INFERENCE OF WIRED NETWORK TOPOLOGY USING MULTIPOINT REFLECTOMETRY

Michael Ulrich and Bin Yang

Institute for Signal Processing and System Theory
University of Stuttgart, Germany
Email: {michael.ulrich, bin.yang}@iss.uni-stuttgart.de

ABSTRACT

We present in this paper a novel algorithm CoMaTeCh for the inference of wired network topology using reflection measurements at multiple cable ends. This is useful for applications where the topology of an existing wired network (e.g. communication networks, powerline networks) is unknown and needs to be reconstructed in a non-intrusive way. Starting with the range and amplitude measurements of reflections caused by impedance discontinuities of the network, our algorithm estimates both the topology and the cable lengths. Using multiple reflection measurements, many ambiguities can be resolved, leading to a unique solution and a low computational effort. It is superior to existing approaches and is tested with both simulated and real data.

***Index Terms*—** Network topology inference, reflectometry, CoMaTeCh, communication networks, smart grid

1. INTRODUCTION

Determining the network structure of power grids to enable power line communication is just one example where information on the network topology is heavily desired. The transfer function of the communication channels strongly depends on the topology and cable lengths [1, 2]. Furthermore, cable length and topology information can be used in diagnostics to detect cable faults in automation systems [3] comparing the reconstructed network to the original one.

We consider the reconstruction of network structure from reflection measurements at the cable ends of a wired network. Ahmed and Lampe suggested an algorithm for this purpose that is based on a single reflection measurement at only one measurement point [4]. It has a low measurement effort, but suffers from a vast computational effort, because the number of possible solutions increases exponentially with the number of measured reflections.

The rooted neighbor-joining algorithm (RNJA) [5], originally developed for higher-level routing applications, was recently proposed to reconstruct the network topology [6, 7]. Its computational complexity is low and the solution is unique, but the algorithm requires distance measurements between all pairs of cable ends of the network. This causes a high measurement effort and is not feasible in many cases.

Hence, an algorithm using measurements from only a few points, but leading to a unique solution is highly desired. Such an algorithm is presented in this paper. It requires at least two reflection measurements and is able to easily identify the core network connecting these measurement points. Cable branches outside this core network are reconstructed in an iterative manner.

This paper is organized as follows: Section 2 reviews reflections in a wired network. Our algorithm CoMaTeCh for the inference of topology is presented in section 3 and evaluated in section 4. Section 5 concludes the paper.

2. REFLECTIONS IN A WIRED NETWORK

According to the transmission line theory, reflections arise at medium discontinuities (cable ends and branch points). The ratio of the reflected wave V_r and incident wave V_i is called the reflection coefficient Γ and is given by

$$\Gamma = \frac{V_r}{V_i} = \frac{Z_b - Z_0}{Z_b + Z_0}. \tag{1}$$

Z_0 is the characteristic impedance of the wire of the incident wave and Z_b is the impedance of the network seen at the discontinuity point. In general, Z_b is a function of the complete network following this discontinuity point. The ratio of the transmitted wave V_t and the incident wave V_i is called the transmission coefficient $T = \frac{V_t}{V_i} = 1 + \Gamma$. Additionally, the waves are damped by the line attenuation, which is approximately proportional to the reflection range.

With Eq. (1), we can simulate a reflection measurement. Starting with an incident wave at the measurement point, the wave is damped as it travels along the wire. It splits as it arrives at a branch point and is damped by the reflection and transmission coefficient, respectively. The resulting waves are treated likewise in a recursive manner, until the amplitude of the waves decays below a minimum level to be measured. The range and amplitude of waves are recorded when the waves return to the measurement point.

