Distributed Network Tomography Applied to Stochastic Delay Profile Estimation

Jakub KOLAR [1], Jan SYKORA [1], Umberto SPAGNOLINI [2]

[1] Dept. of Radioelectronics, Czech Technical University in Prague, Technicka 2, Prague 6, Czech Republic
[2] Dip. di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, Milan, Italy

{kolarj39, Jan.Sykora}@fel.cvut.cz, Umberto.Spagnolini@polimi.it

Submitted September 2, 2019 / Accepted January 14, 2020

Abstract. *In this paper is shown, how delay properties of the edges of a network with stochastic properties can be estimated cooperatively by individual nodes that retain the delay profiles of the entire network. The proposed algorithm adopts null-space projection-based consensus among agents to find individual entries from a set of arbitrary sum-cumulative entities associated with graph edges (e.g., delays associated with edges) based on sums over the network paths. The local estimates of delay profile are estimated using Least Squares (LS). A modified, tailored, iterative consensus algorithm is then employed to distribute information among the neighbors. The distributed network tomography is compared to the conventional centralized solution and also to iterative solvers based on Cimmino, CAV, and Landweber methods applied in a distributed manner.*

Keywords

Network tomography, network delay profile, distributed consensus algorithm, projection-based consensus

1. Introduction

Network tomography [1], [2] is a broad area of methods determining the network characteristics (delays, edge traffic, node processing, etc.) from a set of observable quantities that are sum of local (edge or node) properties (see [3] for comprehensive collection of applications and tools). Large and dense networks with highly dynamic and stochastic node and edge behavior (e.g., the link connectivity, processing load at the node, etc.) present a significant challenge. A centralized solution based on the collection of all measurements from all nodes toward a fusion to solve a set of linear equations has unmanageable extensive signaling overhead. Distributed node-based solutions would be preferable and guarantees to each node to infer the properties of the entire network even if they are not directly observable by each node. These facts motivate our work. The main objective of this paper is to provide such a distributed and node-based algorithm. These fundamental principles make the solution robust and resistant to malfunctions.

Given the size of the inverse problem, one aspect of network tomography problems is to include the projection mechanism to match the null spaces of the solutions [4]. An exhaustive work on distributed solving of sets of linear equations is in [5], but these are only deterministic settings. In [6], the authors address decentralized consensus in distributed networks, however, the model described therein is not stochastic as in our case. In [7] is presented a solution to a problem of parameter estimation in time-varying network topology using a consensus algorithm. However, the system model therein is different, as well as the approach of the solution, which is based on the sub-gradient method. The authors of [8] considered a problem of distributed optimization on a graph using the consensus algorithm, where the objective function was a sum of convex-functions. Further work on the topic of multi-agent optimization may be found in [11].

The problem of consensus estimation on time variable networks was also addressed in [9] and [10], but, again, therein the model does not take into account stochastic properties of the measured quantity and variable is the topology. For recent review work focused on utilization in wireless sensor networks, see [12].

The authors of [13] presented the concept of predicting overall network metrics from measuring only a subset of the paths in the network graph, addressing the issue of huge measurement overhead with a motivation of statistical kriging. Their framework is based on the analysis of matrix carrying possible paths in the graph, referred there as a routing matrix, which is also fundamental to our work but does not consider the distributed solution. Based on this is in [14] developed a kriged Kalman filter approach that on-line selects the paths to estimate delays and presented results on real-world data.

In large networks with a high number of edges, any estimate of the property of individual edges of the entire network would make the signaling diverge if made it centrally. The distributed method confines the estimates to the interactions among neighboring connected nodes. More specifically, let the nth node collect a set of cumulative entities $u_n = \sum_{m\in\mathcal{E}_n} T_m$ over the set of edges $\mathcal{E}_n$, where $T_m = \tau_m + w_m$

is a general property at mth edge, with deterministic (τ_m) and zero-mean stochastic (w_m) part. The goal is to let *each* node to infer individually the *whole* set $\{\tau_m\}_{m=1}^{M}$ from the cumulative values u_n obtained at nth node over the randomly chosen set of paths $\mathcal{E}_n$ with the help of a distributed cooperation with other nodes.

Note, that in the sequel, the edge delay property was selected for a purpose of clearness and also many other real-world choices are available, such as channel coefficients estimation. Using the edge network delay profiles as a distributed network tomography example, the goal is to let every node solve part of the whole inverse problem to estimate the edges delays T_m from a set of cumulative values u_n notwithstanding the rank-deficiency of the linear system by every node based on the randomly chosen path $\mathcal{E}_n$. E.g., power-saving and additional relaxing of the signaling overhead are reasons to use only a subset $\mathcal{E}_n$ of all the paths. Further, the network does not behave deterministically and edge properties (delays) are random and change in every single observation of the edge property.

Contribution of the paper is a distributed network tomography from cumulative stochastic values that solves a set of undetermined systems that is numerically shown to converge to the edge-properties estimated by a centralized system. We also compare the solution with other methods stated in [15], such as Cimmino method [16], Component Averaging [17] and Landweber method [18], that are known to be used in tomography tasks. Even if the exemplary application is for edge delays, the algorithm is generally applicable for any cumulative stochastic quantities. More complex tasks to be addressed are, e.g., channel estimation, synchronization, etc.

The rest of this paper is organized as follows. Section 2 states the problem, defines the system model and introduces notation. Section 3 contains a detailed description of the proposed algorithm. Section 4 evaluates the properties of the algorithm, numerically demonstrates its convergence, and compares the results with selected reference methods. Section 5 contains the paper conclusion.

