On the Feasibility of Distributed Link-Based Channel Assignment in Wireless Mesh Networks

Felix Juraschek Mesut Güneş Matthias Philipp Bastian Blywis
Distributed Embedded Systems - Institute of Computer Science
Freie Universität Berlin and Humboldt University Berlin, Germany
{jurasch, guenes, mphilipp, blywis}@inf.fu-berlin.de

ABSTRACT

In this paper we present an experimental evaluation of the *distributed greedy algorithm* (DGA) for distributed channel assignment in wireless mesh networks. The algorithm has the advantage of preserving the network topology by assigning channels to links instead of interfaces, thus being completely transparent to the routing layer. Our implementation is based on DES-Chan, a framework for the development of distributed channel assignment algorithms. We evaluate the performance in the DES-Testbed, a multi-radio *wireless mesh network* (WMN) with 98 nodes at the Freie Universität Berlin. We present a graph-theoretic analysis of the experiment results and measure the achieved throughput after the channel assignment. We discuss the feasibility of link-based channel assignment and show that the feature of the algorithm of being transparent to the routing layer is not always guaranteed. Additionally, we show the importance of using realistic interference models to fully exploit the performance gain by channel assignment in real network deployments.

Categories and Subject Descriptors: C.2.1 [Network Architecture and Design]: Wireless communication — *Wireless Mesh Networks*; C.2.1 [Network Architecture and Design]: Wireless communication—*Channel Assignment*; C.2.4 [Distributed Systems]: Distributed Channel Assignment

General Terms: Experimental, Performance

Keywords: Wireless Mesh Network, Channel Assignment, Link-Based Channel Assignment

1. INTRODUCTION

Multi-radio mesh routers allow the communication over several wireless network interfaces at the same time. However, this can result in high interference of the wireless transmission leading to a low network performance. Channel assignment for multi-radio *wireless mesh networks* (WMNs) attempts to increase the network performance by decreasing the interference of simultaneous transmissions. The reduction of interference is achieved by exploiting the availability of fully or partially non-overlapping channels. Channel assignment can be applied to all wireless networks based on technologies that provide non-overlapping channels. Currently, wide-spread technologies are IEEE 802.11a/b/g, IEEE 802.11n, and IEEE 802.16 (WiMAX). With the relatively low cost for IEEE 802.11 hardware, the number of deployments based on this technology is increasing and channel assignment algorithms are gaining in importance.

Although channel assignment is still a young research area, many different approaches have already been developed [13]. These approaches can be distinguished into *centralized* and *distributed* algorithms. Centralized algorithms rely on a central entity, usually called *channel assignment server* (CAS), which calculates the network-wide channel assignment and sends the result to the mesh routers. In distributed approaches, each mesh router calculates its channel assignment based on local information. Distributed approaches usually react faster to topology changes due to node failures or mobility and introduce less protocol overhead since communication with the CAS is not necessary. As a result, distributed approaches are more suitable once the network is operational and running.

Another classification considers the frequency of channel switches on a network node. In *fast channel switching* approaches, channel switches may occur frequently, in the extreme for every subsequent packet a different channel is chosen. The limiting factor for dynamic algorithms is the relative long channel switching time with commodity IEEE 802.11 hardware, which is in the order of milli seconds. *Slow channel switching* approaches in contrast, switch the interfaces to a particular channel for a longer period, usually in the order of minutes or hours. *Hybrid* approaches combine both methods.

The contribution of this paper is the experimental evaluation of the *distributed greedy algorithm* (DGA) in a large wireless mesh testbed. The implementation is based on DES-Chan, an open-source framework for distributed channel assignment algorithms in wireless mesh networks. We describe the experimental evaluation in the DES-Testbed with 98 multi-radio IEEE 802.11 mesh routers and compare the results to the original ones [15]. The evaluation has been carried out with a graph-theoretic analysis of the experiment results and measurements of the achieved throughput after the channel assignment. Based on the results, we discuss the feasibility of link-based channel assignment and show that the promising feature of the algorithm to be transparent for routing algorithms, does not hold in reality. The resulting

MobiWac'11, October 31–November 4, 2011, Miami, Florida, USA.

channel assignments show that the network connectivity is not always guaranteed and that dynamic topologies are not supported. Finally, we show that the used interference models have an high impact of the performance gain achievable with channel assignment.

The remainder of this paper is structured as follows. In Section 2 we present recent link-based and interface-based algorithms for distributed channel assignment. Section 3 briefly introduces the DES-Chan framework and Section 4 presents the implementation of the DGA algorithm. The experiments on the DES-Testbed and the results of the evaluation of DGA are described in Section 5. The article concludes with an outlook and future work.

2. RELATED WORK

In this section, we present recent different algorithms for channel assignment. The focus is on distributed algorithms, since they are most suitable for self-organizing wireless mesh networks based on IEEE 802.11. The list of algorithms is not complete, for a more complete overview we refer to the survey in [13].

In the greedy channel assignment algorithm by Ko et. al [7], one wireless interface of each node is switched to a common channel in order to ensure the network connectivity. For additional interfaces, a greedy algorithm selects the least interfering channel in the interference set using an interference cost function which takes the degree of overlap of two channels into account. The interference set comprises all nodes within the 3-hop neighborhood. As an additional constraint, at least one neighbor must have a radio tuned to the selected channel in order to avoid dead interfaces. Channel changes are communicated with a 3-way handshake, in order to avoid channel oscillation.

The *distributed greedy algorithm* (DGA) assigns channels to links instead of interfaces and is therefore topology preserving, meaning that all links are sustained during the channel assignment procedure [15]. The link-based approach has the advantage that it is therefore transparent to the overlaying routing algorithm. A binary interference model, which specifies an interference range of m hops is used. A conflict graph is used to formulate the problem so that the number of edges in the conflict graph shall be minimized. The channel switch is carried out using a 3-way handshake. In order to avoid oscillation, each vertex and channel combination can only be changed once. The algorithm is described in detail in Section 4 when the implementation for DES-Chan is described.

