ColorCast: Deterministic Broadcast in Powerline Networks with Uncertainties

Yvonne Anne Pignolet[1], Stefan Schmid[2,3], Gilles Tredan[3]
[1] ABB Corporate Research, Switzerland; yvonne-anne.pignolet@ch.abb.com
[2] Telekom Innovation Laboratories & TU Berlin, Germany; stefan@net.t-labs.tu-berlin.de
[3] CNRS-LAAS, France; gtredan@laas.fr

***Abstract*—This paper initiates the study of broadcast in a powerline communication network, where nodes communicate by local broadcasts in the grid, and where the quality of the communication links is subject to uncertainty. We first show that state-of-the-art broadcast algorithms for known topologies fail to broadcast messages in such a challenging environment, even when the link quality uncertainty is small. We then present a deterministic algorithm COLORCAST that distributes a message to all nodes in the network in time $\Theta(n)$. The algorithm is based on graph coloring and avoids collisions while guaranteeing a high parallelism. In particular, COLORCAST strictly outperforms existing deterministic broadcast algorithms for unknown topologies in the sense that its time complexity is asymptotically lower than the best possible runtime for the unknown setting. Our formal analysis is complemented with a simulation study on real grid topologies, which confirms the benefits of COLORCAST compared to state-of-the-art protocols.**

I. INTRODUCTION

Broadcast refers to the basic task of transmitting a single message originating from some source node s to all $n-1$ remaining nodes $V \setminus \{s\}$. This fundamental problem has been studied in many settings, from wireless networks consisting of nodes with omnidirectional antennas, to wireline networks with point-to-point communication. An interesting and less well-understood communication network gains in importance with the modernization of the electrical grid infrastructure: powerline communication (*PLC*) networks.

PLC is used by utilities to control and manage the grid without building an additional communication infrastructure. While powerline communication has been used on point-to-point links for many years in the grid's high voltage backbone network, we currently witness a trend towards multi-hop low and medium voltage PLC networks to enable "smart grid" functionalities in the distribution grid [10], [15]. Smart grid applications are envisioned to use medium voltage PLC networks to monitor e.g., current and voltage values at transformers as well as the health status of grid equipment, and to send control commands and settings to switches, circuit breakers and transformers.

This paper initiates the study of the broadcast problem in a medium voltage PLC network. For example, broadcasts are useful in the context of *adaptive protection*: in order to ensure the reliable and efficient operation of power grids, adaptive protection changes the settings of the protection equipment depending on the network load, capacity and configuration. Broadcast communication services are needed for control commands, e.g., to disseminate currently valid settings.

Broadcast in powerline networks also constitutes a challenging algorithmic problem: (1) First, the communication topology may look quite different from the underlying grid topology; this renders the problem difficult even for simple grid topologies (e.g., trees). (2) Similarly to wireless networks, the received signal is typically subject to various types of noise, interference and uncertainties and when more than one device emits a signal collision can occur. Indeed, today there is no well-accepted model for the analysis of MAC or higher-layer protocols for medium voltage PLC.[1] (3) Also hidden terminal problems may occur, and thus carrier sensing and collision avoidance mechanisms (CSMA/CA) cannot void collisions nor reliably detect them when broadcasting without acknowledgments.

Indeed, in PLC, the achievable communication quality between two devices varies depending on the powerline paths between them and the current radio and electrical condition [18]. While studies have shown that in medium voltage networks, the packet success rate is strongly dependent on the distance between the two nodes that communicate and the potential concurrent transmissions from other senders, many other static and dynamic factors, such as the quality of the line, electrical switches, circuit breakers, transformers and loads influence packet transmission [13], [18].

As a consequence, although the powerline communication infrastructure is known, the uncertainty on the effective transmission ranges due to varying link quality at runtime implies a partly unknown communication (and interference) topology on top of which broadcast has to be carried out.

Contributions. This paper initiates the study of reliable broadcast algorithms in powerline networks. First, we introduce an interesting new graph model for powerline networks, where nodes communicate via local broadcasts in the grid, and then extend this model to take into account uncertainty, in the sense that nodes do not know their effective (local broadcast) neighbors. That is, our paper assumes an interesting new position between *known* and *unknown* models: while the *underlying grid graph* is known to the algorithm, the current *communication graph* is subject to uncertainty.

[1] For a selection of existing models (most of them targeted at low voltage use cases) we refer to [3], [4], [14], [17], [19].

