Practical Approach to Identifying Additive Link Metrics with Shortest Path Routing

Xiong Wang[1] Mehdi Malboubi[2] Sheng Wang[1] Shizhong Xu[1] Chen-Nee Chuah[2]
[1]Key Lab of Optical Fiber Sensing & Communications
School of Communication & Information Engineering
University of Electronic Science & Technology of China, Chengdu, China
{wangxiong, wsh_keylab, xsz}@uestc.edu.cn
[2]Department of Electrical & Computer Engineering
University of California, Davis, USA
{mmalboubi, chuah@ucdavis.edu}@ucdavis.edu

***Abstract*—We revisit the problem of identifying link metrics from end-to-end path measurements in practical IP networks where shortest path routing is the norm. Previous solutions rely on explicit routing techniques (e.g., source routing or MPLS) to construct independent measurement paths for efficient link metric identification. However, most IP-networks still adopt shortest path routing paradigm, while the explicit routing is not supported by most of the routers. Thus, this paper studies the link metric identification problem under shortest path routing constraints. To uniquely identify the link metrics, we need to place sufficient number of monitors into the network such that there exist m (the number of links) linear independent shortest paths between the monitors. In this paper, we first formulate the problem as a mixed integer linear programming problem, and then to make the problem tractable in large networks, we propose a Monitor Placement and Measurement Path Selection (MP-MPS) algorithm that adheres to shortest path routing constraints. Extensive simulations on random and real networks show that the MP-MPS gets near-optimal solutions in small networks, and MP-MPS significantly outperforms a baseline solution in large networks.**

I. Introduction

Network providers need to keep track of the state (e.g., delay and loss probability of links) of their network to ensure they can meet the service level agreements (SLAs). A naive approach for monitoring network performance is to directly collect the status information of individual network element. However, direct measurement is costly and ~~may be not~~ feasible due to the lack of measurement support at individual network elements or scalability issues [1]. Thus, an alternative way is to infer the link metric/status by measuring the performance of selected end-to-end paths between a subset of nodes with monitoring capability (monitors), which is referred as end-to-end path based link metric identification problem [1], [3].

The existing solutions for the link metric identification problem can be classified into two general categories: statistical and algebraic approaches. The statistical approaches [4], [5] assume that the link metrics follow some probability distributions, and use various statistical inference techniques to estimate the link metric distributions from measured path metrics. In many cases, the link metrics are additive (e.g., delay) or multiplicative (e.g., loss probability). Since a multiplicative metric can be expressed in an additive form by using the $\log(\cdot)$ function, we can treat the additive and multiplicative metrics equivalently. If the link metrics are additive, the algebraic approaches [6], [7] model the end-to-end path based link metrics identification problem as a system of linear equations, where the unknown variables are the link metrics, and the known constants are the end-to-end path measurements. The goal of the algebraic approaches is to find an algebraic solution for the linear system.

Unlike statistical approaches, the algebraic approaches can accurately and uniquely identify the link metrics. If all nodes are allowed to participate in the measurement process, the link metrics can be easily measured by exploiting multicast trees [8], [9], [10]. But if only selected nodes (monitors) can participate in the measurement process, as assumed in this paper, the problem becomes challenging [7], [11], [12], [13], [14]. Gurewitz and Sidi [7] prove that the directed network (links in different directions have different metrics) is unidentifiable unless every node is a monitor. For the undirected networks, Gopalan and Ramasubramanian [11] give the necessary and sufficient conditions to identify link metrics by using cycles and proposes an efficient algorithm to construct linear independent measurement cycles or paths containing cycles. The same authors also propose an algorithm to find maximum number of linear independent paths/cycles that can be constructed between the given monitors [12]. Since routing along cycles is always prohibited by routing protocols, Ma et al. [13] gives the necessary and sufficient conditions to identify link metrics by using simple paths (paths without cycles), and develop a monitor placement algorithm based on these conditions. Ma et al. [14] propose an efficient algorithm to construct independent measurement paths.

However, the afore-mentioned work [7], [11], [12], [13], [14] must rely on the explicit routing techniques (e.g., source routing or MPLS) to establish independent measurement paths. This is a huge assumption that may render the solutions impractical since most IP-networks adopt shortest path routing paradigm, while the explicit routing is not supported by most of the routers [15]. Thus in this paper, we address the end-to-end path based link metric identification problem under the shortest path constraint, and to the best of our knowledge, our work is the first to consider the problem. We first formulate the problem as a Mixed Integer linear Programming (MIP) problem. Then to efficiently solve the problem in large networks, we propose an heuristic algorithm to place monitors and select linear independent shortest measurement paths between these monitors. Finally, we evaluate our proposed algorithm through simulations on both ISP real networks and synthetic networks.

The rest of the paper is organized as follows. In section II, we first describe the network model and assumptions, and then we present our MIP formulation. Section III presents the proposed monitor placement and measurement path selection algorithm. In section IV, we show the simulation results. Section V concludes this work.