The link-based channel assignment approach by Sridhar et. al is similar to DGA with the difference that it additionally takes the expected traffic-load on the nodes into account [14]. Interference is modeled with a fixed interference range for all nodes and a weighted conflict graph is used to estimate the network-wide interference. The weights for the edges in the conflict graph are specified using a load-matrix that takes the expected traffic for each link into account. The channel assignment problem is then defined as minimizing the sum of the weighted edges of the conflict graph. For every link the owner is specified as the node with the higher cumulative expected traffic, and only this node may assign a channel to this link. The algorithm then works very similar to the previously described approach.

A fast channel switching algorithm is proposed by Kyasanur et. al in [8, 9]. The set of network interfaces on each node are divided into *fixed* interfaces, which stay on a fixed channel, and *switchable* interfaces. If a node wants to communicate with a neighbor, it tunes one of the switchable interfaces to a channel of a fixed interface of the receiving node. The crucial part of this approach is the way how channels are assigned to the fixed interfaces. The authors present two different algorithms for this task. For a simple solution, a well-known function may be used which calculates the channel for a fixed interface based on the node ID. As an alternative, neighborhood information is considered for the channel selection which requires the periodic exchange of topology information messages. Each node then selects the least used channel for its fixed radio.

The *Skeleton Assisted Partition Free* (SAFE) algorithm uses *minimal spanning trees* (MSTs) to preserve the network connectivity [12]. The 2-hop neighborhood is used as interference model. A conflict graph is used and the goal of the algorithm is to minimize the number of edges in the conflict graph.

The channel assignment algorithm consists of two components. A random channel assignment is applied if $C < 2K$, where K is the number of wireless network cards on every node and C the number of non-overlapping channels. Due to the pigeonhole principle, two nodes will share a common channel although they are assigned randomly. The second component of the algorithm introduces the condition that all edges of a MST, the *skeleton*, of the network have to be preserved when $C \geq 2K$. For this, every node randomly chooses a channel set with $K-1$ channels, leaving one interface unassigned. The node broadcasts its chosen channel set, and if links to all skeleton neighbors are already established it assigns a random channel to the unassigned interface. Otherwise it tries to establish links with the not connected skeleton neighbors by assigning a channel which is in the channel set of all skeleton neighbors. If there is such a channel, it is assigned to the interface, if not, a global common channel is used for these links.

Most of the algorithms have been studied in simulations and some have been implemented for small wireless testbeds. The implementations are usually tailored to one particular network, which makes it hard to compare the performance of the different approaches. As next, we introduce DES-Chan, a framework for the development of distributed channel assignment algorithms which aims to close the gap by creating implementations for different approaches.

3. DES-CHAN

The DES-Chan framework has been developed for experimentally-driven research on distributed channel assignment in real network environments. The framework introduces an abstraction layer for operating system specifics and thus enables the researcher to spend most development time on the algorithm logic instead of, for instance, memory management and handling the wireless interfaces. Additionally, DES-Chan provides basic services and data structures that are often required for typical tasks in channel assignment algorithms as elaborated in [6]. DES-Chan is available at the website of the DES-Testbed http://www.des-testbed.net.

3.1 Architecture and components

The architecture and components of DES-Chan are depicted in Figure 1. *DES-Chan Core* is a Python library that provides common functions and data structures for channel assignment algorithms. The framework comprises the following modules for the implementation process of distributed channel assignment algorithms.

- *Interface Management* - The interface management module acts as interface to the operating system and hides testbed-specific characteristics. It provides various functions for configuring network interfaces and retrieving information about their state. Thus it is possible to set up and shut down interfaces, check whether an interface has been set up, and get information of unused interfaces.
- *Neighborhood-Discovery* - The Neighborhood-Discovery module runs as a daemon that determines the links and their quality on each network node based on the ETX link metric [2]. The service provides a network node with its m-hop neighbors, where m can be set by the researcher. For each link, the quality according to the ETX metric and the channel the link is using is provided.
- *Node Communication* - With this module, the researcher can quickly develop the required protocol implementation for exchanging messages among the network nodes, for instance in order to propagate changes in channel assignment or to carry out 3-way handshakes prior to the actual channel switch as proposed in many algorithms to avoid deaf interfaces and ensure a synchronization for channel switches.
- *Network graph* - This component provides data structures for the network graph and the corresponding conflict graph. The edges can be labeled with the number of the channel that is used for the communication.
- *Interference Models* - Currently, the 2-hop interference model is implemented. The implementation supports a binary notion of interference, i.e., two links either interfere or do not interfere, and a more accurate notion of the interference cost taking the spectral distance of two channels into account [7].

3.2 Integration into the DES-Testbed

The 3D-visualization tool DES-Vis of the DES-TestBed Management System (DES-TBMS) [4] has been extended to support a visualization for channel assignment algorithms. This way, a 3D-representation of the channel assignment can be displayed, as shown later in Section 5.

4. DISTRIBUTED GREEDY ALGORITHM

The *distributed greedy algorithm* (DGA) assigns channels to links instead of interfaces and is therefore topology preserving [15]. This has the promising advantage, that the approach is independent of the overlaying routing algorithm by preserving the network topology. A conflict graph is used to formulate the problem so that the number of edges in the conflict graph shall be minimized. In the distributed algorithm a network link between two nodes is owned by the node with the higher node ID and only this node may assign a channel to the link. The node ID can be any unique identifier, such as IP address, MAC address, or host name.