2. System Model

2.1 Network

Let a network be modeled as a graph with N nodes where all the M edges are sequentially numbered by $m \in \{1, M\}$, the delay $T_m \in \mathbb{R}_0^+$ is the stochastic edge property due to propagation and node-dependent processing [1]. At rth observation epoch ($r \in \mathbb{N}$), it is $T_m(r) = \tau_m + w_m(r)$, where the zero-mean stochastic fluctuation $w_m(r)$ is the delay jitter that is assumed as independent and identically distributed (IID) and the edge delay τ_m is constant over all observation epochs.

The goal is to get the estimates $\hat{\tau}_{n,m}$ for all $m \in \{1, \ldots, M\}$ to be available by each network node $n \in \{1, \ldots, N\}$ and, after a number of node-to-neighbors epochs is large enough, it should hold $\hat{\tau}_m = \hat{\tau}_{n,m}$, $\forall n$. The ensemble $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_M]^{\mathrm{T}}$ is the *network delay profile*. The distributed algorithm is iterative and reaches a consensus as the number of epochs $r \to \infty$. However, the algorithm runs for a selected finite number of epochs R to reach the consensus on the estimate $\hat{\boldsymbol{\tau}}$ of the vector $\boldsymbol{\tau}$ and, when the consensus is reached, all the nodes achieve the final estimates $\bar{\boldsymbol{\tau}}_n(R)$. To simplify, we adopted a Gaussian IID sets $T_m(r) \sim N(\tau_m, \sigma_w^2)$ with $\tau_m \gg \sigma_w$.

2.2 Observation Model

Within any epoch r, each node randomly floods the network with probe excitations (e.g., a modification of "ping" when considering the Internet protocol) over randomly chosen set of target nodes. In our particular case of delay profile estimation, these probes are designed to accumulate the delays of all the edges as a single value, i.e., a simple sum. (Note, for other edge properties, the collecting operation shall be different, such as a sum of logarithms of the channel gains.)

As a consequence, each node receives at rth observation period a random set of K observations from many source nodes over many (and possibly overlapping) paths. The received observation must contain only an identification of the edges over which the probe traveled and total cumulated delay over that path.

The nth node receives K observations at the rth observation epoch

$$\mathbf{u}_n(r) = \left[u_n^{(1)}(r), \ldots, u_n^{(K)}(r)\right]^{\mathrm{T}}. \tag{1}$$

In the different observation epochs, the number of the received observations K in node n need not be the same, but for the sake of simplicity, we assume it to be constant. The kth observed cumulated delay at node n is associated with the path over the set of edges, denoted for nth node as $\mathcal{E}_n^{(k)}(r)$. This information is obtained from the list of traversed edges in the received probe ("ping"):

$$u_n^{(k)}(r) = \sum_{m \in \mathcal{E}_n^{(k)}(r)} T_m(r) = \sum_{m \in \mathcal{E}_n^{(k)}(r)} \left(\tau_m + w_m(r)\right). \tag{2}$$

Let the $(K \times M)$ matrix $\mathbf{H}_n(r)$ be the path-indexing at nth node and rth epoch such that the kth row and the mth column entry $H_{n,km}(r) \in \{0, 1\}$ indicates the presence of the mth edge on the $\mathcal{E}_n^{(k)}(r)$ path

$$H_{n,km}(r) = \begin{cases} 1; & \text{if } m \in \mathcal{E}_n^{(k)}(r), \\ 0; & \text{elsewhere.} \end{cases} \tag{3}$$

The cumulated delay observation is thus

$$\mathbf{u}_n(r) = \mathbf{H}_n(r)\mathbf{T}(r) = \mathbf{H}_n(r)\,(\boldsymbol{\tau} + \mathbf{w}(r)) \tag{4}$$

where $\mathbf{T}(r) = [T_1(r), \ldots, T_M(r)]^{\mathrm{T}}$ is the vector of all random edge delays with $\mathbf{w}_n(r) = [w_1(r), \ldots, w_M(r)]^{\mathrm{T}}$. Even if the number of observations K is very large (say $K \to \infty$), the path-indexing matrix $\mathbf{H}_n(r)$ from nth node is typically row-rank deficient as there might be some edges never being sensed by the randomly selected paths $\{\mathcal{E}_n^{(k)}(r)\}_{k=1}^{K}$. Concretely, an abstraction of the "ping" protocol could be used to generate these probes. After arrival to each node, the payload shall be incremented with the edge delay value and the traversed edge label noted. It can be understood from the later description, that the number of nodes performing the measurement can be relaxed. A fraction of nodes could generate the probes in each epoch, as well as all of them.

3. Distributed Consensus Tomography

Each node employs a measurement procedure having the measurements $\mathbf{u}_n(r)$ and the path-indexing matrix $\mathbf{H}_n(r)$. The goal is for each nth node to reach a consensus on the estimate $\hat{\boldsymbol{\tau}}$ of the complete network delay profile $\boldsymbol{\tau}$. This estimation task is after a set of consensus-like iterations that are made complex by some peculiarities of the network tomography problem at hand: (1) each node observes only a limited subset of the edges via its measurements ("ping") and possibly only a small part of the entire network (i.e., likely some edges are being observed only by a subset of nodes); (2) the mth edge is affected by a random jitter $w_m(r)$, thus introducing a random error in cumulative measurements; (3) all edges need to be measured, hereby we assume a connected network and the existence of a unique (consensus) solution of $\hat{\boldsymbol{\tau}}$. In other words, the execution of a sufficient number of the epochs guarantees that all paths are randomly selected by any criteria external to the network tomography algorithm.

