Decentralizing Network Inference Problems with Multiple-Description Fusion Estimation (MDFE)

Mehdi Malboubi, Cuong Vu, Chen-Nee Chuah
Dept. of Electrical & Computer Engineering
University of California, Davis, USA.
{mmalboubi, cnvu, chuah}@ucdavis.edu

Puneet Sharma
Hewlet-Packard Laboratories (HP Labs)
Palo Alto, California, USA.
puneet.sharma@hp.com

***Abstract*—Two forms of network inference (or tomography) problems have been studied rigorously: (a) traffic matrix estimation or completion based on link-level traffic measurements, and (b) link-level loss or delay inference based on end-to-end measurements. These problems are often posed as under-determined linear inverse (UDLI) problems and solved in a *centralized* manner, where all the measurements are collected at a central node, which then applies a variety of inference techniques to estimate the attributes of interest.**

This paper proposes a novel framework for *decentralizing* these large-scale UDLI network inference problems by intelligently partitioning it into smaller sub-problems and solving them independently and in parallel. The resulting estimates, referred to as *multiple descriptions*, can then be fused together to compute the global estimate. We apply this Multiple Description and Fusion Estimation (MDFE) framework to three classical problems: traffic matrix estimation, traffic matrix completion, and loss inference. Using real topologies and traces, we demonstrate how MDFE can speed up computation time while maintaining (even improving) the estimation accuracy and how it enhances robustness against noise and failures. We also show that our MDFE framework is compatible with a variety of existing inference techniques used to solve the UDLI problems.

I. Introduction

Due to the complexity of current's Internet and the exploding volume of data traffic, there are often aspects of the networks that are challenging or infeasible to measure directly. This has drawn researchers to the field of network inference (or network tomography [1]), which involves applying a variety of inference strategies to estimate network's internal characteristics based on a limited set of measurements. Many network inference problems are formulated as Under-Determined Linear Inverse (UDLI) problems. These problems are naturally ill-posed in the sense that the number of measurements are not sufficient to uniquely determine the solution. Two forms of network inference problems [2] have been studied rigorously: (a) origin-destination (path-level) traffic volume estimation based on link-level traffic measurements, such as traffic matrix (TM) estimation [3] or TM completion [4], and (b) link-level parameter's (such as loss, delay, or bottleneck bandwidth) estimation based on end-to-end measurements [5], [6], [7].

Prior work has mostly focused on designing better measurement methodology and inference techniques to improve the accuracy of the solution. For this purpose, side information are incorporated to change an ill-posed problem to a well-posed problem. Side information, based on the application, are provided from different sources , e.g., auxiliary measurements such as NetFlow data [8], and from diverse perspectives, e.g., using underlined deterministic or statistical models [3], [4], [7], [9].

Although the uniqueness and accuracy of the solution are important, many network inference problems need to be solved in a timely manner for practical deployment. Nevertheless, most existing studies attempt to solve the network inference problem in a one-shot, centralized manner, where all measurements are collected at a central node, which then applies domain-specific inference techniques to estimate the attributes of interest.

This paper tackles these network inference problems from a new angle and asks the question: can we design an efficient and robust framework to solve these large-scale UDLI problems in a decentralized manner? Our goal is to speed up the computation process to produce timely estimates (especially in a dynamic network environment), without compromising the accuracy of the solution. Towards this end, we propose Multiple Description and Fusion Estimation (MDFE) framework that decentralizes a large-scale network inference problem by intelligently partitioning it into smaller sub-problems and solving them independently and in parallel. The results, solved in respective sub-spaces and referred to as multiple descriptions, are then fused together to reconstruct the global estimate. Each sub-space could potentially produce a more precise description of a sub-set of the solution; in fact, these descriptions are considered as side/supplementary information for each other, provided from different perspectives.

MDFE is a flexible framework that can be applied to different UDLI problems, and is complementary to the inference techniques proposed previously for solving specific network inference problems. In this paper, we demonstrate how MDFE can be applied to network inference problems such as TM Estimation (TME), TM Completion (TMC) and Loss Inference (LI), and we show, MDFE is compatible with a variety of previously proposed inference techniques, including least square error estimation, expectation maximization, and regularized matrix factorization methods [8], [11], [4], [7].

By reducing the problem complexity, MDFE can significantly speed up the computation time and reduce processing power. Through evaluation using real topologies and data, we demonstrate that this can be achieved without compromising the accuracy of the global estimate. This, specifically, has important implications in distributed and dynamic environments (e.g., distributed data centers or clouds), where inference process must be performed at much faster time scales. This framework is suitable for today's computing paradigm where a large-scale problem can be divided into smaller sub-problems and distributed among multi processors. Also, by exploiting redundancy between different sub-spaces, MDFE can enhance the robustness against noise and failures in the monitoring infrastructures. It also reduces the overhead involved in sending all measurements to a central node for global estimation.

The improvement in the estimation accuracy using MDFE depends on the structure of the problem, sub-space estimation method, partitioning technique and the fusion process which are discussed in this paper. Due to space limitation, we occasionally refer to [12] for the detailed discussion and additional results. Our main contributions are:

- To the best of our knowledge, we are first to develop the concept and theory of MDFE for solving UDLI inference problems. We demonstrate how to effectively design the MDFE framework and realize it in practice.
- We develop and evaluate three algorithms to partition the original large-scale problem into smaller sub-problems under MDFE; we also introduce different fusion methods to combine the multiple descriptions to produce the global solution.
- We demonstrate the efficacy of MDFE in practice by applying it to three important problems in network monitoring and management: TM Estimation (TME), TM Completion (TMC) and Loss Inference (LI).
- Using realistic network topologies and traffic data, we show how MDFE can speed up computation by maintaining (and even improving) the accuracy of the global estimates.

The rest of this paper is organized as follows. Section II discusses the most relevant work in the context of the three example network inference problems. Section III develops the theory of MDFE and addresses main steps in implementation of this framework in practice. In Section IV the performance and efficiency of this framework are evaluated for different applications in networking. Finally, Section V summarizes the main results of the paper.

II. Background and Related Work

There is a rich literature on network tomography and it would be impossible to enumerate all the related work. We would like to emphasize that the main goal of this paper is not to design new, improved algorithm for solving specific network inference problem. Instead, we are proposing a framework for efficiently solving a class of UDLI problems by adopting a divide-and-conquer approach and leveraging existing inference techniques to solve the intelligently partitioned sub-problems under MDFE. In this paper, three network inference problems (TM estimation, TM completion, and loss inference) are used to showcase MDFE framework (see Section IV). Here, we discuss the most relevant work in the context of these network inference problems.