At the network initialization, all links are assigned to the same channel. Each node then iterates over all owned links and changes the channel of the link which results in the largest decrease of interference in the local neighborhood. The largest decrease is achieved with the combination of link u and channel k that removes the highest numbers of edges in the local conflict graph. The interface constraint has to be respected, which means that no more channels can be assigned to a node than it has interfaces. In order to avoid oscillation, each vertex and channel combination can only be changed once.

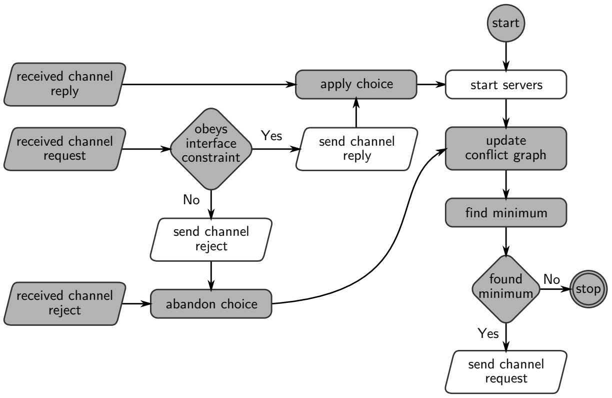


Figure 2: Flowchart of the distributed DGA implementation. Methods of the *DGA* class are colored gray and methods of the *Messaging* class are colored white. Input/Output operations are represented by parallelograms. If a channel request is not answered within a certain time, a channel reject event is generated automatically. The algorithm terminates, when no link-channel combination that further minimizes the interference can be found. Channel update messages are not implemented, because the current channel assignment is retrieved from the DES-Chan *Neighborhood Discovery* module in each iteration.

The channel switch is carried out using a 3-way handshake. This procedure is repeated until the local interference cannot be reduced any further, i.e., all possible (u, k) combinations have been tried. Since each combination is only tried once, the total number of iterations (over all instances of the algorithm) is $O(|V_c|K)$, where $|V_c|$ is the number of vertices in the conflict graph (of the whole network) and K is the number of available channels.

4.1 Implementation

The implementation based on DES-Chan comprises two main classes for the program logic and the communication between individual instances: *DGA* and *Messaging*. Since the approach is a distributed algorithm, it is executed concurrently on every node. The following description refers to the instance running at one specific node. Before the algorithm is started, one of the network interfaces is tuned to a common channel and all other network interfaces are shut down. Then, an instance of the DGA class is created and initialized with an empty conflict graph and a new Messaging object. From that point, the program flow is determined by a series of events, as depicted in Figure 2.

The Messaging object listens for incoming messages from other DGA instances, and uses the *Neighborhood Discov-*

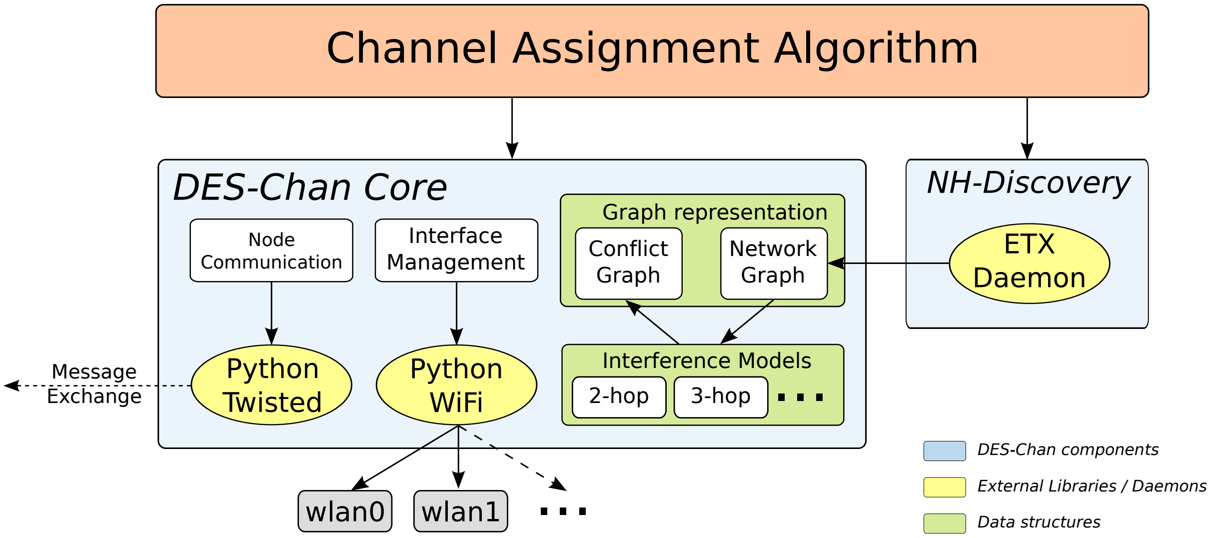


Figure 1: The DES-Chan framework for channel assignment comprises the DES-Chan Core and the Neighborhood Discovery service. The DES-Chan Core comprises functions to configure the wireless network interfaces and the message exchange with neighboring nodes. The Neighborhood Discovery service allows to retrieve the local network information.

ery module from DES-Chan to retrieve the initial network graph. The local conflict graph, which is still empty at that point, is updated with the information of the network graph. Afterwards, the *findMinimum* method is called. It identifies the vertex-channel combination, that minimizes the interference in the local conflict graph. For all these links, it tries every possible channel, calculates the overall interference, and compares it with the current interference level. If the interference can be further reduced, the program tries to apply the combination that provides the largest decrease. Therefore, it sends a `channel request` message to the neighbor which is represented by the conflict graph vertex, asking the DGA instance on that node to change the channel accordingly. If no combination could be found that further reduces the interference, the algorithm terminates.