The network delay profile $\hat{\boldsymbol{\tau}}$ follows from the global consensus estimator that is reached in two phases. First, we perform a *local estimate* of the network delay profile, which takes into account the very limited accessibility of the sensed network edges. In other words, the local estimate is characterized by the high degrees of freedom that are weakly conditioned by a set of local observations that partially sense a subset of edges, and path-indexing matrix $\mathbf{H}_n(r)$ is row-rank deficient. This limitation is addressed by the second phase of *consensus* iteration, where all nodes align the weakly conditioned estimate of the path-indexing subspace by performing a consensus on the null-space of the local observation model.

3.1 Measurement and Local Estimate

Figure 1 outlines the procedure that is detailed below to clarify the steps and usage of variables. The first step is *measurement*: each agent "pings" some subset of the remaining agents, resulting in gathering $\mathbf{u}_n(r)$ and $\mathbf{H}_n(r)$.

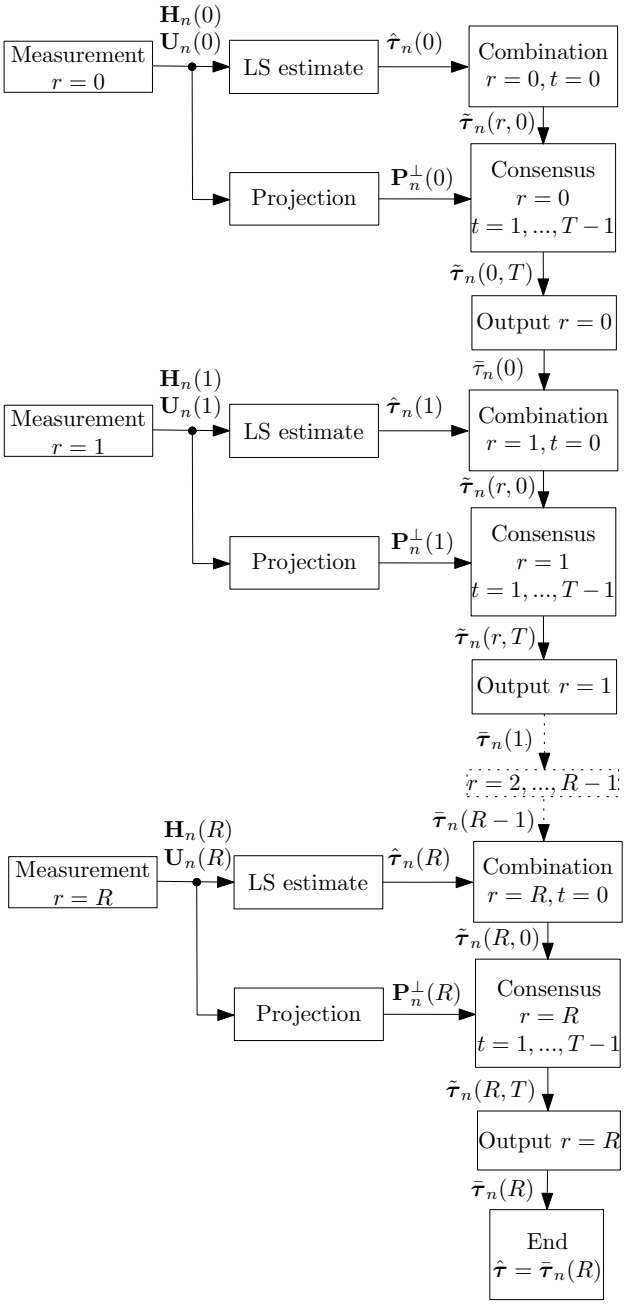

Fig. 1. Block diagram to demonstrate phases of the algorithm and corresponding equations.

See Fig. 1, the epoch variable r is incremented with each "Measurement" therein. The aim of this phase is to reach as much path-diversity as possible, such that the routes between source and destination node should be through as many different edges as possible. The more edges, the better, and also, the higher the number of different measurements shall imply faster convergence. More measurements of identical paths edges might be treated as a special case by simple averaging to reduce the equivalent jitter variance. The model (4) can be rewritten as

$$\mathbf{u}_n(r) = \mathbf{H}_n(r)\boldsymbol{\tau} + \mathbf{w}'(r) \tag{5}$$

where $\mathbf{w}'(r) = \mathbf{H}_n(r)\mathbf{w}(r)$ is an equivalent local observation noise. Equation (4) is a poorly conditioned observation model and the set of linear equations is underdetermined. Since the problem is heavily row-rank deficient, the initial estimator at each node should take that into account. One of the options here is to use *minimum-norm* LS solution [19] $\hat{\boldsymbol{\tau}}_n(r) = \mathbf{H}_n^{\mathrm{T}}(r)(\mathbf{H}_n(r)\mathbf{H}_n^{\mathrm{T}}(r))^{-1}\mathbf{u}_n(r)$. Other options, such as various constrained variants of LS, could be considered, as well.

3.2 Consensus on Null-Space Projection

The goal is to cooperatively reach a global consensus on those components of local estimator results that cannot be solved locally for the underdetermined system of equations. These components lie in the null-space of the local path-indexing matrices. A convenient approach described in [20] is the Accelerated Projection-Based Consensus. For the network tomography, the null-space projection matrix corresponding to nth node shall be defined as

$$\mathbf{P}_n^{\perp}(r) = \mathbf{I} - \mathbf{H}_n^{\mathrm{T}}(r)(\mathbf{H}_n(r)\mathbf{H}_n^{\mathrm{T}}(r))^{-1}\mathbf{H}_n(r) \tag{6}$$

where $\mathbf{I}$ is the identity matrix. The purpose of the orthogonal projection here is to update the estimates in the node n during the Consensus Phase only with data corresponding with the edges that were poorly (or not at all) observed by the node. These are in the null-space of the path-index matrix $\mathbf{H}_n(r)$.

