (area size and shadowing deviation), which make a total of 36 executions. At the end of each algorithm execution, we perform a requirements test. We test the last mile coverage by using the radio connectivity model, we test the backbone coverage by performing a biconnectivity test of the backbone graph (using the MatlabBGL graph library).

5.2 Fault-tolerance

With all inputs the algorithm has generated a network topology, which is fault-tolerant with respect to the requirements defined in section 2.2: the radio coverage remains correct in the case that one base station fails or one link fails. An example graph of the network topology, generated by the algorithm for area size 200/200m and shadowing deviation 8 is shown on figure 4. The related work algorithms [2, 20] generate topologies, which are not fault-tolerant. Their topologies optimize the network throughput, but the backbone network is not a biconnected graph, meaning that base stations and links exist, whose removal disconnect the graph (see figure 3 in [2], and figure 4 in [20]). Figure 4 clearly shows the effect of the shadowing (inhomogeneous environment) on the base station planning. Because of the shadowing some links are shorter than others and in some areas more base stations are needed to provide coverage.

5.3 Termination, running time and minimality

Figure 5 shows the measured cumulative termination of the algorithm within the performed evaluation. Cumulative termination is the percentage of algorithm executions that have terminated *up to* some number of iterations. In 30% of the algorithm executions the algorithm generates a correct fault-tolerant solution directly after the first iteration. In these cases the graph consolidation step was not performed at all. These were the cases with a smaller area sizes: 50/50m and 100/100m. In 80% of the algorithm executions the algorithm generates a correct fault-tolerant topology after the second iteration, which means that only two optimizations and one graph consolidation was needed. The algorithm needed maximum four iterations to complete for all inputs.

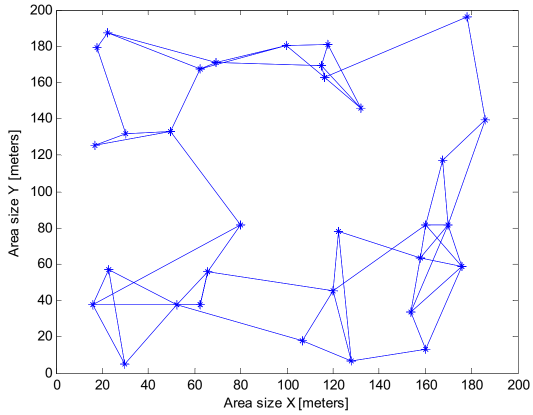

Figure 4. Example fault-tolerant (biconnected) topology produced by the algorithm

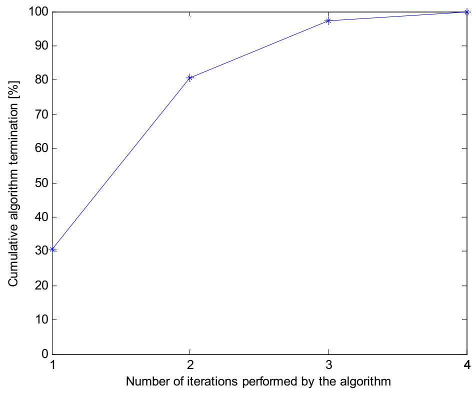

Figure 5. Algorithm termination: 80% from all algorithm executions terminated after 2 iterations. The algorithm needed maximum 4 iterations to complete.

On average over all algorithm executions, 90% of the base stations were selected at the first algorithm iteration. This means that these 90% were selected according to the global optimization function and are optimally placed. Only 10% of the base stations were selected during the incremental correction (subsequent algorithm iterations) in order to ensure the biconnectivity of the backbone. For small area size (50x50) we have evaluated a global optimization algorithm based on integer programming like the related work algorithms [2, 20]. For larger area size this algorith did not complete within 12 hours and we have terminated it. For the small area size our algorithm produced a configuration with 4 base stations, which was equal to the number of base stations used by the globa optimization algorithm. For these reasons we conclude that our algorithm produces a close-to-optimal solution.

For the total 36 executions the algorithm needed about 25 minutes to complete on a laptop with dual core processor at 2.5GHz and 3GB operating memory. This means that for one specific scenario of realistic size the algorithm needs about a minute to complete, which is acceptable time.

6 Conclusions and outlook

The presented algorithm for fault-tolerant base station planning in wireless mesh networks produces correct results, in limited number of iterations under realistic network size. It provides a minimum fault-tolerance (failure of base station or link failure), which is sufficient in most practical situations.