The traffic matrix (TM) is a measure of origin-destination traffic intensity that can be defined at different levels: between routers, IP-prefixes, or even AS domains. It provides essential information for network design, traffic engineering, and anomaly detection. TM estimation is often formulated as a constrained UDLI problem where side information from different sources/perspectives are provided to uniquely identify a more accurate solution. Side information can be provided as link and flow conservation constraints to reduce the feasible solution space [13] or the underlined statistical models of Origin-Destination Flows (ODFs) where Bayesian or Maximum Likelihood Estimation (MLE) techniques are used for TM estimation [1],[11],[14]. In [8] data from multiple sources including SNMP link loads and Sampled NetFlow records are used to provide more accurate estimates.

Another network inference problem is network performance tomography, which is defined as the inference of internal link properties from end-to-end measurements [6]. In [7], links loss rate inference problem is modeled as an UDLI problem where statistical characteristics of congested links are used as side information to uniquely estimate link loss rates. Also, in [10], first and second-order moments of end-to-end measurements are combined to estimate loss rates.

III. Estimation with Multiple Description Fusion

Consider the under-determined linear system of equations Eq.(1) where Y is an $(m \times 1)$ observation vector, H is an $(m \times n)$ observation matrix $(m < n)$ and X is an $(n \times 1)$ vector of unknowns. The general solution to this problem is of the form $\hat{X} = X + \mathscr{N}(H)$ where $\mathscr{N}(H)$ represents a solution from the span of the null space of H; therefore, there are many solutions for this problem. A linear inverse problem is defined as the process of uniquely inferring X as a linear function of observation Y which can be formulated as an un/constrained optimization problem where the main goal is to minimize error $e = Y - HX$ in an appropriate sense.

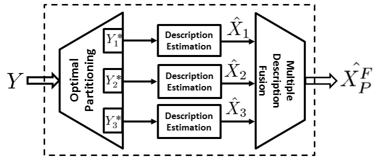
$$Y = HX \tag{1}$$

In MDFE framework, the original (global) UDLI inference problem described by Eq.(1) is partitioned into L local sub-problems shown in Eq.(2), which are independently solved and sub-space estimates/descriptions $\{\hat{X}_i\}_{i=1}^{L}$ are then fused together to provide a more accurate solution in an efficient way (Figure 1(a)). The fusion process is accomplished by applying appropriate weights to each local estimate during the fusion phase. Eq.(3) describes this process where operator $\oplus$ denotes the fusion process of the partitioned problem, that is, combining the subset of unknowns observed and estimated by different sub-spaces. Figure 1(b) gives an intuitive perspective of this Multiple Description Fusion (MDF) process where the original problem is partitioned into 3 sub-problems.

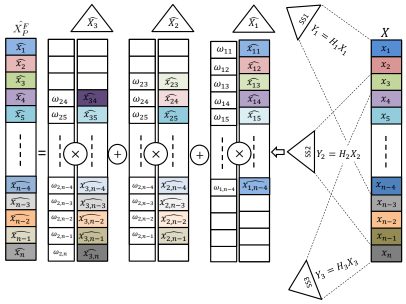
$$Y = HX \Leftrightarrow \begin{bmatrix} Y_1 \\ \hline \vdots \\ \hline Y_L \end{bmatrix} = \begin{bmatrix} H_1 X_1 \\ \hline \vdots \\ \hline H_L X_L \end{bmatrix} \tag{2}$$

$$\hat{X_P^F} = \oplus_{i=1}^{L} \omega_i^F \hat{X}_i \tag{3}$$

To compute $\hat{X}_P^F$, the UDLI inference problem is formulated as the optimization problem in Eq.(4), which illustrates that the overall performance of MDFE framework is a *joint* function of sub-space estimation technique, partition P and fusion process F. Hence, to successfully apply the MDFE framework in practice, three steps must be accomplished correctly: a) effectively partition the problem into sub-problems, b) construct multiple descriptions by adopting proper sub-space estimation techniques to solve the sub-problems, and c) fuse



(a) General block diagram of MDFE.



(b) TM Estimation with MDF from sub-spaces.

Fig. 1: MDFE process: a system perspective.

the sub-space estimates to provide more precise and robust description. The essence of the joint optimization problem in Eq.(4) lies in an NP-hard set partitioning problem that is extremely difficult to solve. Hence, we *decouple* and address steps a-c, independently. Since the estimation techniques to solve specific network infernce problems are well studied, we first discuss how these existing techniques can be leveraged to provide sub-space estimates (step b). Then, taking practical constraints into account, we discuss the design of the most effective partitioning and fusion methods.

$$\hat{X_P^F} = \min_X \|Y - HX\|_p \quad \text{s.t:} \quad \text{Problem constraints} \tag{4}$$

A. MDFE in Practice: Multiple Description Construction

To construct multiple descriptions, sub-inference problems must be properly defined and the best sub-space estimation technique is selected by choosing appropriate norm p in Eq.(4), depending on the characteristics of the input X, matrix H and problem's side information(/constraints).

Let I denotes the set of all indicies of observations ($I = \{1, 2, ..., m\}$) and I_i denotes the i^{th} set of disjoint indices of measurements where $I = \bigcup_{i=1}^{L} I_i$ and $I_i \bigcap I_j = \emptyset$ for $i \neq j$. Then, set $P = \bigcup_{i=1}^{L} I_i$ forms a *Partition* of I. Let J denotes the set of all indices of unknowns ($J = \{1, 2, ..., n\}$) and J_i denotes the i^{th} set of indices of unknowns where $J = \bigcup_{i=1}^{L} J_i$; however, the intersection of J_i and J_j is not necessarily empty. Now, lets $Y_i := \{y_k\}_{k \in I_i}$, $H_i := H(I_i, J_i)$ and $X_i := \{x_k\}_{k \in J_i}$. Accordingly, the original problem Eq.(1) is divided into L sub-problems as $Y_i = H_i X_i$ (see Eq.(2)) and the i^{th} local estimate is computed by solving this sub-problem.