After the `channel request` message has been sent, the program flow is interrupted until a message from another DGA instance is received. Thereby, three message types are distinguished: `channel request`, `channel reply`, and `channel reject`. In order to avoid deadlocks, the Messaging object automatically triggers a `channel reject` event, if a `channel request` message is not answered within a certain time.

When receiving the `channel request` message, the node checks whether it is able to change the channel without violating the interface constraint. If the corresponding network interface is only used for communicating with the sender of the request, it can simply be tuned to the new frequency. If, however, other neighbors are connected via the same network interface, a one-sided channel switch would eliminate these links. In that case, an unused network interface has to be set up and tuned to the respective channel. If there is an unused interface available, or no other links would be affected by the channel switch, the node answers with a `channel reply` message, indicating that the channel can be changed. After the reply has been sent, the node configures its interfaces according to the new assignment. In contrast, if the channel switch would affect links to other neighbors and no unused interfaces are left, the request is answered with a `channel reject` message and the vertex-channel combination is abandoned.

When the originator of the request receives a `channel reply` message, it applies the chosen channel to the particular link. To make sure that the channel can actually be switched to when the `channel reply` message is received, the program had allocated a lock for the unused interface before sending the channel request. Thereby it ensures, that the interface is not used to satisfy a channel request from another neighbor, while waiting for the corresponding `channel reply` message.

If the channel request is answered with a `channel reject` message, the respective link-channel combination is removed from the set of possible combinations. It will not be considered again by the *findMinimum* method. Thereby the termination of the algorithm is guaranteed.

After the requested channel has been either approved or rejected, the current network topology and the local conflict graph are updated accordingly. Afterwards, the next iteration is started and the remaining vertex-channel combinations are investigated until the interference cannot be decreased any further.

4.2 Deviations from the original algorithm

The presented implementation varies slightly from the proposed algorithm in [15]. These changes are attributed to the specifics of the DES-Chan framework. The Neighborhood-Discovery module in DES-Chan always provides up-to-date information about the current channel assignment. This information includes the 2-hop neighbors for each node with the corresponding links, their quality and channel they are tuned on. Since the algorithm instances retrieve this information in each iteration before searching for a suitable vertex-channel combination, they do not need to explicitly advertise channel changes into their local neighborhood.

Therefore, the `channel update` messages can be neglected. This eliminates the need to broadcast messages (which has to be done on all interfaces) and forward them to the 2-hop neighbors. The use of the Neighborhood Discovery module furthermore avoids inconsistencies that could occur, if `channel update` messages get lost. However, this does not reduce the network load, because these messages are implicitly exchanged anyway by the Neighborhood Discovery module.

Additionally, explicit `channel reject` messages have been introduced in the presented implementation. When a node receives a channel request, that cannot be fulfilled because of the interface constraint, it replies with a channel reject message instead of not replying at all. Thereby, the sender does not have to wait until its request times out and can immediately continue with the next iteration.

5. EVALUATION

The evaluation comprises a series of experiments with the DGA implementation on the DES-Testbed. We present graph-theoretic and throughput-based results, emphasize the importance of realistic interference models and discuss the experience with the algorithm in a real network.

5.1 Experiment set-up

The experiments for the evaluation of DGA were carried out on the DES-Testbed, a multi-radio wireless mesh network with 3 IEEE 802.11a/b/g radios on each network node. The testbed nodes are distributed in offices of multiple buildings at the Freie Universität Berlin [1, 5]. The results of the experiments consist of a graph-theoretic analysis of the remaining interference after channel assignment and a consideration of the actually achieved throughput. The results of DGA were compared to the single channel case, where each node uses only one of its interfaces and the interfaces on all nodes are tuned to the same channel. Additionally, the results are compared to the RAND algorithm. The RAND algorithm assigns channels to links randomly, thus also preserving the network topology.

As in the original evaluation [15], we only considered links with a *packet delivery rate* PDR ≥ 0.8 using the link quality values of the ETX daemon. The channels 1 to 13 of the 2.4 GHz spectrum were set as the available channels for the algorithms, since one wireless network card of the mesh routers is a Ralink-based 802.11b/g card only. An interference cost function taking the spectral distance of the channels was used, as shown in Equation (1).

$$I(f_1, f_2, \alpha) = \begin{cases} 0 & \text{if } |f_1 - f_2| \geq \alpha \\ 1 - \frac{(|f_1 - f_2|)}{\alpha} & \text{otherwise} \end{cases} \quad (1)$$

For IEEE 802.11b/g three channels of the 2.4GHz frequency band are theoretically orthogonal. Therefore, we set α to 30MHz, which results in the following set of orthogonal channels $\{1, 7, 13\}$. However, recent studies have shown that this theoretic assumption does not hold in practice [3, 6], and the impact on the achieved results will be discussed later.

The experiments were performed on 98 nodes in the DES-Testbed and repeated for 30 times. For the throughput measurements, `iperf` was used to generate UDP traffic flows with a data-rate of 54Mbit/s. Each replication comprised the following three phases:

1. *Single channel network* - All nodes were configured to set up only one interface on channel 1 in order to create the single channel network case. Afterwards the single-hop throughput on 10 selected non-adjacent links was measured.
2. *DGA* - In the second phase, the nodes were rebooted and the DGA algorithm was executed. After the algorithm had finished and the channel assignment was complete, we measured the throughput again on the same 10 links.
3. *RAND* - Finally, we repeated the process with the RAND algorithm and measured the throughput on the same 10 links again after the channel assignment.

One replication of the experiment lasted roughly 45 minutes, resulting in an overall experimentation time of 1 day.

5.2 Results

A sample channel assignment that resulted from both algorithms and the single channel network is depicted in screen shots of the DES-Vis visualization tool in Figure 8. Hereby, each link color represents a different channel. As can be observed from the link colors, the DGA algorithm mainly used the three channels that were reported to being orthogonal by the used interference model. In contrast, the random assignment by the RAND algorithm resulted in a higher channel diversity.