We first show how existing broadcast algorithms for *known topologies* fail to broadcast a message in finite time, even in a model where the link quality varies only by a small amount (Section III). Given this negative result, we present the distributed and deterministic algorithm COLORCAST to solve the PLC broadcast problem (Section IV). The worst case broadcast time of COLORCAST is $O(n)$; this is strictly lower than the worst-case broadcast time of any *unknown-topology* algorithm. We also report on our simulation study on a Swiss medium voltage grid topology, and compare COLORCAST to a heuristic and randomized approach. Our results suggest that COLORCAST does not only provide good worst-case guarantees on the broadcast time complexity, but also performs well in realistic scenarios (Section V).

II. MODEL

We represent the underlying electrical grid used for powerline communication as a weighted graph, where nodes represent the communication devices and the edges represent the powerlines connecting them as well as their distances. The communication devices (nodes) in the grid can communicate by *local broadcasts*, reaching a certain set of other devices, depending on the current link qualities and simultaneous transmissions: Nodes which are located close in the physical network can *always* communicate; however, depending on the current network conditions and/or configurations nodes in larger distances may not be able to receive messages.

Formally, we model a grid topology as a *directed* weighted graph $G = (V, E, d)$ which connects communication devices (i.e., *nodes*) V along powerline links $E \subset V \times V$, where the distance of edge e is denoted by $d(e)$. Apart from its length, a link's communication quality depends on many factors, such as electromagnetic interference and impedance effects from electric appliances. We model this time-dependent quality as a varying noise level for each link, described by a function $\rho^t : E \rightarrow [0, \rho_{\max}]$. Note that we do not require that $\rho^t(u, v) = \rho^t(v, u)$ for $u, v \in V$ to account for the fact that communication links in PLC are not necessarily symmetric.

The larger a link distance d and the higher the noise level ρ, the less likely is a successful message reception. Concretely, we define a $f : (d, \rho) \rightarrow \mathbb{R}^+$ which is monotonic in both d and ρ and satisfies $f(d, \rho) \geq d$. In other words, f can be used to compute a weight for each link at time t which determines its current link quality. We can interpret this weight as a *virtual distance*. As a shorthand notation, we define $\delta^t(e) = f(d(e), \rho^t(e))$ to be the virtual length of e which can change over time, depending on the noise level.

Given these concepts, we can compute $G^t_{com} = (V, E^t_{com})$, the *communication graph*, connecting nodes which can communicate at a certain time t. In our model, the node transmission power is set to one unit, and we assume that a message reaches all nodes for which the remaining power is still nonnegative, formally: $(v_i, v_j) \in E^t_{com}$ means that there is a path in G s.t. $sp^t(v_i, v_j) \leq 1$, where $sp^t(\cdot, \cdot)$ denotes the length of the shortest path between the two nodes, based on the time-varying virtual distances $\delta^t(\cdot)$.

We define $G^{\perp}_{com}$ to be the "worst case" communication graph where noise levels are maximal, and $G^{\top}_{com}$ to be the "best case" when the noise level is minimal on all edges. We denote the corresponding edge sets by $E^{\perp}$ and $E^{\top}$: $(v_i, v_j) \in E^{\perp} \Leftrightarrow$ j's message will always reach i, even in the worst case, and $(v_i, v_j) \in E^{\top} \Leftrightarrow j$'s message can reach i in ideal conditions.

Since powerlines form a shared medium where concurrent transmissions can collide, we state the following conditions that need to be met to guarantee a successful transmission: Node v_j receives the message sent from v_i at round t if

- v_i and v_j are in communication range at this round: $(v_i, v_j) \in E^t_{com}$
- v_i is the only node in v_j's range to send a message.

If another node v_k in the communication range sends currently, i.e., $(v_k, v_j) \in E_{com}$, then the two messages might interfere and v_j may or may not be able to receive any of v_i and v_j's messages.

We consider the broadcast problem on PLC networks where some node $s \in V$ (the broadcast *root* or *source*) needs to send a message to all other nodes $V \setminus \{s\}$. A synchronous environment is assumed in which time proceeds in discrete rounds: a message transmitted in round t is received in the same round. At $t_0 = 0$, the source s transmits the message and we want to minimize the time t until all nodes V have successfully received the message.