In future work we will evaluate the effect of the algorithm on our generic method for high availability of the radio coverage [11]. The purpose is to evaluate whether in the case of degraded radio coverage in some real environment, the proposed reconfiguration actions from the base station planning algorithm restore the normal (redundant) state of the radio coverage. Further work will be to provide a formal proof of the termination property of the algorithm. In addition, we will extend the algorithm for a configurable amount of fault tolerance, i.e. the radio coverage remains correct if k base stations fail *and/or* l links fail.

7 Acknowledgement

This work has been partially done in the EU-project flexWARE [1], grant number 224350.

References

[1] flexWARE - Flexible Wireless Automation in Real-Time Environments. Research project funded under the EU-FP7 programme. http://www.flexware.at/.

[2] E. Amaldi, A. Capone, M. Cesana, I. Filippini, and F. Malucelli. Optimization models and methods for planning wireless mesh networks. *Elsevier Journal on Computer Networks*, 52(11):2159–2171, 2008.

[3] D. Avresky and N. Natchev. Dynamic reconfiguration in computer clusters with irregular topologies in the presence of multiple node and link failures. *IEEE Transactions on Computers*, 54(5):603–615, 2005.

[4] L. L. Bello and E. Toscano. An adaptive approach to topology management in large and dense real-time wireless sensor networks. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, 5(3):314–324, 2009.

[5] D. M. Blough, M. Leoncini, G. Resta, and P. Santi. The k-neighbors approach to interference bounded and symmetric topology control in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(9):1267–1282, 2006.

[6] S. Bosio, A. Capone, and M. Cesana. Radio planning of wireless local area networks. *IEEE/ACM Transactions on Networking*, 15(6):1414–1427, 2007.

[7] T. M. Chan, K. F. Man, K. S. Tang, and S. Kwong. A jumping-genes paradigm for optimizing factory wlan network. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*, 3:33–43, 2007.

[8] R. Diestel. *Graph Theory*. Springer-Verlag, Heidelberg, 2005.

[9] C.-Y. Hsu, J.-L. C. Wu, S.-T. Wang, and C.-Y. Hong. Survivable and delay-guaranteed backbone wireless mesh network design. *Journal of Parallel and Distributed Computing*, 68(3):306 – 320, 2008. Wireless Mesh Networks: Behavior, Artifacts and Solutions.

[10] S. Ivanov and E. Nett. Fault-tolerant coverage planning in wireless networks. In *27th IEEE International Symposium on Reliable Distributed Systems (SRDS 2008), 6-8 October, Napoli, Italy*, 2008.

[11] S. Ivanov, E. Nett, and S. Schemmer. Planning available WLAN in dynamic production environments. In *7th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems*, 2007.

[12] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott. Experimental evaluation of wireless simulation assumptions. In *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 78–82, New York, NY, USA, 2004. ACM Press.

[13] X.-Y. Li, P.-J. Wan, Y. Wang, and C.-W. Yi. Fault tolerant deployment and topology control in wireless networks. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 117–128, New York, NY, USA, 2003. ACM.

[14] MathWorks. Matlab Global Optimization Toolbox – User's Guide, March 2010.

[15] B. Milic. *Distributed Biconnectivity Testing in Wireless Multi-hop Networks*. PhD thesis, Humboldt University – Berlin, 2009.

[16] B. Milic and M. Malek. Npart - node placement algorithm for realistic topologies in wireless multihop network simulation. In *Simutools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, pages 1–10, ICST, Brussels, Belgium, Belgium, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[17] E. Nett. Wlan in automation - more than an academic exercise? In *Dependable Computing - Second Latin-American Symposium, LADC 2005, Salvador, Brazil*, 2005.

[18] T. S. Rappaport. *Wireless Communications - Principles and Practice*. Prentice Hall PTR, 2002.

[19] A. Ray. Planning and analysis tool for large scale deployment of wireless sensor network. *International Journal of Next-Generation Networks (IJNGN)*, 1(1):29–36, December 2009.

[20] A. So and B. Liang. Optimal placement and channel assignment of relay stations in heterogeneous wireless mesh networks by modified bender's decomposition. *Elsevier Journal on Ad Hoc Networks*, 7(1):118–135, 2009.

[21] J. Weczerek and S. Ivanov. Einfaches planen von drahtlosen automatisierungsnetzwerken. In *10. Wirless Technologies Kongresses, Bochum 22-23.09.2008*, 2008.

[22] A. Willig. Recent and emerging topics in wireless industrial communications: A selection. *IEEE Trans. Industrial Informatics*, 4:102–121, 2008.