Since many UDLI inference problems in networking, communication and signal processing are formulated as Linear Least Square Error Estimation (LLSEE) problems and to develop the basic theory of MDFE, here, we consider the unconstrained L_2 minimization as our sub-space estimation technique, that is, $p = 2$ in Eq.(4) without any constraints. Hence, it is assumed that input vector X does not include unusual inputs that differ in size by large order of magnitudes. Accordingly, in the global case, the LLSEE is computed using the pseudoinverse of H ($H^{\dagger}$) which can accurately obtained using the Singular Value Decomposition (SVD) with computation complexity $O(mn^2)$ flops. Also, the i^{th} local LLSEE estimate is computed using the pseudoinverse of H_i (an $m_i \times n_i$ matrix) with complexity $O(m_i n_i^2)$ flops (see Eq.(6)). Note that, the solution of global and local problems (i.e. Eq.(5) and Eq.(6)) could be different because the null space of H and H_i are not necessarily equal, and comparing with global case, it can be shown that based on the structure of H, MDFE can *exactly* reconstruct X (see App.C in [12]).

$$\hat{X} = \min_X \|Y - HX\|_2 = H^{\dagger} Y = (H^T (HH^T)^{-1}) Y \tag{5}$$

$$Y_i = H_i X_i \Rightarrow \hat{X}_i = H_i^{\dagger} Y_i = H_i^T (H_i H_i^T)^{-1} Y_i \tag{6}$$

B. MDFE in Practice: Partition Design

The accuracy of redundant estimates from sub-spaces depends on the design of partition P that can be formulated as an integer optimization problem to achieve the best possible performance. Assuming there are m measurements and L sub-spaces, then there are $S_{m,L} = \frac{1}{L!} \sum_{j=0}^{L} (-1)^{L-j} \binom{L}{j} j^m$ ($S_{m,L}$ denotes Stirling number of the second kind) partitions. The number of partitions with K elements in each subset (where $K \propto \frac{m}{L}$) is a fraction of Stirling number $S_{m,L}$ that is still a large number in large-scale networks, where $m >> L$. To simplify this NP-hard problem and maximize the MDFE performance, pseudo-optimal or heuristics algorithms are developed. Note that in these algorithms, L is assumed to be known a-priori. In fact, the number of sub-spaces is a design criteria which must be selected by considering other constraints such as the number of processors or the required processing gains (see performance criteria in Table I).

In Alg.1, the effectiveness of sub-spaces are sequentially measured and maximized to form partition P. Here, the criterion used to evaluate the partition choice is the Condition Number (CN) of the observation matrix H_i which is defined as the ratio of the maximum and minimum singular values of the matrix H_i and it is an indication of the quality of a matrix and it determines a bound ($CN \geq 1$) on the rate at which the solution will change with respect to a change in measurements. The lower the CN, the more well-conditioned problem and the more accurate solution are. In Alg.1, the best sub-spaces are sequentially chosen to get the best possible partition $P = \bigcup_{i=1}^{L} I_i$ with the lowest CN, which can provide a well behaved partition P, and a more accurate and stable solution in each sub-space. This algorithm starts from the first row of H and sequentially chooses the row that minimize the CN of the sub-matrix. This continues to complete the first sub-space I_1 with K rows. After removing these K rows from H, the algorithm repeats from the beginning. The sub-problems can be solved in parallel or sequentially. Note that the CN of each individual row of H is one; however, this is not an interesting case because: 1) in practice, the number of processors/sub-spaces (L) are limited (in parallel case), and 2) large L's reduces processing gain Δ_s (in sequential case).

Algorithm 1 : Greedy CN based Partitioning

Initialization: $I = \{1, ..., m\}$ and $i = 1$
while $i \leq L$ **do**
- Construct I_i by sequentially choosing K rows of H with lowest CN
- Set $I = I \backslash I_i$ and $i = i + 1$
end while

Algorithm 2 : QRP based Partitioning

Initialization:
- Compute the QRP factorization of routing matrix H
- Divide $|R|$ into L batches with almost similar successive values
- Construct P_0 as rows of H corresponding to indicies of $|R|$ in each batch
- Set $i = 1$
while $i \leq L$ **do**
- Modify the boundaries of set I_i (by extending or shrinking the boundaries of the set)
- Check the performance until the maximum gain is achieved
- $i = i + 1$
end while

In the second pseudo-optimal partitioning algorithm, Alg.2, the design of partition P is based on the structure of the observation matrix H captured by QR decomposition of H, where H is represented as:

$$H = QR = Q_{m\times m} \left[\begin{array}{cc} R^{11}_{r\times r} & R^{12}_{r\times(n-r)} \end{array} \right] \quad (7)$$

with orthonormal matrix Q, upper-triangular matrix R^{11}, and $rank(H) = r(= m)$. For rank deficient matricies, QR decomposition with pivoting, known as QRP, is used to solve linear system of equations and recognize singularities or rank deficiency. Here, the pivoting strategy attempts to produce R^{11} as well-conditioned as possible. Accordingly, the diagonal elements of $|R|$ occur in decreasing order, revealing the linear in/dependence among the rows of H [15]. In Alg.2 diagonal elements of matrix $|R|$ are grouped to construct initial partition P_0 where each batch consists of a set of indicies of successive diagonal entries of matrix $|R|$. Initial Partition P_0 is then modified, by extending or shrinking the boundaries of sets $\{I_i\}_{i=1}^{L}$, to improve the performance of MDFE and achieve a pseudo-optimum partition P. In this process, appropriate set of sample inputs X must be used to evaluate the performance in each step. In this algorithm, observation matrix H is assumed to be full row-rank. Thus, rows corresponding to very small values of $diag(|R|)$ are removed. The performance of these two algorithms are **close** to optimal; for example, on a smaller network, the first two algorithms are close to optimal by 0.05% and 3%, respectively (see Table 7 in [12]).

The third algorithm (Alg.3) is a heuristic partitioning algorithm that uses the topology of the network where L nodes with highest degrees are selected as clustering nodes. Observation measured at clustering nodes along with measurements that can be transfered to these nodes with minimum cost (e.g. communication cost & delay) form a partition of the set of measurements I. This heuristic partitioning algorithm is important where the *nature* of the estimation problem is distributed and communication costs and delay are critical factors that must be considered in the implementation of MDFE framework.

Algorithm 3 : Graph based Partitioning

Choose L nodes with highest degree as clustering nodes
while $i \leq L$ **do**
- $I_i = \{$the index of observations measured at i^{th} cluster node$\}$
- $I_i = I_i \bigcup \{$measurement's indices transfered to i with minimum cost$\}$
- $i = i + 1$
end while

C. MDFE in Practice: Fusion Algorithm

MDFE process is completed by applying fusion process F to the local estimates. Here, three different weighting functions (Eq.(8)) are considered in Eq.(3). Let ICN and RoD denote the Inverse Condition Number and Rank over Dimension (i.e. # of unknowns in each sub-space), respectively. By applying fusion operator $\oplus$, unknowns observed in different sup-spaces are combined to produce the final estimate $\hat{X}^F_P$. The first two fusion functions choose x_i from the sub-space with highest ICN or RoD, while the third one computes the average of the observed x_i's produced by different sub-spaces. This averaging process, by itself, can improve the accuracy of the estimation by increasing the Signal-to-Noise Ratio (SNR). These fusion techniques are also efficient because the computation overhead of using these fusion methods are negligible compared with the computation time of sub-space estimation techniques, especially for large-scale problems.

$$\omega^{ICN}_{ij} = \begin{cases} 1, & \text{if } i \in \text{sub-space with highest ICN and } j \in J_i \\ 0, & ow \end{cases} \quad (8)$$

$$\omega^{RoD}_{ij} = \begin{cases} 1, & \text{if } i \in \text{sub-space with highest RoD and } j \in J_i \\ 0, & ow \end{cases}$$

$$\omega^{Avg}_{ij} = \frac{1}{\#\text{ of repetition of } X_j \text{ among all sub-spaces}} i = 1 : L; j = 1 : n$$

D. The Efficiency of MDFE

MDFE is an efficient framework that can improve the performance of system from different perspectives, In fact, MDFE not only reduce processing time/power, but also provide better estimates in most cases, and can improve the robustness of the system against noise and failures.