The *consensus* iterations by each node are over the path-index null-spaces. For each observation period r, we perform a number of consensus update steps indexed by the variable t. See "Consensus" in Fig. 1, t variable counts consensus iterations. The consensus updates $\tilde{\boldsymbol{\tau}}_n(r,t)$ are

$$\tilde{\boldsymbol{\tau}}_n(r,t+1) = \tilde{\boldsymbol{\tau}}_n(r,t) + \epsilon\mathbf{P}_n^{\perp}(r)\left(\sum_{j\in\mathcal{N}_n}\frac{\tilde{\boldsymbol{\tau}}_j(r,t) - \tilde{\boldsymbol{\tau}}_n(r,t)}{|\mathcal{N}_n|}\right) \tag{7}$$

where $t \in \{0,1,...,T-1\}$, $\mathcal{N}_n$ denotes the set of neighbors of the nth node and ϵ is a selected loop gain, which choice is described e.g. in [21]. Note, complete graph topology need not be known in each node. After $t = T-1$ consensus updates within the rth period, one gets

$$\bar{\boldsymbol{\tau}}_n(r) = \tilde{\boldsymbol{\tau}}_n(r,T) \tag{8}$$

which highlights how the measurements by all the other nodes eased to reach the network delay estimate in the rth observation epoch.

The consensus update procedure at rth epoch takes into consideration the results of previous measurement epochs. The updates are initialized at $t = 0$ either by weighted combination of the previous period result or by a simple local estimate at $r = 0$. For $t = 0$, the weights are designed as running average on consensus

$$\tilde{\boldsymbol{\tau}}_n(r,0) = \begin{cases} \hat{\boldsymbol{\tau}}_n(0); & \text{for } r = 0 \\ \frac{1}{1+r}\hat{\boldsymbol{\tau}}_n(0) + \frac{r}{1+r}\bar{\boldsymbol{\tau}}_n(r-1); & \text{for } r \in \{1,...,R\} \end{cases} \tag{9}$$

where R is the total number of observation periods, denoted "Combination" in Fig. 1. The final resulting null-space consensus at the node n on all local estimates comprising all observation periods is $\bar{\boldsymbol{\tau}}_n(R)$. If the consensus is actually reached, all nodes have the result

$$\hat{\boldsymbol{\tau}} = \bar{\boldsymbol{\tau}}_n(R), \ \forall n \tag{10}$$

where vector $\hat{\boldsymbol{\tau}}$ is the resulting global consensus estimate of the network delay profile.

4. Numerical Results and Discussion

4.1 Reference Solution and Complexity

To adequately compare our solution with other approaches, we opted to compare our numerical results with: (1) centralized solution, where the fusion center obtains all the observations, and (2) hybrid distributed solutions. The metrics are: (1) the computational complexity, (2) the signaling complexity and (3) the convergence. The centralized reference solution is based as follows: all the N nodes perform within R epochs K measurements; then, all these KR measurements are sent to one fusion center; the overdetermined set of NKR linear equation is solved to estimate $\boldsymbol{\tau}$.

On the complexity, each node with K measurements solves in distributed scenario K equations with computation complexity $\mathcal{O}_{\mathrm{c}}(K^3)$, reusing part of the LS solution (compare (5) and (6)) computes the projection matrix with complexity $\mathcal{O}_{\mathrm{c}}(K^2)$. During T exchanges each node repeats the consensus phase with $\mathcal{O}_{\mathrm{c}}(T|\mathcal{N}_n|M^2)$ operations, where $\mathcal{O}_{\mathrm{c}}(M^2)$ stands for the projection matrix multiplication. The computation complexity is for nth node in R epochs $R \times (\mathcal{O}_{\mathrm{c}}(K^3) + \mathcal{O}_{\mathrm{c}}(T|\mathcal{N}_n|M^2))$ operations. The centralized solution needs $\mathcal{O}_{\mathrm{c}}((NRK)^3)$ operations (the additional operations to solve possible duplicities in the set are neglected). Further, in the distributed approach, the computation is distributed to all nodes, and this provides an additional benefit.

Concerning the amount of data to be transferred, in the distributed algorithm, the nth node transmits to its neighbors $\mathcal{N}_n$ its estimate $\tilde{\boldsymbol{\tau}}$ of M edge delays in R epochs and repeats T consensus phases. This sums up to $\mathcal{O}_{\mathrm{s}}(TRM|\mathcal{N}_n|)$ values to be signaled. On the contrary, in the centralized case, one needs to transfer all measurement results, which is $K(M+1)$ values per each node for $\mathbf{u}_n(r)$ and $\mathbf{H}_n(r)$. Once the estimate $\hat{\boldsymbol{\tau}}$ is computed, it should be distributed back to all nodes, and the overall signaling complexity is $\mathcal{O}_{\mathrm{s}}(K(M+1)R+M)$ transfers per node. The comparison is more complex in this case, and it varies in different scenarios (e.g., one could account for the accumulation of messages due to the routing from each node to the centralized processing node).

The analysis of the convergence of consensus in terms of iterations T needs to take into account the network connectivity as routinely employed in the consensus method and the redundancy exploited by the repeated "ping" sensing iterations.

A proportion of T and R, while keeping $TR = \text{const}$, can vary. A trade-off appears between the gain resulting from new observations (when increasing R), represented by new inputs of $\mathbf{u}_n(r)$ and $\mathbf{H}_n(r)$ and the rate of convergence inside one epoch, related to the value of T. Of course, the dynamics of the network and the delay jitter influence the results.