5.2.1 Fractional Network Interference

As in the original evaluation, the *fractional network interference* (FNI) was used as metric for the graph-theoretic consideration [15]. The calculations are based on the network graphs that were obtained from the algorithm runs on all nodes of the DES-Testbed. The local network graphs of all DGA instances were combined to a single network graph that corresponded to the whole testbed. The same interference model as used by the algorithm instances was then applied to generate the conflict graph and to calculate the overall interference in the network. Afterwards, all links in the network graph were set to the same channel and the overall interference was calculated again. The ratio of these two values results in the FNI. More formal, the FNI is given by:

$$FNI = \frac{\sum_{i \in E_{CA}} weight(i)}{\sum_{i \in E_{SC}} weight(i)} \quad (2)$$

where E_{CA} is the set of edges of the conflict graph after channel assignment and E_{SC} is the set of edges of the conflict graph in the single channel network.

The results are shown in Figure 3. After channels have been assigned with the DGA algorithm, the FNI was reduced to 0.28 in the median. Thus, the overall interference in the network was reduced by 72% compared to the interference in a single-channel network. Furthermore, the low variation in the results shows that the algorithm is reliable and achieves reproducible results. In contrast, the results from the RAND algorithm show a FNI of 0.43 in the median.

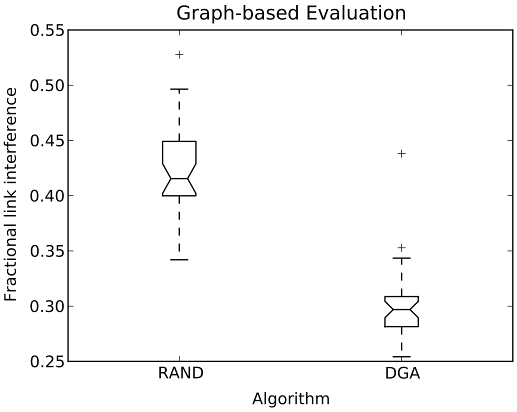


Figure 3: FNI evaluation of the channel assignment DGA and RAND. The graph theoretic consideration shows, that the DGA algorithm reduces interference by 72% compared to a single-channel network and by 15% compared to random channel assignment.

Thus, even random channel assignment significantly reduces the interference compared to a single-channel network, but the DGA algorithm still achieves a further reduction of 15%.

The presented results correspond to those that Subramanian et al. obtained for their distributed greedy algorithm on random graphs. For a setup with 50 nodes, 3 orthogonal channels, and 3 wireless network interfaces per node, the authors found that their algorithm achieves an average FNI of 0.3. The lower bound for this setup, which was obtained from *integer linear programming* (ILP) formulations, was reported to be around 0.2. The correspondence between the experimental results at the DES-Testbed and the theoretic analysis of the original authors validate the DGA implementation.

5.2.2 Throughput

For the throughput-based evaluation, 10 links were selected that were also considered for the channel assignment and thus have low packet loss rates. UDP flows were generated with `iperf` on all links simultaneously for 60 seconds with an data-rate of 54Mbit/s. The resulting flows were all non-adjacent, i.e., every node was involved in at most one flow. The throughput was measured for the single-channel case, and the channel assignment computed by DGA and RAND. To capture the performance of the whole network, the throughput of all links was summed up for each of these cases. The aggregate throughput achieved with the channel assignment from DGA and RAND was then put in relation to the throughput in the single-channel network.

As shown in Figure 4, the actual throughput improvement is not as high as promised by the graph-theoretic results. The graph shows that the throughput that is achieved after channels were assigned randomly is almost as high as after an assignment of the DGA algorithm. The aggregate throughput, that was measured on the 10 links after both channel assignments, was in the median 25% higher than in the single-channel network. Thus, although the assignment of the DGA algorithm is, according to the interference model, better than the assignment of the RAND algorithm, the actual throughput do not reflect these results. This confirms the predictions made earlier, that the used interference model based on the theoretical orthogonal channels is not appropriate for real network deployments [3, 6].

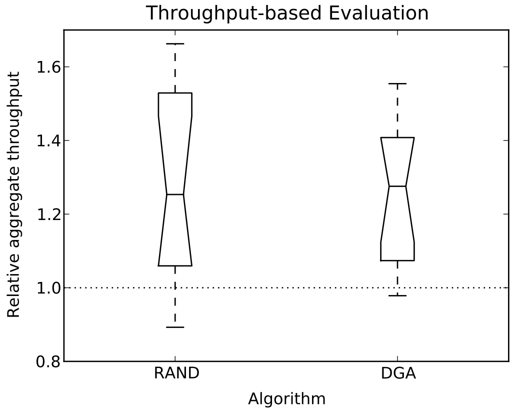


Figure 4: Throughput evaluation of the channel assignment DGA and RAND. The throughput results indicate that the graph based FNI evaluation holds only for realistic interference models. The throughput compared to the single channel case was improved by 25% by the random assignment and DGA. The variation of the results of the random assignment is higher than with DGA.