To guarantee that broadcast can terminate successfully regardless of the network conditions, we require the worst case scenario to still offer a solution, in other words we only consider scenarios where $G^{\perp}_{com}$ is connected. We assume that the nodes know the physical grid network topology G in advance. This assumption makes sense in a smart grid scenario since there the devices do not move and are designed to be in operation for decades. Nodes have an upper bound on the noise level $\rho_{\max}$, and the function f. What is unknown to the nodes is the current noise level $\rho^t, \forall t$ and hence the resulting current communication graph.

In the remainder of this paper, we will refer to the powerline broadcast problem with unknown link qualities by **PBC**. We will measure the total number of communication rounds used by an algorithm ALG in the worst case. Concretely, the time complexity is the time when each node has received the message.

III. THE CHALLENGE OF UNCERTAIN LINKS

Unknown current network conditions which influence the communication topology render the implementation of broadcasts significantly more difficult. In order to acquaint the reader with the model and to highlight the challenge of uncertain communication links, we first discuss a small example where existing algorithms based on known topologies fail to broadcast a message.

Many efficient broadcast algorithms for known topologies rely on the notion of a *wave front*: the algorithmic wit is focused on the frontier of nodes having the message (the potential senders) and their immediate vicinity (the potential receivers). Informally, in our model with unknown transmission ranges, it is *a priori* impossible to plan where the wavefront will be at a given time; moreover, since the communication graph is directed (and possibly only weakly connected), the *a posteriori* knowledge of which nodes received the message may remain local as well.

Let us now consider the classic *ranked-gathering-tree* approaches in more detail, and in particular the representative algorithm described in [8]; henceforth, we will refer to this algorithm by GPX. GPX requires a known topology G. First, GPX computes an arbitrary breadth-first spanning tree T on G. Hop distances in this tree are denoted by $d_T(\cdot,\cdot)$. Subsequently, the nodes in T are assigned a *rank* as follows: a leaf is of rank 1. For a non-leaf node v, we first determine the ranks of its children. If x is the maximal rank of any child and if only one child is of rank x, we set $\texttt{rank} = x$; otherwise, we set $\texttt{rank} = x + 1$. It is easy to see that the maximal rank of any node in G is at most logarithmic in the number of nodes (i.e., $O(\log n)$): the rank only increases when a node has at least two children of this rank. Based on this ranked spanning tree, a ranked *gathering spanning tree* rooted at the source node is computed. The tree edges at hop distance i from the root connecting nodes with the same rank can be scheduled in a collision-free manner.

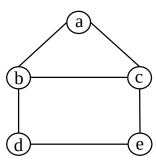

Fig. 1. Bad example where GPX fails.

Unfortunately, these approaches fail under link uncertainties. Let us consider the physical powerline graph G in Figure 1. Assume the source a wants to broadcast. Moreover, assume that all edges $e \in E(G)$ are short, such that the incident nodes are also neighbors in the communication graph. Under unknown transmission ranges, any possible gathering tree can be chosen for GPX. In our example, there are three non-isomorphic options (Figure 2): (1) The gathering tree assuming minimal virtual distances, in which case, there is only one breath-first spanning tree, and hence also the gathering tree T is unique: the edge set E of T is $E = \{\{a, b\}, \{a, c\}, \{b, d\}, \{c, e\}\}$. (2) Alternatively, consider the gathering tree where all communication ranges or virtual distances are twice as large. Also here, the breadth first spanning and hence the gathering tree T is unique: a star rooted at a, i.e., $E = \{\{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}\}$. (3) A mixed scenario where the transmission ranges vary. In this case, we can have a gathering tree T where c is connected to e in addition to d.

Note that the maximum rank of all the gathering spanning trees is 2, and for both the minimal and the maximal range trees, the root is the only node with rank 2. In all these possible gathering trees, GPX can fail to broadcast: messages from nodes c and b collide at node d

Theorem 1: The broadcast algorithm GPX fails to solve **PBC** if link qualities differ from the expected ones.

Proof: Consider the simple setting illustrated Figure 1. After round 1, assuming only b and c received the message from source a, GPX needs to decide whether b, c or both b and c should transmit, where the network can be in any of the four configurations shown in Figure 3.

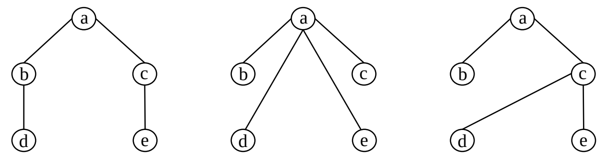

Fig. 2. Three non-isomorphic spanning trees for the topology in Figure 1.