4.2 Reference Methods

The first method that we decided to use for comparison is Component Averaging (CAV) described in [17]. The CAV method was introduced as an iterative parallel technique suitable for large and sparse systems of linear equations, and this makes it a useful reference option. Instead of orthogonal projections with scalar weights, it uses oblique projections and diagonal weighting matrices [17].

The second used, also projection-based method, is known Cimmino method [22], introduced in [16]. The third used, Landweber method [18], is also a common choice for the tomography computations [17], and it is suitable for our system model since it was designed to be used with noisy measurements. As shown below, the Landweber method performed the best in simulations. Stopping rules need to be considered for these methods, e.g., limit the maximum number of iterations. Other methods are used for tomography tasks, e.g., Kaczmarz, ART, SART and DROP [23], however, we believe that the above three examples are satisfactory to demonstrate the performance of the described algorithm [17].

There are multiple choices on how to design the reference solutions. Here the design is as follows: In the initialization phase, each node computes the estimate based on its local knowledge of $\mathbf{u}_n(1)$, $\mathbf{H}_n(1)$ with the CAV, Cimmino, and Landweber methods, respectively. In the subsequent rth epochs, the nth node shares its measurements, represented by $\mathbf{u}_n(r)$ and $\mathbf{H}_n(r)$ with its neighbors $\mathcal{N}_n$. Furthermore, the nodes keep the history of their measurements. This can be straightforwardly written with an artificial, identity observation matrix of size M, and the estimate from the previous epoch $r-1$ is taken instead of the vector of the measured values. The reference solution is then obtained as a solution of the extended set of linear equations. Such an extended set then carries information about the actual measurement of the given node, measurements of all its direct neighbors, and the previous epochs. The numerical methods are then used to solve these extended sets of equations.

The advantage of this approach is a significant reduction in the size of the set of equations compared to the centralized approach. Calculations are distributed among all the nodes to avoid the centralized approach. However, it is important to note that the nodes need not reach the same solution of the edge property, which could be critical in some applications. This is inevitable without any form of consensus. Another drawback is a significant overload caused by transmitting the measurements between the neighbors. These are other reasons why the proposed method is preferable.

Note, the implementation of the methods in our simulations simultaneously uses the whole set of equations during the iterations, which ensures that the specific organization of the extended set is not essential.

4.3 Numerical Analysis

In this section, we present several results for various scenarios. In the distributed method, in every epoch r each node performs K measurements to obtain the estimate $\hat{\boldsymbol{\tau}}_n(r) = \mathbf{H}_n^{\mathrm{T}}(r)(\mathbf{H}_n(r)\mathbf{H}_n^{\mathrm{T}}(r))^{-1}\mathbf{u}_n(r)$, and processes T steps of (7) to reach the final consensus phase. In centralized approach, at every epoch r the ensemble of equations are NKr and augments by NK new sets at every subsequent epoch. To illustrate the level of convergence over the epoch r in term of root mean squared error per edge we use expression:

$$\mathrm{RMSE}(r) = \mathrm{E}\left[\sqrt{\frac{1}{N}\sum_{n=1}^{N}\frac{1}{M}\sum_{m=1}^{M}(\tau_m - \hat{\tau}_{n,m})^2}\right] \tag{11}$$

where E[.] is numerically approximated by averaging of ρ Monte Carlo runs of the algorithm with preserved settings and topology, but different realizations of jitter and routing. Also, the results computed by the centralized approach are provided for reference (neglecting possible traffic delay for measurement fusion).

Note, the simulation was implemented in MATLAB software (R2019a), on a server equipped with Intel® Xeon® CPU E5-2420 v2 @ 2.20GHz and 48 GB of RAM. MATLAB implementation of the algorithm is provided in [24] and utilizes software package [15]. No special MATLAB toolbox is required. The implementation is computationally quite demanding, especially for larger topologies, because of all the reference methods needed to be computed.

The first topology under test is a fully connected graph with 10 nodes, as shown in Fig. 2. To demonstrate the convergence, we provide the results for different values of measurements K. The parameters of the topology are: $\rho = 10$, $T = 500$, $R = 7$ and $\sigma_w^2 = 40$. This settings holds for all the provided examples, with an exception of value of T, as addressed explicitly, later. In Figs. 3 and 4, are shown results of RMSE according to (11) for $K = 18$ and $K = 72$, respectively.

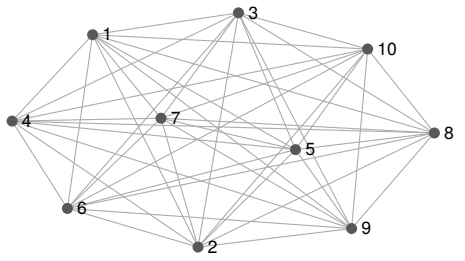

Fig. 2. Examined topology 1: Fully connected graph with $N = 10$.

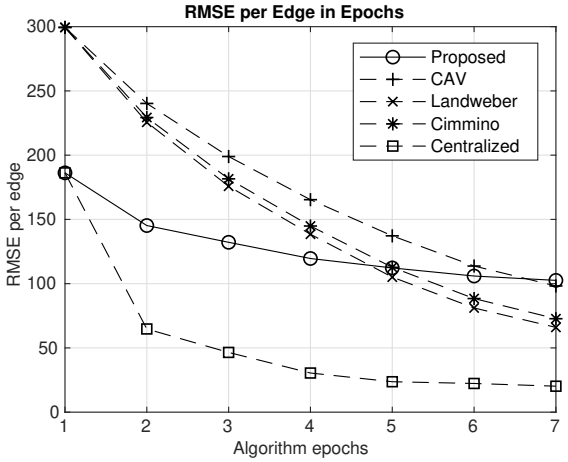

Fig. 3. Comparison of methods for $K = 18$, $N = 10$, $T = 500$.