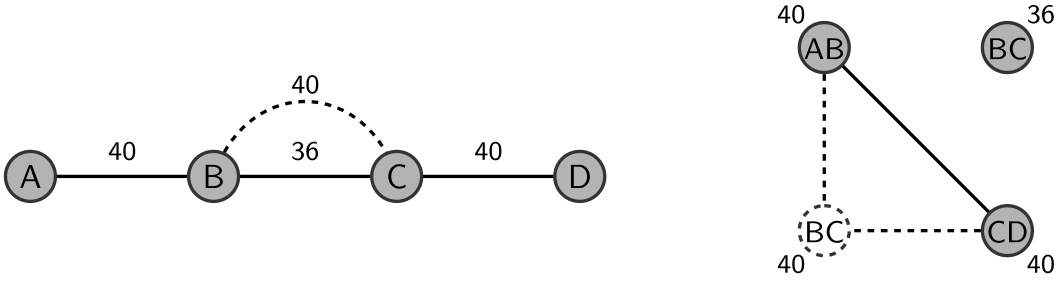
5.3 Discussion

5.3.1 Interference models

In order to validate the used interference model, we run additional experiments on the DES-Testbed. According to the utilized model, the set of channels $1, 7, 13$ is interference free. In order to validate the theoretically orthogonality, we calculated the *link interference ratio* LIR of two non-adjacent flows [10]. The LIR expresses the interference of two links by relating the aggregate throughput of both links when they are active individually to the aggregate throughput when they are active simultaneously. A LIR value of 1 indicates that the two links do not interfere at all, whereas a LIR value of 0.5 means that the aggregate throughput is halved when both links are active at the same time. The setup for the experiment is depicted in Figure 5. We run the experiments 40 times for each channel distance.

The results, as shown in Figure 6, imply that the used interference model does not hold in reality. Considering the LIR values in respect to the channel distance, no channel combination on the 2.4 GHz band is truly orthogonal. With a channel distance of at least 30MHz, the median of the LIR is above 0.8 which promises a higher throughput. Still, even with a channel distance of 60MHz the channels are not fully orthogonal.

For this reason, the focus of future work will be on more realistic interference models by considering measurement-based interference estimations in order to create more accurate models as proposed in [10, 11]. The promising potential of distributed channel assignment algorithms will only be fully exploited with more realistic and accurate interference models.



(a) Network graph (b) Conflict graph

Figure 7: Implicit links may be created, when the channel of an existing link is changed. In the example, link $B \leftrightarrow C$ was changed to channel 36, but B and D also have an additional interface on channel 40 in order to communicate with A and D respectively. Therefore also an implicit link on channel 40 exists between B and C.

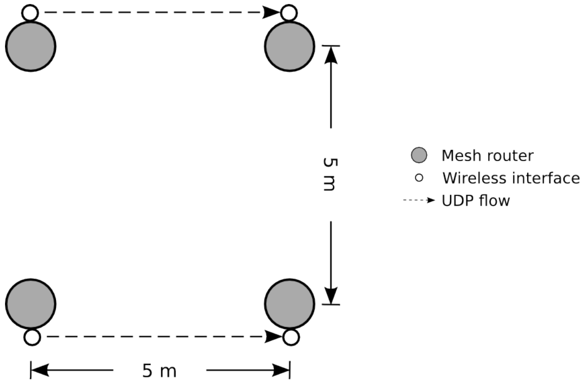


Figure 5: Experiment setup for measuring the effect of spectral channel distance to the LIR with non-adjacent traffic flows. Two node pairs are selected which are located in the same room. The LIR of the two links between the node pairs is measured.

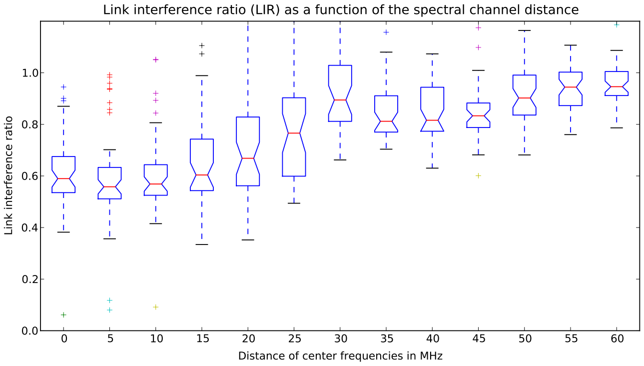


Figure 6: Results of LIR of non-adjacent flows for channel combinations on the 2.4 GHz band. The median LIR of 1 is almost achieved with a channel center distance of 60 MHz. With a channel distance of 30MHz, the median of the LIR is just above 0.8 which implicates that the full potential of orthogonal channels can not be achieved.

5.3.2 *Feasibility of link-based channel assignment*

Next to the evaluation based on the FNI and the achieved throughput, we took a closer look at the final channel assignments after running DGA. The results revealed some interesting aspects of the algorithm:

- *Implicit links* - The algorithm by Subramanian et al. considers only one link between two nodes in the network. Nodes are either connected or not, but multiple links between the same nodes using different radios are not taken into account. The channels of these links can be changed by the algorithm, but links do neither break apart, nor do new ones come up during the execution of the algorithm. However, in a real setup it may occur, that by changing the channel of one link, a new link is created implicitly. For example in Figure 7, the algorithm has changed the link $B \leftrightarrow C$ to channel 36. However, both have another interface tuned to channel 40 in order to communicate with their neighbors A and D respectively. Therefore, also an implicit link on channel 40 exists between B and C.

  The algorithm is not aware of this implicit link and will therefore not consider it in the channel assignment procedure. However, these implicit links violate the feature of the algorithm that it is topology preserving and therefore transparent for routing algorithms. Routing algorithms, which operate on the overlying IP layer, have no information about which links are desired and which links were created unintentionally. Therefore they may choose routes that include implicit links. Since the implicit links were not included in the interference calculations of the channel assignment algorithm, it is likely that the use of these links causes more interference than the use of the desired links. Thus, routing algorithms, that are neither aware of implicit links nor use a metric that favors channel diverse paths, are not able to utilize the full performance potential that was induced by the channel assignment. This contradicts the assertion of Subramanian et al., who claim that their approach can be used without specialized routing algorithms [15].