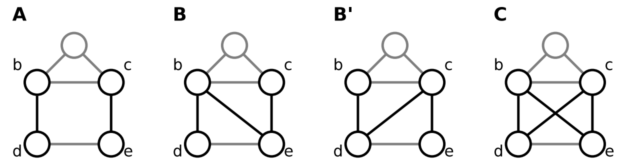

Fig. 3. At round 1, nodes *b* and *c* have a message to transmit. Yet 4 different communication graphs can be realized.

Figure 4 shows the different possible outcomes on the bet made by GPX: in rows, the planned topology is shown, in columns, the actual topology. We see that if GPX plans for **A** but **C** is realized, e and d will not be able to receive the message due to collision, since in topology **A** nodes e and d send simultaneously. In general, assuming a too dense topology leads to nodes not receiving the message (lower diagonal of the table), and assuming a too sparse topology leads to collisions (upper diagonal of the table).

There is no line containing only OK's, so a scenario where GPX fails always exists. ■

IV. COLORCAST AND ANALYSIS

This section presents a broadcast algorithm which avoids the problem of GPX [8] in settings with link uncertainties. COLORCAST (Algorithm 1) is based on a coloring approach, and seeks to conservatively avoid collisions by scheduling two nodes u and v which may interfere at some node w (i.e. if $\exists w$ s.t. $(u, w) \in E^{\top}$ and $(v, w) \in E^{\top}$) in different rounds. This is achieved by computing the following coloring-based schedule. First, an arbitrary *Minimal (Connected) Dominating Set* $\texttt{CDS}$ is computed on $G^{\perp}_{com}$. This $\texttt{CDS}$ ensures connectivity in the sense that any two dominators are within each other's communication range, even when the link qualities are worst possible, due to high noise levels. Subsequently, starting from the source, COLORCAST computes a breadth-first spanning tree on $\texttt{CDS}$, the worst-case communication graph with a high noise level, and divides the tree into layers L_i of increasing distances. Then COLORCAST computes a minimal coloring on the dominator nodes of $\texttt{CDS}$, for each layer L_i separately, in the L_i-induced subgraph of $G^{\top}_{com}$ *with minimal noise level*. Henceforth this graph is simply denoted by $G(L_i)$. Let ξ_i denote the chromatic number of $G(L_i)$ and $\xi = \sum_{i=1} \xi_i$ the sum of the chromatic number over all layers. By this layer coloring, COLORCAST *avoids collisions* entirely: each color constitutes an independent set on the interference graph, and the nodes cannot interfere, even if the noise level is locally or even globally lower than $\rho_{\max}$. Figure 5 illustrates the layering and coloring of the COLORCAST Algorithm.

	A	B	B'	C
A	OK	e: Col.	d: Col.	e, d: Col.
B	e: $\emptyset$	OK	e: $\emptyset$	e, d: Col.
B'	d: $\emptyset$	d:$\emptyset$	OK	e, d: Col.
C	d/e: $\emptyset$	OK/ e:$\emptyset$	d:$\emptyset$ / OK	OK

Fig. 4. Table enumerating possible outcomes: each column is an actual topology, each row is the topology expected by GPX. *Col.* stands for collision and $\emptyset$ stand for no message received.

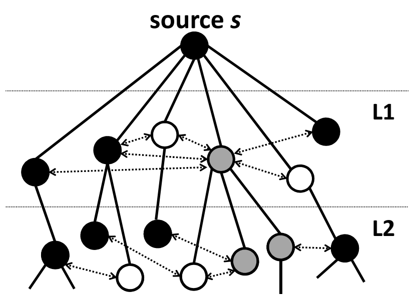

Fig. 5. Visualization of COLORCAST. The algorithm structures nodes along layers (here: two layers), starting from the source s and at low-range intervals ($\rho_{\max}$ noise). Each layer is colored, as indicated by the different node colors (black, grey, white). The spanning tree on the connected dominating set is shown in solid lines, while interference edges (for the layer coloring with respect to $E^{\top}$ interference) are dotted. Communication links are not shown explicitly in this figure.