First published in the Proceedings of the 23rd European Signal Processing Conference (EUSIPCO-2015) in 2015, published by EURASIP

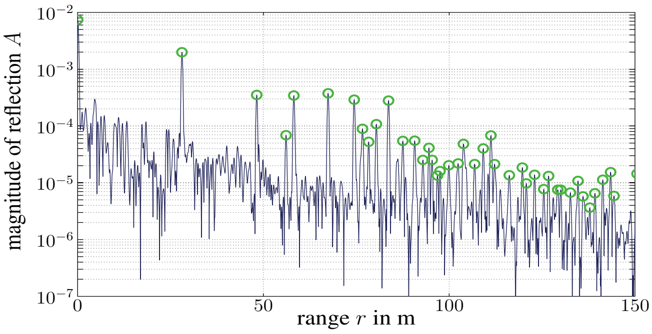


Fig. 1: Example of a reflection measurement. The reflections (marked with circles) are detected using an OS-CFAR detector.

Input: Reflections measured at several cable ends

Determine range and amplitude of reflections
Determine the core graph G_{core} connecting all measurement points (*Connect*)
$G_{\text{curr}} = G_{\text{core}}$
while $R_{rem} \neq \emptyset$ **do**
  Determine all reflections R_{rem} that can not be explained by G_{curr}
  Analyze R_{rem} to find a new node (*Map*)
  Determine all possible graphs (*Test*)
  Evaluate these graphs and return the graph with the lowest cost G_{choose} (*Choose*)
  if *cost*(G_{choose}) $<$ *cost*(G_{curr}) **then**
    $G_{\text{curr}} = G_{\text{choose}}$
  else
    Delete the reflections that caused *Map* to find the new node from R_{rem}
  end
end

Table 1: Overview of the CoMaTeCh algorithm

In practical applications, frequency domain reflectometry (FDR) is often preferred over time domain reflectometry (TDR), as FDR provides a higher signal-to-noise ratio and resolution [8–10]. Fig. 1 shows an example of such a FDR measurement using a vector network analyzer. It shows the magnitude A of the reflections at the measurement point as a function of the range r.

3. COMATECH ALGORITHM

3.1. Overview

Starting with the reflection measurements at several cable ends, our algorithm CoMaTeCh (**Co**nnect-**Ma**p-**Te**st-**Ch**oose) tries to find a graph representing the topology of the network as well as the length of all cable branches.

We represent a wired network as a weighted graph $G = \{V, E, W\}$. The set of nodes V represents all cable ends and branch points. The set of edges E represents all cable branches. The edge weights in W correspond to the lengths of all branches.

Like in [4–7], we assume a tree topology for the network, which does not contain any loops. This is necessary for CoMaTeCh to get a unique solution. This assumption is valid for low voltage grids [11] and most communication channels. For the simulation of reflections of a given network as it will be used in both CoMaTeCh experiments in 4., we approximate the impedance Z_0 in Eq. (1) by Z_0/N where N is the number of transmission branches for the current branch point. The underlying simplification is that we consider only the local neighborhood of the branch point and that all N branches have the same characteristic impedance Z_0. For the calculation of Γ at the cable ends (leaf nodes), the load impedance must be known. We assume that the load impedance is either much larger than Z_0 (or open) or much smaller than Z_0 (or short), resulting in $|\Gamma| \approx 1$.

Table 1 gives an overview of the CoMaTeCh algorithm. In the first step, the range and amplitude of reflections are extracted from the measurements. Next, the so called core graph connecting all measurement points is identified. Further nodes of the graph are added to the core graph in an iterative manner. To justify all reflections that cannot be explained by the graph of the current iteration, different extensions of the core graph are studied and rated.

In the following, we describe the major ideas of these steps in details. The algorithmic details are omitted due to limited space.

3.2. Detection of reflections

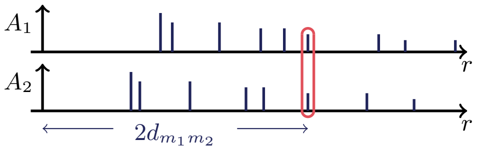
The first step is to detect the peaks in the reflection measurements to distinguish between reflections and noise. This is a nontrivial task, since the measurement shows many peaks in a varying noise level, see Fig. 1. We applied the ordered statistics constant false alarm rate (OS-CFAR) detector known from radar [12] to detect the peaks by using a dynamic threshold. For the estimation of the range and amplitude, we use parable interpolation. The detected peaks are marked with circles in Fig. 1.