- *Dynamic links* - In contrast to the assumptions made by the algorithm, wireless links are very dynamic in real setups, even in stationary wireless mesh networks. Their quality may change over time and they exhibit different characteristics on different channels. Thus, existing links may break and new links may emerge during the course of the algorithm as has happened during the evaluation. Since the algorithm does not account for that, the resulting channel assignment may be sub-optimal, in the worst case, channels are assigned to links which temporarily do not exist and thus can not be used. As a solution, the information of the network topology could be updated periodically to notice recent changes and adapt to them accordingly to avoid dead links.

- *Poorly connected nodes cannot be reached* - The specific topology at the DES-Testbed, and most likely also at other real-world setups, requires to use low-quality links in order to reach all nodes. Selecting only high-quality links (quality $\geq 80\%$ in the implementation of Subramanian et al.) as basis for channel assignment excludes a fraction of nodes that can only be reached via low-quality links. Since the low-quality links do not exist at all in the view of the algorithm instances, they break apart completely when the nodes change their channels, which can result in network partitioning, which is not desired.

In summary, we consider the link-based approach not yet fully feasible for the application in real environments. The issues with the implicit links has to be fixed in order to be fully transparent to the routing layer. Additionally, the algorithm should operate on the actual topology and include low-quality links to ensure the network connectivity.

6. SUMMARY AND OUTLOOK

We presented an experimental evaluation of the distributed, link-based channel assignment algorithm DGA based on the DES-Chan framework. The evaluation was performed in the DES-Testbed with 98 multi-radio mesh nodes covering a graph-theoretic analysis of the experiment results and measurements of the achieved throughput after the channel assignment. Based on the results, we discuss the feasibility of link-based channel assignment and show that the promising feature of the algorithm to be transparent for routing algorithms, is not guaranteed. We identified challenges in link-based channel assignment with DGA, such as the impact on the network connectivity and consequences of dynamic topologies. Finally, we showed that realistic interference models are required to fully exploit the performance gain achievable with channel assignment.

Future work will focus on solutions for the identified challenges of link-based channel assignment and the drawbacks of DGA. Additionally, studies on more realistic, measurement-based interference models will be carried out. With these models, it is likely to close the gap between the graph-based and throughput-based results obtained from the experiments on the DES-Testbed. Further on, we will implement an interface-based algorithm, which assigns links to interfaces instead of links. An comparison of such an approach to DGA in a large network such as the DES-Testbed will hopefully be insightful for novel algorithms.

7. REFERENCES

[1] B. Blywis, M. Guenes, F. Juraschek, and J. Schiller. Trends, advances, and challenges in testbed-based wireless mesh network research. *Mobile Networks and Applications*, 15:315–329, 2010. 10.1007/s11036-010-0227-9.

[2] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 134–146, New York, NY, USA, 2003. ACM.

[3] P. Fuxjager, D. Valerio, and F. Ricciato. The myth of non-overlapping channels: interference measurements in ieee 802.11. In *Proc. Fourth Annual Conference on Wireless on Demand Network Systems and Services WONS '07*, pages 1–8, 2007.

[4] M. Günes, B. Blywis, F. Juraschek, and O. Watteroth. Experimentation Made Easy. In ICST, editor, *Inproceedings of the First International Conference on Ad Hoc Networks*, Ontario, Canada, September 2009.

[5] M. Günes, F. Juraschek, B. Blywis, Q. Mushtaq, and J. Schiller. A testbed for next generation wireless networks research. *Special Issue PIK on Mobile Ad-hoc Networks*, IV:208–212, Oktober-Dezember 2009.

[6] F. Juraschek, M. Günes, M. Philipp, and B. Blywis. State-of-the-art of distributed channel assignment. Technical Report TR-B-11-01, Freie Universität Berlin, FB Mathematik und Informatik, Jan 2011.

[7] B.-J. Ko, V. Misra, J. Padhye, and D. Rubenstein. Distributed channel assignment in multi-radio 802.11 mesh networks. pages 3978–3983, 2007.

[8] P. Kyasanur, J. So, C. Chereddi, and N. H. Vaidya. Multichannel mesh networks: challenges and protocols. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 13(2):30–36, 2006.

[9] P. Kyasanur and N. H. Vaidya. Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 10(1):31–43, 2006.

[10] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill. Estimation of link interference in static multi-hop wireless networks. In *IMC '05: Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 28–28, Berkeley, CA, USA, 2005. USENIX Association.

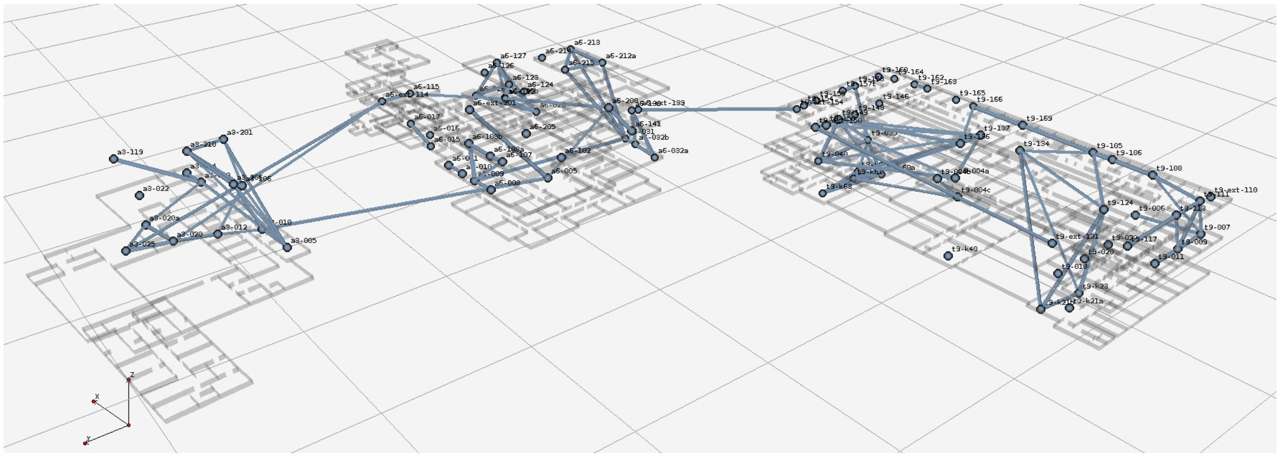
[11] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference in static wireless networks. *SIGCOMM Comput. Commun. Rev.*, 36(4):51–62, 2006.