Algorithm 1 COLORCAST $(G(V, E, \ell), s, f, \rho_{\max})$

/* $E^{\perp}$-connected dominating set construction */
1: $D \leftarrow \texttt{CDS}(G^{\perp}_{com})$
2: $T \leftarrow$ a spanning tree of the $G^{\perp}_{com}$subgraph induced by $D \cup \{s\}$
3: let $L_i = \{v \in T | d_T(v, s) = i\}, \forall i \in [0, \max_{v \in T} d_T(v, s)]$
4: let R be an integer array of size $|V|$ initialized at $\perp$
/* *round assignment* */
5: $round \leftarrow 0$;
6: **for each** L_i **do**
7: $G(L_i) \leftarrow (L_i, E_i = \{v_i, v_j \in L_i^2 | \exists w \in V, (v_i, v_j) \in E^{\top} \wedge (v_j, w) \in E^{\top}\})$
8: let $\Xi_i : L_i \rightarrow [1, \xi_i]$ be a coloring of $G(L_i)$
9: **for each** $v \in L_i$ **do**
10: $R[v] \leftarrow round + \Xi_i(v)$
11: $round \leftarrow round + \xi_i$
12: **return** R

Lemma 1: COLORCAST (Algorithm 1) produces a collision-free schedule.

Proof: Let $G(V, E, d)$ be a powerline network, and $s \in V$ the broadcast source. Let R be the transmission schedule produced by COLORCAST, i.e., the binary variable $R[v_i, t] \in \{0, 1\}$ indicates whether node v_i transmits in round t. The proof proceeds by contradiction: Assume that two or more nodes in node w's communication range all transmit during round t. Let i and j be two of the nodes responsible for this potential collision, $R[v_i, t] = R[v_j, t] = 1$. Since a simultaneous transmission of these nodes may cause a collision we deduce that $(v_i, w) \in E^{\top} \wedge (v_j, w) \in E^{\top}$.

First observe that if $R[v_i, t] = R[v_j, t] = 1$, due to the update of $round$ in Line 12, then necessarily i and j belong to the same layer. Let L be this layer, and Ξ the corresponding coloring obtained in Line 9. Because of Line 11, necessarily $\Xi(v_i) = \Xi(v_j)$. Since Ξ is a legal coloring of $G(L)$, we deduce that $(v_i, v_j) \notin E^{\top}$, which contradicts the existence of w as close to a and b, and hence the definition of E (Line 8). ■

Lemma 2: At the end of the schedule produced by COLORCAST, all nodes have obtained the broadcast message.

Proof: Let $G(V, E, d)$ be a powerline network, and $s \in V$ the broadcast source. Let R be the schedule produced by Algorithm 1. We first show by induction on the layers L_i that all the nodes of T get the message. The inductive step is that *if all the nodes of layer L_i have the message at round k, then all the nodes of layer L_{i+1} have the message after round $k + \xi_i$.*

The base case is simple: for $i = 0$, the source s has the message by definition and $L_0 = \{s\}$. Now assume that at some round $k < \infty$, all nodes from layer L_i have the message. Let $u \in L_{i+1}$, u has at least one parent p in T on layer L_i. Observe that from round k to round $k + \xi_i$, all nodes of layer L_i will forward the message ($\forall v \in L_i, k < R[v] \leq k + \xi_i$). Thanks to Lemma 1 we know there cannot be any collisions, and since $(u, p) \in E^{\perp}$ and therefore necessarily $(u, p) \in E(G^k_{com})$, u will receive the message. Thus at round $k + \xi_i$ all the nodes of layer L_{i+1} have the message, which proves the induction step.

Thus all the nodes of T received the message at some point in time. Since they forward it at some later step since T is a dominating set of $G^{\perp}_{com}$, all the nodes get the message eventually. ■

Theorem 2: COLORCAST solves the **PBC** problem on power line networks with uncertainty in time at most n.

Proof: The correctness of the solution produced by COLORCAST follows from Lemma 2. Let us now consider the length of the schedule. Observe that each node sends in exactly one time slot due to the disjoint layers and the coloring. Moreover there are no time slots without at least one node transmitting, otherwise the coloring would not be minimal. Thus the schedule comprises at most n time slots. ■

Note that this time complexity is tight in the sense that no algorithm can achieve a better time complexity on a chain network with distances that prevent communication over more than one hop. Furthermore, we emphasize that the length of the schedule is n, and not $O(n)$, i.e., no constant factors are hidden.