MDFE is able to provide more accurate estimates due to two factors. First, partitioning increases the redundancy between descriptions, produced by observing each unknown x_j from different sub-spaces (Figures 1(b)). This redundancy is used by the fusion process to enhance the SNR and improve the accuracy of estimation. The amount of redundancy depends on the number of subsets (L) in partition P and the structure of observation matrix H. This redundancy is evaluated by two measures: a) the sum of the Number of Unknowns (NoU) observed by different subspaces ($Rdn1$) and b) the sum of the ratio $RoD_i := \frac{rank(H_i)}{n_i}$, which conceptually represents the contribution of each independent measurement into the estimation of each unknown x_j, as shown below:

$$Rdn1 = \sum_{i=1}^{L} NoU_i \quad \& \quad Rdn2 = \sum_{i=1}^{L} RoD_i \quad (9)$$

Second, partitioning does not change the input-output relationship; however, reducing the NoUs and removing non-observed unknowns *could* help to group more-similar unknowns in sub-spaces. Therefore, LLSEE generates more coherent and robust solution in the absence of unusual inputs. In addition, the row partitioning of observation matrix H improves the CN of H_i's (Prop.1); therefore, sub-spaces with better CN can provide more accurate and robust estimates. In fact, in the presence of noisy observations, sub-spaces with lower CN not only reduces the covariance of error but also they are potentially able to attain lower Mean-Square Error (MSE) because the lower bound on MSE is reduced by partitioning (Prop.2). This fact was also verified through our direct investigation where we observed a strong positive correlation between the performance of MDFE and the CN of sub-spaces [12]. Since MDFE can provide more redundant and better estimates; it can also improve the robustness of the system against noise, failure and information-loss, in the computing and monitoring infrastructures.

Proposition.1: Let H be a matrix in ($R^{m\times n}$ with rank m) and H_i denotes a matrix constructed from a set of rows of H. Then: $CN(H_i) \leq CN(H)$. Proof: See App.A in [12].

Proposition.2: Let $Y = HX + \epsilon$ where $\epsilon \sim \mathcal{N}(0, \sigma_0^2 I_m)$ denotes measurement noise. Then: 1) $Var(\mathcal{E}) \propto CN(H)$ where $\mathcal{E} := X - LLSEE(X) = X - H^{\dagger}Y$, and 2) the lower-bound for the MSE of LLSEE is reduced by partitioning.Proof: See App.A in [12].

Besides improving accuracy, MDFE can also reduce processing time, significantly. This is achieved by reducing the dimension of the problem in each sub-space. Considering the complexity of LLSEE in each sub-space i.e. $O(m_i n_i^2)$, since $m_i < m$ and $n_i < n$, local inference problems can be solved more efficiently. Using parallel computing infrastructures, the processing time can be bounded by maximum local processing time. However, using sequential computing infrastructures, reduction in processing time can be achieved if the sum of local processing times is less than the global processing time. In this case, the number of sub-spaces must be carefully chosen. Considering the fact that processing power is also proportional with the computational complexity of the problem, the same argument can be used to show that, based on the number of sub-spaces, MDFE can also be a power efficient framework where the sum of local processing powers is less than global processing power. The proportionality of processing time/power with complexity of the problem can provide a criteria to choose the number of sub-spaces (L). From this point of view, the designer of the system can choose L to achieve required reduction in processing time/power (see Sec.5 in [12]). Note that L is chosen with a reasonable balance between improvement in accuracy and processing gains and considering other practical constraints.

E. MDFE: Performance Evaluation Metrics

The performance of the MDFE is evaluated using various criteria which are introduced in Table I. $\hat{X^G}$ denotes the global estimate, $\hat{X_w^F}$ denotes the MDFE estimate (where w denotes fusion function ICN, RoD or Avg in Eq.(8)), and $Gain_{L2}$ quantifies the performance improvement using MDFE framework comparing with global case. Parallel and sequential processing gains (Δ_p and Δ_s) measure the reduction in computation time using MDFE structure. As it was explained, sequential processing gain can also be an indication of the reduction in processing power using MDFE. The effectiveness of MDFE with *relative error norms* are also shown in [12].

$GE = \frac{\left\lVert X - \hat{X}^G \right\rVert_2}{\lVert X \rVert_2}$, $FE_w = \frac{\left\lVert X - \hat{X}_w^F \right\rVert_2}{\lVert X \rVert_2}$	$Gain_{L2} = 100 \times \frac{GE - FE_{Avg}}{GE}$
$\Delta_p = 100 \times \frac{GPT - max(\{LPT_i\}_{i=1}^{L})}{GPT}$	$\Delta_s = 100 \times \frac{GPT - \sum_{i=1}^{L} LPT_i}{GPT}$

TABLE I: Performance evaluation criteria where w denotes the type of the fusion function (that is, ICN, RoD or Avg in Eq.(8)) , and GPT and LPT_i denote Global and i^{th} Local Processing Times, respectively.