[12] M. Shin, S. Lee, and Y. ah Kim. Distributed channel assignment for multi-radio wireless networks. *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, 0:417–426, 2006.

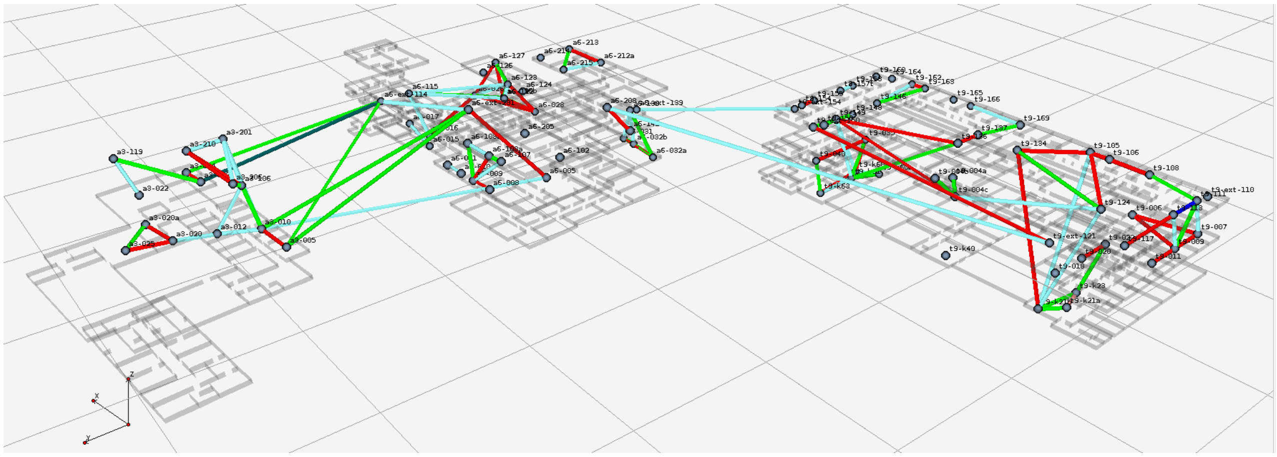
[13] W. Si, S. Selvakennedy, and A. Y. Zomaya. An overview of channel assignment methods for multi-radio multi-channel wireless mesh networks. *Journal of Parallel and Distributed Computing*, pages –, 2009.

[14] S. sridhar, J. Guo, and S. Jha. Channel assignment in multi-radio wireless mesh networks : A graph-theoretic approach. 2009.

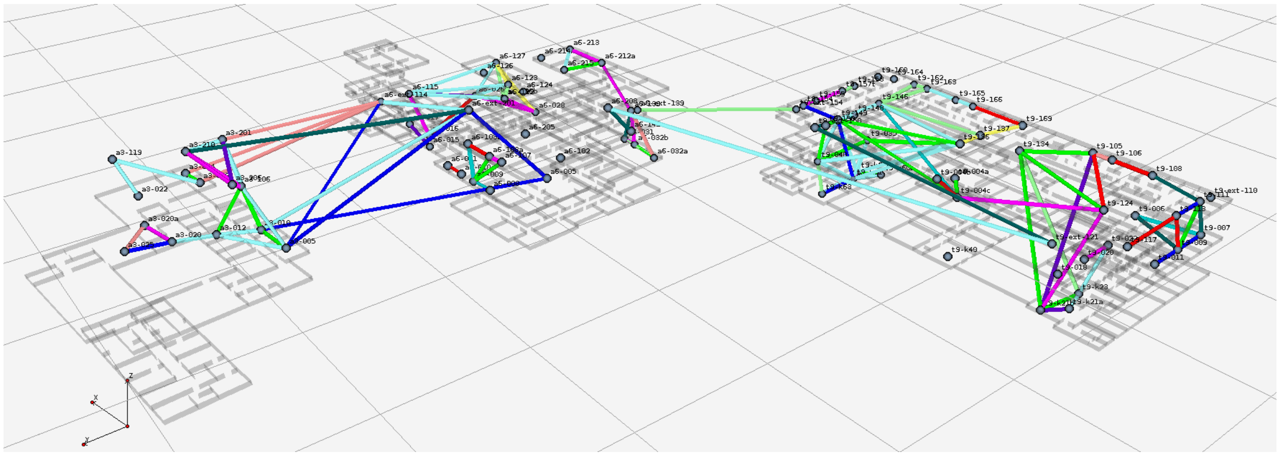
[15] A. P. Subramanian, H. Gupta, S. R. Das, and J. Cao. Minimum interference channel assignment in multiradio wireless mesh networks. *IEEE Transactions on Mobile Computing*, 7(12):1459–1473, 2008.



(a) Single channel network



(b) Channel assignment with DGA



(c) Channel assignment with RAND

Figure 8: The images have been taken from the DES-Vis 3D-visualization tool for the DES-Testbed. First, all links are shown which are used for the channel assignment. Both assignments only take links into account with a quality of $\frac{1}{ETX} = 0.8$. Resulting channel assignment of DGA and RAND at the DES-Testbed. Different link colors correspond to different channels. While the random assignment created a high channel diversity, DGA mainly used the three channels that were reported to being orthogonal by the used interference model that took the spectral distance into account.