V. SIMULATIONS

We evaluate our algorithm on the topology of a real urban electrical grid of a town in Switzerland (population approx. 20k, area approx. 14 km^2), see Figure 6. The grid consists of 93 nodes (primary substations and ring main units) connected by 107 edges. Typically, the distances between two neighboring ring main units are between 200m and 2000m in this area of Switzerland. Hence, we use the powerline connectivity information provided by the utility as the graph $G = (V, E)$, and choose the weights ℓ uniformly at random between 200 and 2000. This corresponds to a grid without elements that disconnect PLC links (like open switches and transformers), and hence, we study a scenario with the maximum number of possible collisions. We generate 100 different graphs $G = (V, E, \ell)$ in this manner.

We investigate the effect of a multiplicative link quality function, i.e., $f(d(e), \rho^t(e)) = d(e) \cdot (1 + \rho^t(e))$ for a static scenario where the noise level may differ between edges, but it does not vary over time: $\rho^t(e) = \rho(e) \in [0, \rho_{max}]$ (the time complexity of the algorithm is independent of the current unknown conditions). We quantify the influence of varying ρ_{max} on the diameter and time complexity of broadcast algorithms with values derived from realistic scenarios. The powerline communication characteristics described in [17] and the model presented there lead to a bit error rate (BER) of close to 0 up to a distance of 2000m, and then increase sharply. Based on this, we interpret $f(d, \rho)$ as a virtual distance in meters and assume that within a communication range of 2000m nodes can communicate with each other and thus do not consider longer edges.

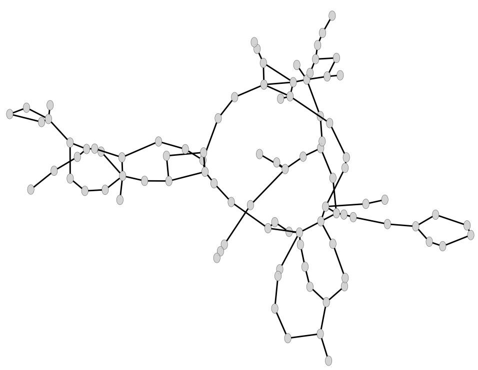

Fig. 6. Topology of a medium voltage grid of a town in Switzerland.

To have a benchmark for the performance of ColorCast, we also implemented the simple randomized Decay algorithm (cf Algorithm 2) described in [2] with a time complexity of $O((D + \log(n/\epsilon)) \cdot \log \Delta))$ with high probability, where D is the network diameter and ϵ is a parameter. While the maximal degree Δ can reach n, but can also be much lower, we ran Decay both with $\Delta = n$ and with $\Delta = \max \text{degree} \left(G^{com}_{\rho_{min}}\right)$. In order to avoid penalizing Decay for the fact that it is a randomized algorithm, we set ϵ to 1. This has the drawback that in our simulations Decay is not always reaching all nodes. However, in our experiments this event occurred in less than 4 % of all cases, and we believe that the algorithm is well suited as a benchmark in this setting.

Algorithm 2 Decay(Δ, n, ϵ)

set k to $\log \Delta$; wait until receiving a message m
for $\log(n/\epsilon)$ times **do**
  wait until $time \mod k = 0$
  repeat
    transmit m, set $coin$ to 0 or 1 with equal probability
  until $coin = 0$ or sent k times

A. Influence of ρ_{max}

Figure 7 *(left)* plots the average duration of a broadcast on the topology of Figure 6, for different $\rho_{\max}$ values. We compare ColorCast with Decay parametrized with two different estimates for Δ: maximum degree D of $G^{com}_{\rho_t}$ or number of nodes n. While the time complexity of ColorCast does not vary with the actual ρ_t, it influences the performance of Decay. Therefore, we run Decay on the same topology with three different assignments and average them: (i) In the first scenario, ρ_t is set to 0 for all edges; (ii) in the second scenario, ρ is chosen uniformly at random between 0 and ρ_{max} for each edge; (iii) in the third scenario, edges are subject to ρ_{max}. Since ColorCast only relies on ρ_{max} and not on the actual virtual distance of the edges, its performance is not affected by these different scenarios.

ColorCast clearly outperforms Decay on this medium-sized grid even in the scenario with most uncertainty and despite the randomized approach of Decay which theoretically allows for asymptotically lower runtimes on average. When ρ_{max} is chosen uniformly at random, this increases the effective diameter from 12.37 to 17.38 and the average degree shrinks from 11.3 to 5.1 for maximal noise levels. However, as can be seen by the confidence interval for Decay, even when the noise level is maximal on all edges, ColorCast always completes broadcast faster.