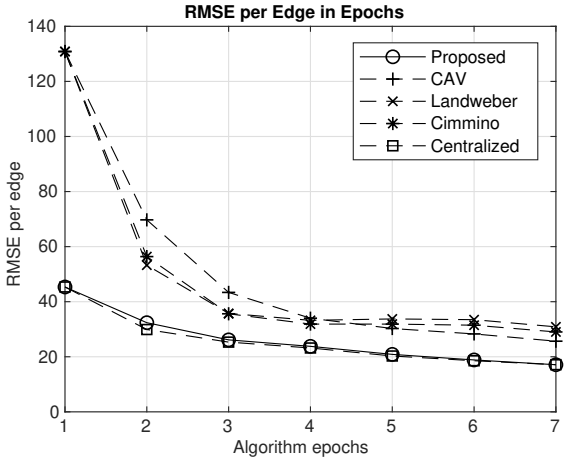

Fig. 4. Comparison of methods for $K = 72$, $N = 10$, $T = 500$.

In these figures, we can easily compare the performance of the different approaches. For all the scenarios, the Centralized approach reached the fastest convergence behavior. This is not surprising since it has access to all the measurements of all the nodes at the same time, but this comes at the price of enormous overhead to transmit the measurements to the fusion center, and the size of the resulting set of equations is enormous. The Cimmino method was used to solve it. Next, the reference methods Cimmino, CAV, and Landweber performed about the same in all scenarios and converged to the solution determined by the centralized solution. Landweber method seems to be the best, but the difference is merely negligible. In both settings, the proposed algorithm converges to the centralized solution. We observe in Figs. 3 and 4, that choice of K has a dramatic effect on the behavior of the convergence. For $K = 18$ in Fig. 3, the convergence process was very slow. When increased to $K = 72$, the performance of the proposed algorithm is comparable to the centralized solution.

Further, the effect of the choice of the number of consensus steps T is demonstrated. Figure 5 shows the value of RMSE of the proposed algorithm during its individual steps of the consensus phase for an insufficiently small number of steps, specifically $T = 50$.

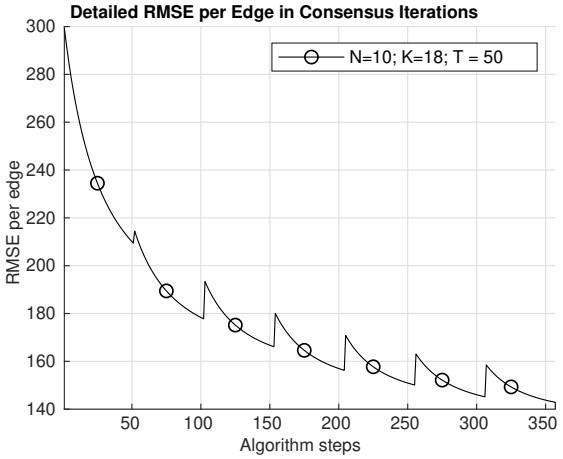

Fig. 5. Visualization of individual steps of the algorithm for insufficient number of consensus steps.

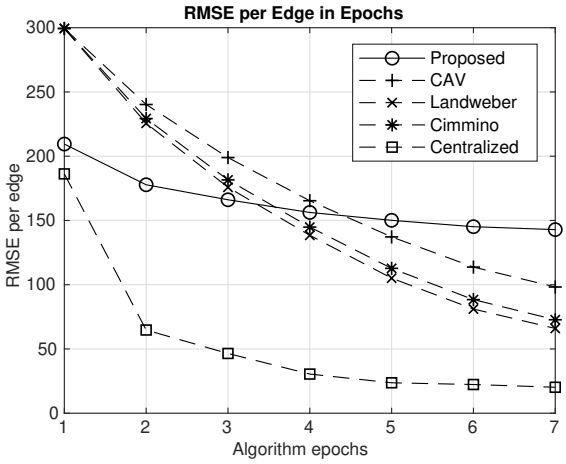

Fig. 6. Comparison of methods for $K = 18$, $N = 10$, $T = 50$.

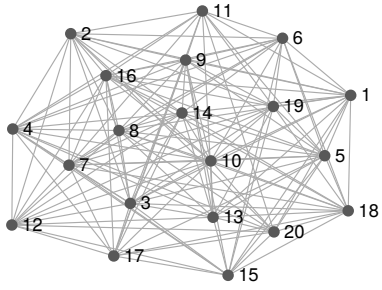

Fig. 7. Examined topology 2: Dense network with $N = 20$. Average degree of node is 15.

It is clearly seen that consensus was not reached within individual epochs, and this severely affected the overall comparison of results in Fig. 6. Compare this with Fig. 3, which differs only in the value of T.

Another results are provided for topology with $N = 20$ nodes, shown in Fig. 7. This graph is no longer fully connected, but the topology is still very dense, and the average degree of a node is 15. Analogical results as previously are presented in Fig. 8 for $K = 38$ and in Fig. 9 for $K = 95$.

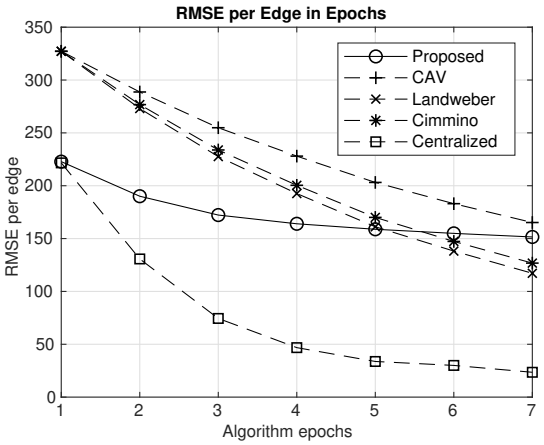

Fig. 8. Comparison of methods for $K = 38$, $N = 20$, $T = 500$.