3.3. Connect step

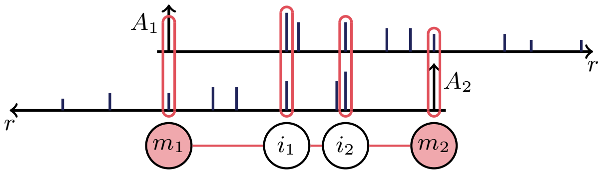
The step *Connect* identifies the so called core graph G_{core} connecting all measurement points. A core graph contains all connecting paths between each pair of measurement points.

The reflections measured at one pair of measurement points m_1 and m_2 are evaluated by comparing their reflection range. A wave coming from m_1, reflected at m_2 and then recorded at m_1 travels the same path as a wave from m_2, reflected at m_1 and recorded at m_2, just in the opposite direction. Hence, the distance $d_{m_1 m_2}$ between these two measurement points m_1 and m_2 is easily found as half of the range of the first common reflection in both measurements, see Fig. 2a. For this comparison, we introduce the tolerance δ_r for range and δ_a for amplitude in order to cope with measurement inaccuracy.

First published in the Proceedings of the 23rd European Signal Processing Conference (EUSIPCO-2015) in 2015, published by EURASIP

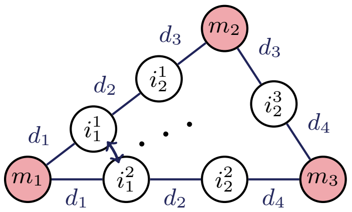


(a) The distance between two measurement points is determined by finding the first common reflection.

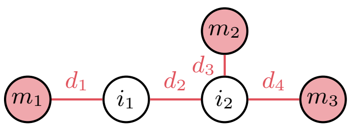


(b) One measurement is mirrored and shifted by $2d_{m_1 m_2}$. Now all common reflections correspond to nodes of the connecting path.

Fig. 2: Step Connect: The connecting path between two measurement points is detected.



(a) Starting from all connecting paths, edges with the same weight (cable length) are merged.



(b) The core graph after merging.

Fig. 3: Step Connect: All connecting paths are merged into one graph.

To find all nodes on the path between m_1 and m_2, one measurement (here m_2) is mirrored and shifted by $2d_{m_1 m_2}$. Now, the connecting path between m_1 and m_2 is seen from the same side and the common reflections in both measurements correspond to the nodes on the path, see Fig. 2b.

These connecting paths between each pair of measurement points are merged to the core graph G_{core} (Fig. 3b) that connects all measurement nodes. This is done by successively merging edges of the same weight (cable length), starting from the measurement nodes, see Fig. 3a.

3.4. Map step

In the step *Map*, all reflections that cannot be explained by the current graph G_{curr} are determined. The initial value of G_{curr} is the core graph from the previous step. We simulate the reflections for each measurement point based on the reflection model in section 2 and compare them to the measured ones.

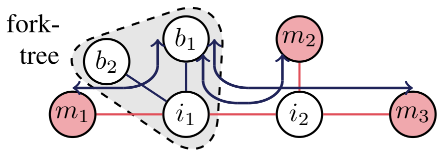


Fig. 4: Step Map: The node b_1 is added to the core graph, if reflections with proper distances to the measurement points m_i are found.

All measured reflections that are not within the tolerance (δ_r, δ_a) of the simulated ones are added to the set of remaining reflections R_{rem}.

The reflections in R_{rem} are used to detect further nodes of the network. In Fig. 4, a node b_1 is added to the existing node i_1, if R_{rem} contains reflections with the proper distances to the measurement nodes m_1, m_2 and m_3. To find such nodes, we compare the reflections in R_{rem} from all measurement points with each other. We call i_1 the root node of a new fork-tree growing from i_1.

The new node b_1 is only a hypothesis. It is validated by *Test* and *Choose*.