F. MDFE: Illustrative Example

This illustrative example shows how MDFE framework can improve the accuracy to *exactly* reconstruct unknown vector X. Consider UDLI problem $Y = HX$ where $Y = [y_1, y_2]^T$, $H_{2\times 3} = [1,1,0;0,1,1]$ and true X is $X = [x_1, x_2, x_3]^T = [1,1,1]^T$ and $Y = [2,2]^T$. Assume we partition this problem into two sub-problems defined as $y_1 = H_1 X_1 = [1,1][x_1, x_2]^T$ and $y_2 = H_2 X_2 = [1,1][x_2, x_3]^T$ where $CN(H) = 1.73$, $CN(H_1) = CN(H_2) = 1$ and $RoD_1 = RoD_2 = \frac{1}{2}$. The global solution of original problem is $\hat{X}^G = X + \mathscr{N}(H)$ and local estimates are $\hat{X}_1 = X + \mathscr{N}(H_1)$ and $\hat{X}_2 = X + \mathscr{N}(H_2)$. It can easily be verified that $\mathscr{N}(H)$ and, $\mathscr{N}(H_1)$ and $\mathscr{N}(H_2)$ are different; hence, the global and local solutions may be different (see App.C in [12]). In this example, the global LLSEE is computed using Eq.(5) as $\hat{X}^G = [\frac{2}{3}, \frac{4}{3}, \frac{2}{3}]^T$ and local LLSEE are computed using Eq.(6) as $\hat{X}_1 = [1,1,0]^T$ and $\hat{X}_2 = [0,1,1]^T$. By fusing local descriptions using Eq.(3), the MDFE estimates are computed as $\hat{X}_{ICN}^F = \hat{X}_{RoD}^F = [\hat{x}_1(1), \hat{x}_i(2), \hat{x}_2(3)]^T = [1,1,1]^T$ (where i denotes the subspace with maximum ICN and/or RoD) or $\hat{X}_{Avg}^F = [\hat{x}_1(1), \frac{\hat{x}_1(2)+\hat{x}_2(2)}{2}, \hat{x}_2(3)]^T = [1, \frac{1+1}{2}, 1]^T = [1,1,1]^T$, and accordingly, $GE = 0.3333$ and $FE_{ICN} = FE_{RoD} = FE_{Avg} = 0$.

IV. Network Inference using MDFE

The main goal in this section is to show the effectiveness of MDFE framework in different applications, including TM estimation, TM completion and loss inference. In fact, we illustrate that MDFE framework is compatible with a variety of existing inference techniques used to solve the UDLI problems. We also show that MDFE is effective for inputs with different distributions and on networks with different topologies. Among these, different partitioning algorithms are used to show the effectiveness of MDFE framework and partitioning techniques.

Here, three different networks are considered, including: 14-Node Tier-1 PoP Topology (Figure 2), Abilene [17] and GEANT [16] networks. The routing matrix H of the first network is a (50×182) matrix with density $D = 0.0415$ ($D = \frac{\#of non-zero entries}{m \times n}$). $H_{Abilene}$ is a (30×144) matrix with density $D = 0.0353$ and H_{Geant} is a (74×529) matrix with density $D = 0.036$. All routing matrices are binary and full row-rank. In addition, synthetic inputs are generated using three different distributions [3]: 1) Uniform

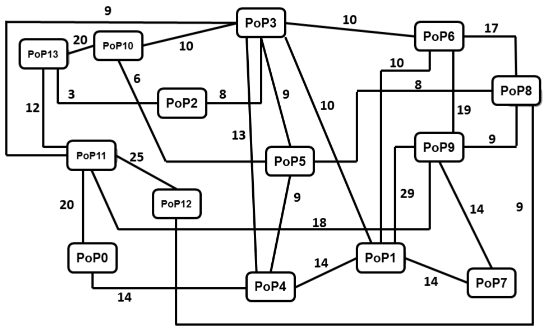


Fig. 2: 14-Node Tier-1 POP Topology [3].

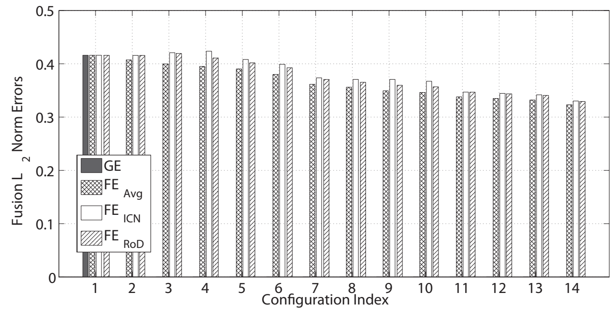


Fig. 3: Global&MDFE errors v.s. # of subspaces ($X \sim U(100, 500)$).

Network	Date	Duration	Resolution	TM Size
Abilene	2004-05-01	1 week	5 min.	144 × 2016
GEANT	2005-01-08	1 week	15 min.	529 × 672

TABLE II: Real Datasets under study.

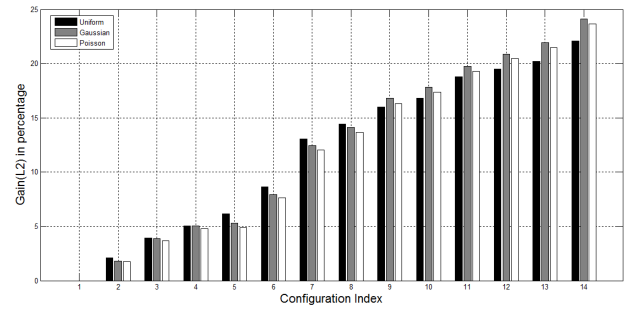


Fig. 4: $Gain_{L2}$ for different distributions.

distribution where $x_i \sim U(100, 500)$; 2) Gaussian distribution where$x_i \sim \mathcal{N}(\mu_i, 40)$ (where $\mu_i \sim U[100, 500]$), and 3) Poisson distribution where $x_i \sim Pois(\lambda_i)$, $\lambda_i \sim U[100, 500]$. These synthetic data are used to evaluate the performance of TME on the first network (Figure 2), and likewise, real network data (Table II) are used on Abilene and GEANT networks (real data follows heavy-tailed distributions). Note that, although we use synthetic data for TME in networking, inputs with Uniform, Gaussian and Poisson distributions are appeared in many applications in communication, signal processing and control, where MDFE can also be applied to improve the performance of system.

A. Traffic Matrix Estimation

Considering $Y = HX$, TME is an UDLI inference problem where X is the TM (each entry of X represents an ODF in the network) and it is estimated by knowing routing matrix H and observing link load measurement vector Y. In the first evaluation, the network shown in Figure 2, is partitioned into $L = 1, ..., 14$ sub-spaces using Alg.3 and synthetic TM inputs are applied to generate Y; then MDFE with LLSEE is used to infer $\hat{X}^F_w$. For each L, this process is repeated using a Monte-Carlo simulation. Figures 3 and 4 show the improvement achieved by applying MDFE for TM estimation on TM inputs with different distribution (as in [3], [11]) where the number of sub-spaces (or equivalently Configuration Index) varies from 1-to-14. Figure 5 shows that the redundancy of observed unknowns is increased as the number of local sub-spaces is increased. Also, the ICN of local sub-spaces have been improved (Prop.1). These figures prove the proportionality of MDFE performance with the enhancement of ICN and redundancy of unknowns observed in different sub-spaces. In addition, Figure 3 indicates the performance of different weighting functions (Eq.(8)) in the multiple description fusion process. Among these, computing the mean value of observed ODFs (using ω^{Avg}) have the best performance. However, RoD and ICN based fusion techniques can also achieve good performance while reducing the communication cost in the distributed/decentralized implementation of MDFE. On average, these improvements are almost achieved over 80% of the iterations.