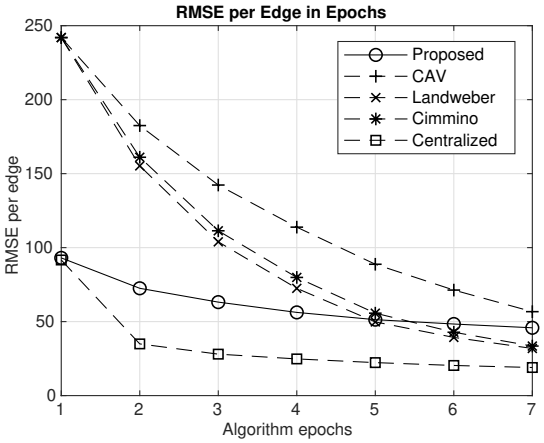

Fig. 9. Comparison of methods for $K = 95$, $N = 20$, $T = 500$.

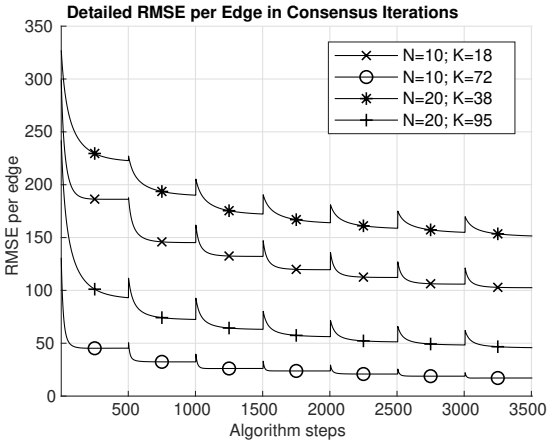

Fig. 10. Visualization of individual steps of the algorithm. These results were obtained during the previous experiments for both topologies and all numbers of measurements.

This time, the difference between the results for different values of K is not that significant. However, we can still clearly observe that an increased number of measurements significantly boosts the performance of the proposed algorithm. Comparing the different figures for different values of K, we can see that with the increasing number of measurements K, the achieved level of RMSE decreases faster. This is clear since the sets of equations to be solved are better conditioned, having more observations. However, the complexity of the calculations rapidly increases, as described above.

Finally, the evaluation of RMSE during individual steps of the consensus phases is shown in Fig. 10. For the purpose of comparison, results for both topologies and all values of K are shown together. These figures are useful to ensure that the number of iterations T is sufficient and to map the rate of consensus. Using these graphs, parameter T might be tuned for a specific application. Adaptive modification of parameter T could be designed based on the relative change of the estimated value. Note, the peaks appear on the curve at the moments of receiving new measurements. These were also observed in, e.g., [6] and can be affected by parameterizing of (9). Also, in this figure can be easily seen that the number of measurements K has a very significant impact on the result. For both topologies, the difference of achieved results for the different number of measurements is at this moment easily comparable.

To summarize this section, we provided results for two dense topologies, where the performance of the proposed distributed algorithm was compared to described reference solutions. While the centralized solution performed the best, its severe disadvantages were previously addressed. A comparison with selected reference methods was also provided. Firstly, it was shown how the number of measurements K affects the performance of the individual methods. Specifically, a higher value of K significantly accelerates the convergence. Secondly, we demonstrated how small value of consensus steps T decreases performance. The convergence of the proposed algorithm was successfully demonstrated.

5. Conclusion

In this paper, we presented a system model and a distributed algorithm to estimate a network delay profile with stochastic properties. It is challenging, especially in large and dense networks, to perform this task efficiently, avoiding malicious signaling overhead, while keeping the required estimate robust to inevitable stochastic properties of the real-world networks. The distributed nature of the proposed algorithm clearly overcomes this challenge. The fundamental core principle of the proposed method is to utilize a projection-based consensus algorithm to distribute local estimates over the network. This proposed solution was compared with a centralized solution. It is clear that a centralized solution forms a lower bound of performance since all measurements are available. However, the signaling overhead is its significant drawback. The proposed solution was also compared with other reference solutions based on the convention, iterative solvers. Numerical analysis showed that our solution converges to the same results as the reference solutions, and justifies its validity.

Acknowledgments

This work was supported by the Ministry of education, youth and sports of the Czech Republic, grant LTC17042, and by the Grant Agency of the Czech Technical University in Prague, grant No. CTU SGS19/069/OHK3/1T/13 and No. SGS20/068/OHK3/1T/13.