3.5. Test step

We are assuming that two additional nodes b_1 and b_2 with the distances $d_{i_1 b_2} > d_{i_1 b_1}$ have to be connected to i_1 in Fig. 4 in order to justify the reflections in R_{rem}. In the step *Map*, only distances to the root nodes are examined, but not the topology of the fork-tree added to i_1. In fact, b_2 can be connected directly to i_1 with the distance $d_{i_1 b_2}$ as in Fig. 4, or to b_1 with the distance $d_{b_1 b_2} = d_{i_1 b_2} - d_{i_1 b_1}$.

In [4], the whole network is one large fork-tree, resulting in a large number of possible topologies. In CoMaTeCh, the number of possibilities and hence the computational effort is only growing with the number of nodes in that one fork tree and is hence significantly lower. In order to find out which topology of the fork-tree best explains the measured reflections, all possible topologies of the fork-tree are determined in this step.

3.6. Choose step

In the step *Choose*, all above candidate graphs are compared and that one with the lowest cost is returned. This is done by again simulating the reflections of the current graph G_{curr} at all measurement points and comparing them with the measured ones. For this purpose, we assume a Gaussian mixture model (GMM)

$$A(r) = \sum_{n=1}^{N} \frac{a_n}{\sqrt{2\pi\delta_r^2}} \exp\left(-\frac{(r-r_n)^2}{2\delta_r^2}\right) \tag{2}$$

where all N reflections with range r_n and amplitude a_n are replaced by Gaussians. This is not a real pdf, as its integral is nor 1, but motivated by the assumed density of every single reflection. We use Eq. (2) to model both the measured reflection

First published in the Proceedings of the 23rd European Signal Processing Conference (EUSIPCO-2015) in 2015, published by EURASIP

at the m-th measurement point $A_m^{\text{meas}}(r)$, where r_n and a_n are from the detection step in 3.2, and the simulated reflection $A_m^{\text{sim}}(r, G)$ for a candidate graph G, where r_n and a_n are extracted from the reflection simulation for G. The best graph G is determined by minimizing the squared difference between the simulated $A_m^{\text{sim}}(r, G)$ and measured $A_m^{\text{meas}}(r)$ over all measurement points:

$$\min \sum_{m=1}^{M} \int_0^{\infty} \left(A_m^{\text{meas}}(r) - A_m^{\text{sim}}(r, G)\right)^2 dr \tag{3}$$

As this is a discrete optimization task, the cost of all possible topologies must be calculated and compared. The graph with the lowest cost is used for the next iteration.

3.7. Discussion

Now we briefly discuss some properties of CoMaTeCh. The number of measurement points has a great impact on the performance of CoMaTeCh. If the number of measurements is too small, reflections may match by accident, causing CoMaTeCh to place wrong nodes. This is mostly prevented by the step *Choose*, but may still occur.

The computational effort of CoMaTeCh depends on the size of the largest fork-tree. It determines how many candidate graphs must be evaluated in *Test* and *Choose*. Therefore, a larger core graph causes smaller fork-trees and results in a lower computational effort. This results to the rule that the measurement points should be selected as far distant as possible to achieve a large-size core graph.

4. EXPERIMENTS

4.1. Evaluation criteria

For the discussion of the test results in this section, we use two criteria. Dependent on the application, both criteria can be important.

The first score α_c describes the percentage of networks which are reconstructed completely in both topology and cable length. The second score α_s measures the overall similarity between the original graph G_1 and the reconstructed graph G_2. It expresses to which degree the network is reconstructed. As in [13], we calculate the maximum common subgraph (mcs) in both topology and cable length and compare its order to the order of G_1 and G_2.

$$\alpha_s(G_1, G_2) = \frac{|\operatorname{mcs}(G_1, G_2)|}{\max(|G_1|, |G_2|)} \tag{4}$$

The order $|G|$ of a graph denotes the number of its nodes.