Figure 6 shows that processing gain is significantly improved when the TM estimation problem is distributed among local sub-spaces; note that this gain is independent of partitioning algorithm and can be achieved where communication delays are negligible in comparison with processing times (see [12]). This figure also indicates that there is an optimum number of sub-spaces (5 in this case) for sequential TM estimation (Δ_s). Since the processing power is a function of the complexity of the algorithm, Δ_s in Figure 6 also indicates that MDFE can reduce the processing power. When the number of parallel processors are limited; the performance of MDFE can be increased by the partitioning of each sub-problem into multiple sub-problems where multiple description fusion can be performed in multi-stages at each local node. MDFE framework is also match with the architecture of today's multi-processor computing systems where a large-scale system can be divided into smaller sub-problems solved by each processor. This not only facilitates the problem of storing a large scale system, but also further reduction in processing time can be achieved by using local-fast memories.

- MDFE Robustness: MDFE improves the robustness of the system against noise in link load measurements and lossy informations (due to failures in communication networks and computing infrastructure). According to [8], noise in link load measurements (due to disalignment of polling intervals) can be modeled as a White Gaussian Noise (WGN); therefore, we added WGN to link measurement vector Y with different SNRs to evaluate the performance. Table III shows that MDFE is able to achieve better improvement in the presence

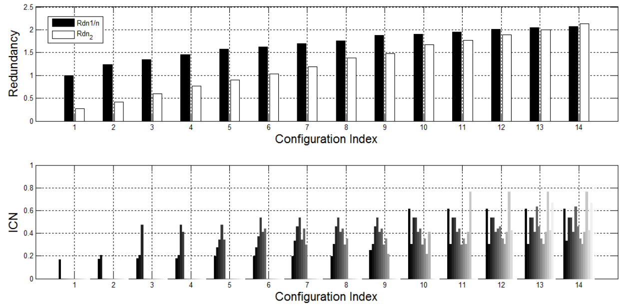


Fig. 5: The characteristics of local sub-spaces.

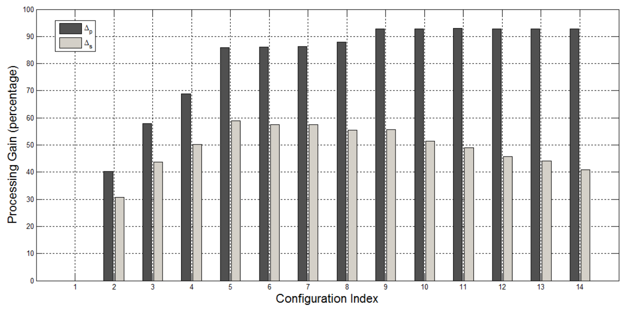


Fig. 6: Processing gain v.s. # of sub-spaces.

of noisy link load measurements. Our results also indicate that MDFE is robust against sub-space erasure in the system. To increase the robustness, sub-spaces with higher number of observed unknowns (e.g. sub-space 4) must be effectively protected and/or the number of sub-spaces must be increased (for further results on real data see Sec.7 in [12]).
- EM compatibility: To show the compatibility of MDFE with EM algorithm (as an ML estimator) we implemented TME method in [11] using MDFE framework. Table IV summarizes the estimation gain of MDFE when two different TM estimation methods (LLSEE and EM) are used. It shows that: 1) MDFE reduces the estimation error in both cases, which implies that MDFE is compatible with both TME methods, and 2) using the prior knowledge about the distribution of the TMs can improve the accuracy of the MD-TME.

- Compatibility of MDFE with different sources of data: Nowadays, NetFlow records are widely supported by vendors and deployed in most of the operational IP networks; then, complete/partial TM measurements can be used as side information to improve the accuracy of TM estimates. However, real TM measurements and SNMP data are noisy due to sampling and polling processes, respectively [8]. To address these challenges we adopt the TME method in [8] which is formulated as:

$$\hat{X} = X + \epsilon^X \;\;\&\;\; \hat{Y} = HX + \epsilon^Y \;\;\Rightarrow\;\; V = CX + \epsilon \quad (10)$$

where $\hat{X}$ denotes the TM measurement from NetFlow, $\hat{Y}$ denotes SNMP link load measurements, ϵ^X and ϵ^Y are respectively Gaussian noises in NetFlow and SNMP records, $V = [\hat{X}; \hat{Y}]$, $C = [I_n; H]$ and $\epsilon = [\epsilon^X; \epsilon^Y]$. Then, having

SNR(dB)	-6	-3	0	3	6
GE	0.6797	0.6668	0.6488	0.6242	0.5928
FE_{Avg}	0.5823	0.5673	0.5469	0.5212	0.4903
Erased Sub-Space	1	2	3	4	5
GE	0.4295	0.4276	0.4287	0.4861	0.4647
FE_{Avg}	0.4166	0.3882	0.4078	0.5503	0.4800

TABLE III: Performance of MDFE in the presence of noise and sub-space erasure (using Alg.2 with $L = 5$ on network in Figure 2).

LLSEE (GE)	EM (GE)	LLSEE (FE_{Avg})	EM (FE_{Avg})
0.4126	0.3215	0.3626	0.2956

TABLE IV: LLSEE and EM based TM estimation: A comparison (using Alg.2 with $L = 5$ on network in Figure 2).

covariance matrix $\mathcal{K} = E[\epsilon\epsilon^T]$; X is estimated by:

$$\hat{X} = \left(C^T \mathcal{K}^{-1} C\right)^{-1} C^T \mathcal{K}^{-1} V \quad (11)$$

To apply our MDFE framework on this setup, Alg.1 is used to partition the network into $L = 8$ sub-spaces. Then, Eq.(11) is properly adapted to solve the problem in each sub-space (where corresponding parameters C_i, $\mathcal{K}_i$ and V_i are used based on routing matrix $H_i(I_i, J_i)$ and observations $\hat{Y}_i$ and $\hat{X}_i$ in each sub-space). Accordingly, processing gains $\Delta_p = 88\%$ and $\Delta_p = 58\%$ are achieved, indicating that MDFE spped up the process, significantly. In addition, Figure 7 shows $Gain_{L2}$ at different SNRs, indicating that MDFE can improve the performance. This gain is remarkable at low NetFlow SNRs (i.e. low sampling rates) where sampling and storing overheads are challenging limitations for direct measurement of TMs. Therefore, MDFE can be utilized to propose a new hybrid TM measurement method where important TMs can be measured with higher sampling rates and MDFE is applied on the other TMs to improve the accuracy of TM estimation. This is of particular importance in today's network monitoring systems where sampling and storing a sheer volume of today's traffic and quick TM estimation are challenging problems, particularly for large scale and dynamic environments. Therefore, MDFE not only is compatible with the idea of using multiple sources of data [8], but also it can enhance its performance.