B. Impact of Scale

In order to study the impact of larger network sizes, we iteratively attach two copies of the basic network to each other. The two copies are connected by adding links between three randomly chosen pairs of nodes of the two copies. This is reasonable, since larger distribution grids often have the same density as smaller distribution grids. In this manner, we construct 100 networks with $n = 93 \cdot 2^k$ nodes, for k between 0 and 5; thus, the largest networks contain 2976 nodes.

Figure 7 *(center)* studies how the broadcast time depends on the network size n. According to our construction of networks, the diameter grows roughly linearly with the size of the network, while the average degree grows by around 5. With larger network sizes, the factors in the O-notation matter less and Decay starts to exhibit better performance than ColorCast for networks with more than around 1000 nodes. However, the difference in number of rounds is not very large, i.e., for networks with 2976 nodes, ColorCast needs 263 rounds on average (std 14) while Decay using the maximum degree for Δ finishes after 204 rounds (std 42). In general, Decay is subject to a high variance (vertical bars represent the standard deviation of runtime over 100 runs), the different topologies influence ColorCast' performance only slightly. This, in combination with the fact that Decay cannot guarantee that the message reaches all nodes in all cases and fails to do so in up to 4% of all runs as well as the fact that Medium Voltage Grids are not arbitrarily large, shows that ColorCast is well suited for PLC networks.

Figure 7 *(right)* sheds more light on these performance results: it shows the size distribution of the layers over the 100 different runs on each topology. A low number of colors facilitates a parallel traversal of the layers, and hence ensures a quick propagation of the broadcast message. Since the majority of layers are traversed in less than 10 rounds, regardless of the topology size, ColorCast is efficient.

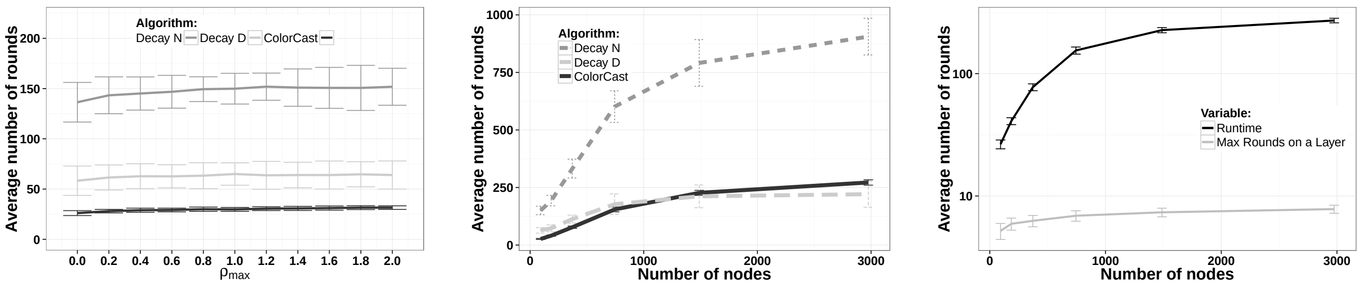

Fig. 7. Impact of ρ_{max} *(left)* and the network size *(center)* on the broadcast time for ColorCast and Decay. Number of colors per layer *(right)*.

VI. Related Work

To the best of our knowledge, the broadcast problem under uncertain link qualities in grid networks has not been considered so far. However, several results from models with or without knowledge on the topology also apply in our setting. For an excellent overview of the broadcast problem in various radio network models (with an emphasis on time-complexity), we refer the reader to the survey by Peleg [16].

Existing broadcast models often differ in that either the network does or does not support collision detection, or in that the radio network topology is known or not. Our model can be seen as an instance of a radio network model, in the sense that the transmissions of nodes are subject to topological constraints and collisions can happen when more than one node transmit at the same time. However, in contrast to most radio models, the underlying graph describes a PLC network with a different signal propagation than in radio networks.

A seminal work on the time-complexity of broadcast in multi-hop networks is by Bar-Yehuda et al. [2] who show an exponential gap between deterministic and randomized algorithms in the same collision model we adopt. Moreover they present a simple distributed randomized oblivious algorithm for unknown directed networks, which works in our model too. We used this algorithm to compare ColorCast in simulations and demonstrated that depending on the setting, they can outperform each other.