References

[1] COATES, A., HERO III, A. O., NOWAK, R., et al. Internet tomography. *IEEE Signal Processing Magazine*, 2002, vol. 19, no. 3, p. 47–65. DOI: 10.1109/79.998081

[2] TSANG, Y., COATES, M., NOWAK, R. D. Network delay tomography. *IEEE Transactions on Signal Processing*, 2003, vol. 51, no. 8, p. 2125–2136. DOI: 10.1109/TSP.2003.814520

[3] MOLOISANE, A., GANCHEV, I., O'DROMA, M. *Internet Tomography: An Introduction to Concepts, Techniques, Tools and Applications.* 1st ed., Newcastle upon Tyne (UK): Cambridge Scholars Publishing, 2013. ISBN: 978-1-4438-4421-5

[4] CHEN, A., CAO, J. Network tomography based on 1-D projections. *Lecture Notes-Monograph Series*, 2007, vol. 54, p. 45–61. DOI: 10.1214/074921707000000238

[5] MOU, S., LIU, J., MORSE, A. S. A distributed algorithm for solving a linear algebraic equation. *IEEE Transactions on Automatic Control*, 2015, vol. 60, no. 11, p. 2863–2878. DOI: 10.1109/tac.2015.2414771

[6] ZHAO, L., SONG, W. Z. Decentralized consensus in distributed networks. *International Journal of Parallel, Emergent and Distributed Systems*, 2018, vol. 33, no. 6, p. 550–569. DOI: 10.1080/17445760.2016.1233552

[7] MATEI, I., BARAS, J. Performance evaluation of the consensus-based distributed subgradient method under random communication topologies. *IEEE Journal of Selected Topics in Signal Processing*, 2011, vol. 5, no. 1, p. 754–771. DOI: 10.1109/JSTSP.2011.2120593

[8] NEDIC, A., OZDAGLAR, A. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 2009, vol. 54, no. 1, p. 48–61. DOI: 10.1109/TAC.2008.2009515

[9] OLFATI-SABER, R., MURRAY, R. M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 2009, vol. 49, no. 9, p. 1520–1533. DOI: 10.1109/TAC.2004.834113

[10] YU, H., JIAN, J. Multi-agent consensus with a time-varying reference state in directed network with switching topology and time-delay. In *International Conference on Wavelet Analysis and Pattern Recognition*. Baoding (China), 2009, p. 476–481. DOI: 10.1109/ICWAPR.2009.5207458

[11] NEDIC, A., PANG, J., SCUTARI, G., et al. *Multi-agent Optimization*. 1st ed., rev. Cetraro (IT): Springer International Publishing, 2018. ISBN: 3319971417

[12] KAMATH, G. *Decentralized Convex Optimization for Wireless Sensor Networks*. PhD. Thesis, Georgia State University, 2016. 73 pages. [Online] Cited 2019-12-06. Available at: https://scholarworks.gsu.edu/cs_diss/117

[13] CHUA, D. B., KOLACZYK, E. D., CROVELLA, M. Network Kriging. *IEEE Journal on Selected Areas in Communications*, 2006, vol. 24, no. 12, p. 2263–2272. DOI: 10.1109/JSAC.2006.884025

[14] RAJAWAT, K., DALL'ANESE, E., GIANNAKIS, M. B. Dynamic network delay cartography. *IEEE Transactions on Information Theory*, 2014, vol. 60, no. 5, p. 2910–2920. DOI: 10.1109/TIT.2014.2311802

[15] HANSEN, P. R. AIR Tools II: algebraic iterative reconstruction methods, improved implementation. *Numerical Algorithms*, 2018, vol. 79, no. 1, p. 107–137. DOI: 10.1007/s11075-017-0430-x

[16] CIMMINO, G. Approximate solutions for systems of linear equations (in Italian). *La Ricerca Scientifica*, 1938, vol. 9, p. 326–333.

[17] CENSOR, Y., GORDON, D., GORDON, R. R. Component averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems. *Parallel Computing*, 2001, vol. 27, no. 1, p. 777–808. DOI: 10.1016/S0167-8191(00)00100-9

[18] LANDWEBER, L. An iteration formula for Fredholm integral equations of the first kind. *American Journal of Mathematics*, 1951, vol. 73, no. 3, p. 615–624. DOI: 10.2307/2372313

[19] SCHARF, L. L. *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. 1st ed., rev. Boston (USA): Addison-Wesley, 1991. ISBN: 9780201190380

[20] AZZIAN-RUHI, N., LAHOUTI, F., AVESTIMEHR, S., et al. Distributed solution of large-scale linear systems via accelerated projection-based consensus. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Calgary (Canada), 2018, p. 6358–6362. DOI: 10.1109/ICASSP.2018.8462630

[21] OLFATI-SABER, R., FAX, J. A., MURRAY, R. M. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 2007, vol. 95, no. 1, p. 215–233. DOI: 10.1109/JPROC.2006.887293

[22] ARIOLI, M., DUFF, I., NOAILLES, I., et al. A block projection method for sparse matrices. *SIAM Journal on Scientific and Statistical Computing*, 1992, vol. 13, no. 1, p. 47–70. DOI: 10.1137/0913003

[23] GARBELETTO, C., DONINI, M., RAGAZZONI, R., et al. Kaczmarz and Cimmino: Iterative and layer-oriented approaches to atmospheric tomography. *Proceedings SPIE*, 2016, vol. 9909, no. 1, p. 1327–1345. DOI: 10.1117/12.2232494

[24] KOLAR, J. *Implementation of Distributed Network Tomography Applied to Stochastic Delay Profile Estimation*. Matlab script, [Online] Cited 2019-12-8. Available at: https://www.dropbox.com/sh/gdl6rf1ynzhzdvv/AAAcXwsXh4N1n4iI386ULbOda?dl=0

About the Authors ...

Jakub KOLAR received his B.Sc. from the Faculty of Electrical Engineering, Czech Technical University in Prague.

Jan SYKORA is a Professor in the Faculty of Electrical Engineering at the Czech Technical University in Prague, and a consultant for the communications industry in the fields of advanced coding and signal processing.

Umberto SPAGNOLINI is a Professor in Signal Processing and Telecommunications at Politecnico di Milano, Italy. Prof. Spagnolini's research focuses on statistical signal processing, communication systems, and advanced topics in signal processing for remote sensing and wireless communication systems. He has authored 300 patents and papers in peer-reviewed journals and conferences.