B. Traffic Matrix Completion

In [4], a Sparsity Regularized SVD (SRSVD) method is introduced for TM Completion (TMC) where the columns of traffic matrix Z is formed by the unknown vector X in our TME setup at different times ($t = 1, ..., \mathcal{T}$). Now, assuming Z can be factored as $Z_{n\times\mathcal{T}} = LR^T$; then TMC is formulated as the following optimization problem to find missed entries of Z.

$$\hat{Z} = \min_{L,R} \left\|\mathcal{B} - \mathcal{A}(LR^T)\right\|_F^2 + \lambda\left(\|L\|_F^2 + \|R\|_F^2\right) \quad (12)$$

Here, $\mathcal{B}$ and $\mathcal{A}$ respectively denote the set of measurements and a linear operator satisfying $\mathcal{A}(Z) = \mathcal{B}$. To apply our MDFE framework, we adopt this method and modified the formulation in Eq.(12). In our Modified SRSVD (MSRSVD) method, $A_t = [diag(M_t); H]$ and $b_t = [X_t. * M_t; Y_t]$ where M_t is a binary column vector (where zeros represent missing

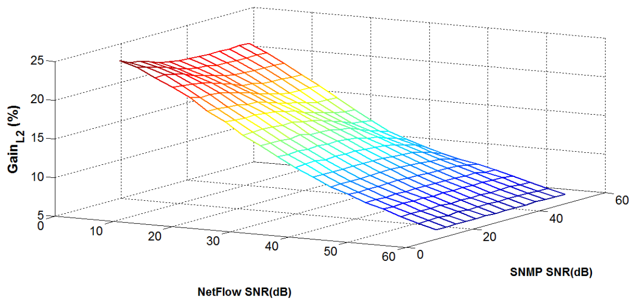


Fig. 7: $Gain_{L2}$ vs. SNR in NetFlow and SNMP.

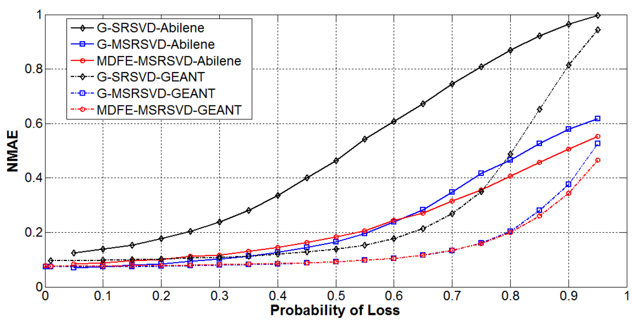


Fig. 8: NMAE v.s probability of loss in TM completion (G: global).

entries), H is the routing matrix, $Y_t = HX_t$ denotes t^{th} link load measurement vector and $.*$ denotes an element-wise product; accordingly, $\mathcal{A} = \{blockdiag(\mathcal{A}, A_t)\}_{t=1}^{\mathcal{T}}$, $\mathcal{B} = [b_1; ...; b_{\mathcal{T}}]$ and $\mathcal{M} = [M_1, ..., M_{\mathcal{T}}]$. Figure 8 shows that our new MSRSVD TM completion method significantly improves the performance where TMC is applied onto normalized TMs where X_{base} [4] is assumed to be known. It also compares the TM completion performance between Global-TMC and MDFE-TMC on real Abilene and GEANT networks and data. Here, Alg.1 is used for partitioning where $L^{Abilene} = 10$ and $L^{GEANT} = 6$ and we set MSRSVD as sub-space TMC technique in MDFE framework. Also, $\lambda_{Abilene} = 0.01$ and $\lambda_{GEANT} = 0.1$ and for both networks we fixed $r = 2$ (i.e. rank-2 approximation). The Normalized Mean Absolute Error (NMAE) is computed over interpolated values as $NMAE = \frac{\sum_{i,j:\mathcal{M}(\rangle,\rfloor)=0} |Z(i,j) - \hat{Z}(i,j)|}{\sum_{i,j:\mathcal{M}(i,j)=0} |Z(i,j)|}$. It is clear that, MDFE can improve the performance for high loss probabilities where MDFE framework reduces the number of unknowns in each sub-space and helps to improve spatial-temporal correlations. For low loss rates, the performance of both methods are close together. However, MDFE can speed-up the TMC process and improve its robustness.

C. Loss Inference

Considering $Y = HX$, loss inference is an UDLI problem where H is a routing matrix, and X and Y are defined as $X = \{x_j\}_{j=1}^n = \{log\hat{\phi}_{e_j}\}_{j=1}^n$ and $Y = \{y_i\}_{i=1}^m = \{log\hat{\phi}_i\}_{i=1}^m$. Here, $\hat{\phi}_i$ represents the fraction of S probes that arrive correctly at the destination and $\hat{\phi}_{e_j}$ is the fraction of probes from all paths passing through link e_j that have not been dropped by that link [7]. Here, a Loss Inference Algorithm (LIA) is

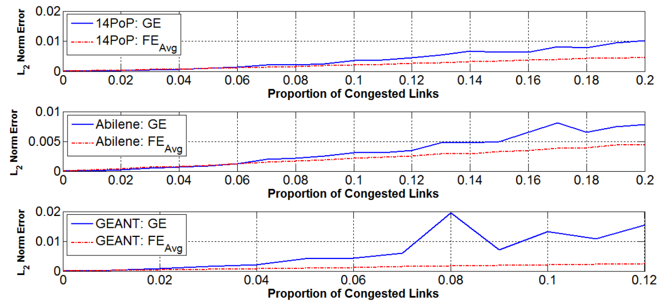


Fig. 9: GE and FE_{Avg} vs. the proportion of congested links where $L^{14PoP} = 5$, $L^{Abilene} = 5$, $L^{GEANT} = 14$, # of beacons S=1000 and # of iterations = 100.

adopted from [7] as the sub-space estimation technique to apply the MDFE framework. Three real network topologies are considered and the proportion of the links that are congested is fixed and is varied to evaluate the performance of MDFE framework in terms of GE and FE_{Avg}. Here, congested and non-congested links have loss rates uniformly distributed in [0.05, 0.2] and [0,0.002], respectively. Figure 9 shows the improvement achieved by applying MDFE for loss inference where Alg.2 is used to construct sub-problems. It is shown that by increasing the number of sub-spaces (L) the performance of MDFE is improved, and MDFE is more effective for higher loss-rates.