For large fork-trees, the computational complexity increases. We are currently using a MATLAB-implementation of CoMaTeCh that is not yet optimized for runtime. Therefore, we define a timeout of 30 minutes, after that CoMaTeCh

test number	# nodes	# measurement points	tolerance δ_r (times average edge length)	σ_r of noise (times average edge length)	$\alpha_{c,tot}$	$\alpha_{c,fin}$	$\alpha_{s,tot}$	$\alpha_{s,fin}$
1	8	2	$6{\cdot}10^{-5}$	0	78.2%	99.3%	95.6%	99.3%
2	8	2	$6{\cdot}10^{-5}$	$1{\cdot}10^{-5}$	64.7%	90.2%	88.8%	94.9%
3	14	2	$6{\cdot}10^{-5}$	$1{\cdot}10^{-5}$	2.3%	5.0%	58.4%	29.5%
4	14	4	$6{\cdot}10^{-5}$	$1{\cdot}10^{-5}$	26.5%	45.2%	79.2%	72.9%

Table 2: Test results with simulated reflections

is interrupted and the current graph is returned as the result. In order to reflect the timeouts, $\alpha_{s,fin}$ and $\alpha_{c,fin}$ denote the above defined scores by considering only finished reconstructions without timeouts. $\alpha_{s,tot}$ and $\alpha_{c,tot}$ consider all runs, also those interrupted by the timeout.

4.2. Experiment with simulated reflections

In the first experiment, we use simulated reflections for CoMaTeCh. For this purpose, we generate 1000 random networks in each test for a given number of nodes and measurement points. We calculate the average of the scores $\alpha_{c,to}$, $\alpha_{c,fin}$, $\alpha_{s,to}$ and $\alpha_{s,fin}$. The results are summarized in Tab. 2.

In the experiments 2-4, we manipulate the simulated reflections by adding Gaussian distributed noise $\mathcal{N}(0, \sigma_r^2)$ with the standard deviation σ_r to each reflection range. The tolerance δ_r of CoMaTeCh is chosen as $6\sigma_r$, as almost all (99.73%) of the values are within the range $\pm 3\sigma$. The amplitude is multiplied by a truncated $\mathcal{N}(1, \sigma_a^2)$ factor which is always larger than a positive threshold. We use $\delta_a = 1.8$ for the amplitude tolerance. For a comparison, we use the same tolerances in test 1.

In the 1st test, we analyze networks consisting out of 8 nodes. A full reconstruction of the network is reached in almost all cases. In the 2nd test, the measurement noise lowers the reconstruction scores, but still most topologies can be reconstructed. We increase the number of nodes to 14 in the 3rd test, while still using only two measurements. As the measurement points are chosen randomly, the core graphs are small, the size of fork-trees and thus the computational complexity are high and most simulations (89%) are stopped by the timeout. Furthermore, many reflections match by accident, causing CoMaTeCh to add fictitious nodes. The 4th test increases the number of measurements to 4. The core graphs become larger, the fork-trees smaller and therefore the scores higher.

A comparison to the algorithm of Ahmed and Lampe [4] is impossible because that algorithm returns over $8! > 40,000$ possible solutions even for test 1. The RNJA algorithm from [5] is not applicable here, because it requires measurements at all cable ends.

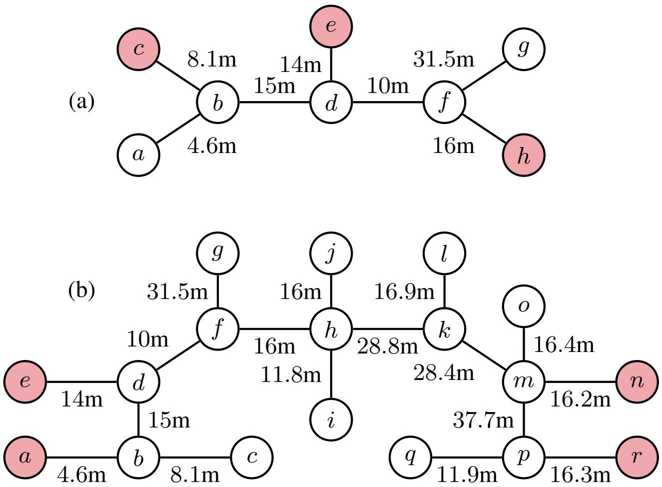


Fig. 5: Two networks under test. FDR measurements are carried out at the highlighted cable ends

4.3. Experiment with measured reflections

Two real networks with cables of type RG58U are tested in this experiment. The first network in Fig. 5a consists of 8 nodes. FDR measurements are collected at three nodes c,e, and h. If we use all three or only two of them (c, e) and (e, h), the complete network is perfectly reconstructed. If we use the two measurements at (c, h), the network is also recovered, but with two additional branches.