When restricted to deterministic algorithms on unknown networks, lower bounds of $\Omega(n \log n / \log(n/D))$ and $\Omega(n \log D)$ are presented in [11] and [6] respectively. Deterministic algorithms for this scenario achieve broadcast in time $O(n \log n)$ [11], $O(n \log^2 D)$ [7] and $O(D\Delta \log^\alpha n)$ [6], where Δ is the maximum degree of the network and $\alpha \geq 2$ unless n is known ($\alpha = 2$) or n and Δ are known ($\alpha = 1$). If nodes know their neighbors, a depth-first-tree approach leads to a broadcast time of $O(n)$ [1].

In our model nodes know the possible communication topology of the whole network, however they do not know their *effective* neighbors. For scenarios where nodes are aware of the actual topology, a message can be broadcast in asymptotically optimal time [8] with a randomized algorithm or deterministically [5], [12].

Finally, there is an interesting thread of research on the abstract MAC layer model: a model abstracting away low-level details such as signal propagation and contention. In [9], Ghaffari et al. study the dependence between the structure of unreliable links and achievable broadcast time complexity.

Acknowledgements

Stefan Schmid is supported by the DAAD-PHC PROCOPE program, and is part of the INP visiting professor program. Gilles Tredan is supported by the DAAD-PHC PROCOPE program.

References

[1] B. Awerbuch. A new distributed depth-first-search algorithm. *Information Processing Letters*, 20(3):147–150, 1985.

[2] R. Bar-Yehuda, O. Goldreich, and A. Itai. On the time-complexity of broadcast in multi-hop radio networks: An exponential gap between determinism and randomization. *J. Comput. Syst. Sci.*, 1992.

[3] L. Ben Saad, C. Chauvenet, B. Tourancheau, et al. Simulation of the RPL Routing Protocol for IPv6 Sensor Networks: Two Cases Studies. In *SENSORCOMM*, 2011.

[4] G. Bumiller. Power-line Physical Layer Emulator for Protocol Development. In *ISPLC*, 2004.

[5] F. Cicalese, F. Manne, and Q. Xin. Faster centralized communication in radio networks. In *Algorithms and Computation*. 2006.

[6] A. Clementi, A. Monti, and R. Silvestri. Selective families, superimposed codes, and broadcasting on unknown radio networks. In *Proc. 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 709–718. Society for Industrial and Applied Mathematics, 2001.

[7] A. Czumaj and W. Rytter. Broadcasting algorithms in radio networks with unknown topology. In *Journal of Algorithms*, pages 492–501, 2003.

[8] L. Gasieniec, D. Peleg, and Q. Xin. Faster communication in known topology radio networks. In *PODC*, 2005.

[9] M. Ghaffari, E. Kantor, N. Lynch, and C. Newport. Multi-message broadcast with abstract mac layers and unreliable links. In *Proc. ACM Symposium on Principles of Distributed Computing (PODC)*, pages 56–65, 2014.

[10] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke. Smart grid technologies: communication technologies and standards. *IEEE transactions on Industrial informatics*, 2011.

[11] D. Kowalski and A. Pelc. Broadcasting in undirected ad hoc radio networks. *Distributed Computing*, 18(1):43–57, 2005.

[12] D. R. Kowalski and A. Pelc. Optimal deterministic broadcasting in known topology radio networks. *Distributed Computing*, 2007.

[13] H. Liu, J. Song, B. Zhao, and X. Li. Channel study for medium-voltage power network. In *ISPLC*, 2006.

[14] E. Malacasa and G. Morabito. Characterization of PLC Communication Channel: a Networking Perspective. In *WSPLC*, 2009.

[15] A. Patel, J. Aparicio, N. Tas, M. Loiacono, and J. Rosca. Assessing communications technology options for smart grid applications. In *IEEE SmartGridComm*, 2011.

[16] D. Peleg. Time-efficient broadcasting in radio networks: a review. In *ICDCT*, 2007.

[17] Y.-A. Pignolet, I. Rinis, D. Dzung, and A. Karaagac. Multi-Interface Extensions for PLC / Wireless Simulator. In *WSPLC*, 2012.

[18] Z. Tao, Y. Xiaoxian, Z. Baohui, C. Jian, Y. Zhi, and T. Zhihong. Research of noise characteristics for 10-kv medium-voltage power lines. *IEEE Transactions on Power Delivery*, 2007.

[19] F. Versolatto and A. Tonello. Analysis of the PLC Channel Statistics using a Bottom-up Random Simulator. In *ISPLC*, 2010.