D. MDFE with Set-Covering

LLSEE is not effective in the presence of unusual inputs. Therefore, in many cases, L_1 and L_∞ constrained minimization techniques can be effectively applied to UDLI inference problems with heavy-tailed distributed inputs. These problems are generally solved using numerical optimization techniques. To show that MDFE framework is applicable for this set of problems, redundant set $\mathscr{C} = \bigcup_{i=1}^{L} I_i$ is defined to cover set I where $I_i \bigcap I_j$ $(i \neq j)$ is not necessarily empty. In fact, in this case, set-partitioning problem is changed to set-covering problem which is still an NP-hard problem for large-scale systems. Here, we consider GEANT network and its real data set which contains large amount of unusual inputs. To find cover $\mathscr{C}$, compatible with MDFE framework, we randomly choose subsets $\{I_i\}_{i=1}^{L}$ so that set I is covered, that is, $\mathscr{C} = I$ (more structured set-covering algorithms are under investigation). Then, using a small subset of inputs, GE and FE_{Avg} are computed and compared. This process is repeated to achieve desirable performance and the best cover is used to test the algorithm on the whole data set. Table V indicates the improvement achieved by applying MDFE framework where constrained optimization techniques represented by Eq.(13) (our more effective formulation) and Eq.(14) (adopted from [8]) are used and solved using CVX for TM estimation in sub-spaces.

$$\hat{X} = \min_{X} \|Y - HX\|_\infty \text{ s.t. } X \geq 0 \tag{13}$$

$$\hat{X} = \min_{X} \|Y - HX\|_1 + \lambda \|x\|_1 \text{ s.t. } X \geq 0 \tag{14}$$

Opt. Method	GE	FE_{Avg}	$Gain_{L2}(\%)$
Eq.(13)	0.5116	0.4800	6.1751
Eq.(14)	0.6093	0.5736	5.8616

TABLE V: MDFE performance using L_1 and L_∞ optimization techniques ($L = 5$).

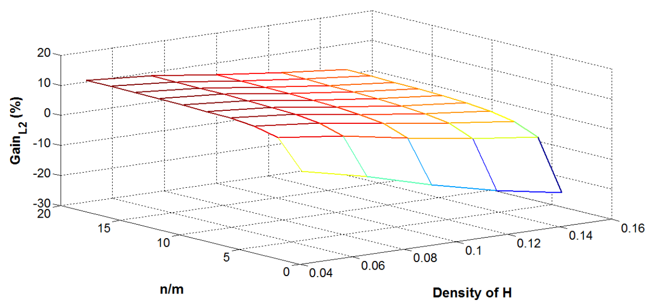


Fig. 10: $Gain_{L2}$ v.s. ratio $\frac{n}{m}$ and D (using Alg.1 and LLSEE where $X \sim U(0,1), m = 50, L = 5$).

E. MDFE on Random Observation Matrices

To show that MDFE can be applied on a wider range of problems, we did an extensive Monte-Carlo simulation over random-binary observation matrices. Figure 10 shows that MDFE can significantly improve the performance for fat (large $\frac{n}{m}$) and low-density matrices where partitioning helps to construct sub-spaces with smaller number of coherent unknowns, observed in different sub-spaces, with the capability of producing more precise estimates (since CN of sub-spaces are improved). All three networks used in our study are fat and low-density matrices.

V. Conclusion

In this paper, a novel approach for solving UDLI problems was introduced where a large-scale sparse problem is partitioned and solved in sub-spaces. By fusion the solution from sub-spaces, we not only showed that the accuracy of the solution is improved, but also the efficiency (processing time/power reduction) and robustness of system are enhanced. These are important factors, particularly, in distributed and dynamic environments where accurate, quick and efficient inference are highly demanding. We examined the performance of MDFE in different applications, and we showed that MDFE framework is flexible and compatible with inputs with different distributions and with a variety of sub-space estimation techniques, and it is match with today's multi-processors computing architectures.

References

[1] Y. Vardi, "Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data", Journal of American Statistical Association, p. 365-377, 1996.

[2] R. Castro, M. Coates, G. Liang, R. Nowak, B. Yu, "Internet Tomography: Recent Developments", Statistical Science, Vol. 19, No. 3, 2003.

[3] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, C. Diot, "Traffic Matrix Estimation:Existing Techniques and New Directions", In Proc. of ACM SIGCOMM'02.

[4] Y. Zhang, M. Roughan, W. Willinger, L. Qiu, "Spatio-Temporal Compressive Sensing and Internal Traffic Matrices", In Proc of ACM SIGCOMM, 2009.

[5] "Multicast-based inference of network-internal characteristics", http://gaia.cs.umass.edu/minc.

[6] N.G. Duffield, "Network Tomography of Binary Network Performance Characteristics", IEEE Transactions on Information Theory, 52(12):53735388, Dec. 2006.

[7] H.X. Nguyen, P. Thiran, "Network Loss Inference with Second Order Statistics of End-to-End Flows", ACM-IMC, 2007.

[8] Q. Zhao, Z. Ge, J. Wang, J. Xu, "Robust Traffic Matrix Estimation with Imperfect Information: Making use of Multiple Data Sources", In Proc. of SIGMETRICS06.

[9] A. Nucci, K. Papagiannaki, "Design, Measurement and Management of Large-Scale IP Networks", Cambridge, 2009.

[10] D. Ghita, Nguyen, M. Kurant, K. Argyraki, P. Thiran, "Netsope: Practical Network Loss Tomography," IEEE INFOCOM, 2010.

[11] J. Cao, D. Davis, S.V. Wiel, B. Yu, "Time Varying Network Tomography: Router Link Data", Journal of American Statistical Association, pages 1063-1075, 2000.

[12] M. Malboubi, C.N. Chuah, P. Sharma, "Decentralizing Network Inference Problems using Multiple-Description Fusion Estimation (MDFE) with Emphasis on Traffic Matrix Estimation", Technical Report, www.ece.ucdavis.edu/cerl/techreports/2012-2/

[13] O. Goldschmidt, "ISP Backbone Traffic Inference Methods to Support Traffic Engineering", In Internet Stat. and Metrics Analysis (ISMA), San Diego, CA, Dec 2000.

[14] C. Tebaldi, M. West, "Baysesain Inference of Network Traffic using Link Count Data", Journal of American Statistical Association, pages 557-573, June 1998.

[15] J.W. Demmel,"Applied Numerical Linear Algebra",SIAM,1997.

[16] S. Uhlig, B. Quoitin, S. Balon, J. Lepropre, "Providing Public Intradomain Traffic Matrices to the Research Community", ACM SIGCOMM CCR, 36(1), 2006.

[17] www.cs.utexas.edu/ yzhang/research/AbileneTM.