The second network in Fig. 5 consists of 18 nodes and has a total length of roughly 300 m. In this case, the network analyzer could not register reflections from all nodes of the network due to increased attenuation. As CoMaTeCh needs at least one reflection of a node to detect it, the complete network could not be reconstructed. We simulated reflections for this network at the nodes a, e, n and r in order to guarantee the measurement of at least one reflection from each node and then applied CoMaTeCh. The network could then be completely reconstructed using these 4 measurements. This shows that not CoMaTeCh, but rather the reflection measurement is the bottleneck of the system.

One possible improvement for the future is to enhance the reflection measurement to even penetrate a large network. Another future work is to use CoMaTeCh to identify different overlapping parts of a network and to assemble them to the complete network.

5. CONCLUSIONS

In this paper, a new algorithm for the reconstruction of cable network topology from reflection measurements is proposed. By utilizing measurements at multiple cable ends, we obtain a fast and reliable algorithm requiring only a low measurement effort. The performance of this algorithm is confirmed by tests with both simulated and measured data.

REFERENCES

[1] Tooraj Esmailian, Frank R. Kschischang, and P. Glenn Gulak, “In-building power lines as high-speed communication channels: Channel characterization and a test channel ensemble,” *International Journal of Communication Systems*, vol. 16, pp. 381–400, 2003.

[2] Petr Mlynek, Jiri Misurec, Martin Koutny, and Pavel Silhavy, “Two-port network transfer function for power line topology modeling,” *Radioengineering*, vol. 21, pp. 356–363, 2012.

[3] Mostafa Kamel Smail, Lionel Pichon, Marc Olivas, Fabrice Auzanneau, and Marc Lambert, “Detection of defects in wiring networks using Time Domain Reflectometry,” in *IEEE Transactions on Magnetics, Vol. 46, No. 8, August 2010*, 2010.

[4] Mohamed Osama Ahmed and Lutz Lampe, “Power line network topology inference using Frequency Domain Reflectometry,” in *IEEE ICC 2012 - Selected Areas in Communications Symposium*, 2012, pp. 3419–3423.

[5] Jian Ni and Sekhar Tatikonda, “Network tomography based on additive metrics,” Tech. Rep., Department of Electrical Engineering, Yale University, New Haven, CT, USA, 2008.

[6] Mohamed Osama Ahmed and Lutz Lampe, “Power line communications for low-voltage power grid tomography,” in *IEEE Transactions on Communications, Vol. 61, No. 12, December 2013*, 2013.

[7] Lutz Lampe and Mohamed O. Ahmed, “Power grid topology inference using power line communications,” in *IEEE SmartGridComm 2013 Symposium - Communication Networks for Smart Grids and Smart Metering*, 2013.

[8] David Dodds, Muhammad Shafique, and Bernardo Celaya, “TDR and FDR identification of bad splices in telephone cables,” in *IEEE CCECE/CCGEI*, 2006.

[9] David Dodds, “Single-ended FDR to locate and specifically identify DSL loop impairments,” in *ICC*, 2007.

[10] Bernard Celaya and David E. Dodds, “Single-ended DSL line tester,” in *CCECE 2004- CCGEI 2004, Niagara Falls, May 2004*, 2004.

[11] Olaf G. Hooijen, “On the relation between network-topology and power line signal attenuation,” Tech. Rep., Signaal Communications, Huizen (NL), 1998.

[12] Hermann Rohling, “Radar CFAR thresholding in clutter and multiple target situations,” in *IEEE Transactions on Aerospace and Electronic Systems*, 1983.

[13] Horst Bunke and Kim Shearer, “A graph distance metric based on the maximal common subgraph,” *Pattern Recognition Letters*, vol. 19, pp. 255–259, 1998.

First published in the Proceedings of the 23rd European Signal Processing Conference (EUSIPCO-2015) in 2015, published by EURASIP