II. Problem Formulation

In this section, we first briefly introduce the network model and assumptions considered in this paper, and then we formulate the end-to-end shortest path based link metric identification problem as a MIP.

A. Network Model and Assumptions

We model the IP network as a connected undirected graph $G(V, L)$, where V is the set of nodes (routers) and L is the set of links. Let $n = |V|$ and $m = |L|$ denote the number of nodes and links, respectively. Each link $l \in L$ is associated with an unknown performance metric x_l and a given routing weight w_l. In real networks running shortest path routing protocols, each link will be traversed by at least one of the shortest paths.

We assume that some nodes in the network can be directly connected to monitors, which can send and receive probe packets. Measurement paths, which start and end at distinct monitors, are the shortest paths determined by the given link weights. If there are multiple equal cost shortest paths between two monitors, we assume that the two monitors can measure all of the equal cost shortest paths between them by sending probe packets with different five-tuples flow identification. Let P be the set of all shortest paths in the network. Let $\mathbf{X} = (x_1, x_2, ..., x_m)^T$ denote the vector of link metrics and $\mathbf{Y} = (y_1, y_2, ..., y_c)^T$ denote the vector of the available shortest path measurements, where c is the number of shortest measurement paths. Based on the above assumptions and notations, the measurement problem can be represented by the linear system $\mathbf{AX} = \mathbf{Y}$, where $\mathbf{A}$ is the routing matrix whose (i, j)th entry is a binary representing whether link j appears in path i.

Evidently, to uniquely determine $\mathbf{X}$, $\mathbf{A}$ must have full rank, i.e., rank($\mathbf{A}$)= m. In other words, we must find m linear independent shortest paths between monitors to take measurement. If every node is monitor, $\mathbf{A}$ is a identity matrix and $\mathbf{X}$ simply equals $\mathbf{Y}$. But in order to reduce measurement cost, we need to minimize the number of monitors placed in the network. So the objective of this paper is to find a placement of the minimum number of monitors in $G(V, L)$ that enables the unique determination of all link metrics by measuring the m linear independent shortest paths between these monitors.

B. MIP Formulation

For ease of description, we first present parameters and variables used in the MIP.

$\mathbf{a}_p = [a_{p1}, a_{p2}, \cdots, a_{pl}, \cdots, a_{pm}]^T$: The binary vector for the shortest path p, where a_{pl} is a binary constant used to indicate whether link l is included by the shortest path p. i.e., a_{pl} equals 1 if link $l \in p$, and 0 otherwise.

$\mathbf{b}_l = [b_{l1}, b_{l2}, \cdots, b_{li}, \cdots, b_{lm}]^T$: The binary vector for link l. $\mathbf{b}_l$ is a binary vector, where b_{li} equals 1 if $i = l$, and 0 otherwise.

δ_k^p: A binary constant indicates whether node k is an endpoint of the shortest path p, i.e., δ_k^p equals 1 if if node k is an endpoint of the shortest path p, and 0 otherwise.

C: A large constant.

σ_p^l: A real variable, which denotes the linear commbination coefficient of vector $\mathbf{a}_p$ used to linearly represent vector $\mathbf{b}_l$.

u_p: A binary variable, which denotes whether the shortest path p is chosen for measurement, i.e., u_p equals 1 if the shortest path p is selected, and 0 otherwise.

v_k: A binary variable, which indicates whether node k is selected to place monitor, i.e., v_k equals 1 if node k is selected to place monitor, and 0 otherwise.

z: A integer variable, which denotes the number of monitors needed to be placed in the network.

As discussed in the previous subsection, the objective of our problem is to minimize the number of monitors placed in the network. Thus, the objective can be formulated as:

$$minimize \quad z = \sum_{k=1}^{n} v_k \tag{1}$$

To uniquely identify the link metrics, we need to select m linear independent shortest paths between the monitors. It means that the binary vector $\mathbf{b}_l$ for each link $l \in L$ can be represented as a linear combination of the binary vectors of the m shortest paths.

$$\sum_{p=1}^{|P|} \sigma_p^l \cdot \mathbf{a}_p = \mathbf{b}_l, \qquad \forall p \in P, l \in L \tag{2}$$

If the shortest path p is selected as one of the m linear independent paths, at least one of the linear combination coefficients for the shortest path p (σ_p^l) is not equal 0. So we can use the following constraints to ensure that $u_p = 1$ if $|\sigma_p^l| > 0 \ \forall l \in L$.

$$\sigma_p^l \leq u_p \cdot C, \qquad \forall l = 1, 2, \cdots, m, \forall p \in P \tag{3}$$

$$-\sigma_p^l \leq u_p \cdot C, \qquad \forall l = 1, 2, \cdots, m, \forall p \in P \tag{4}$$

To minimize the measuring cost, we only need to select m linear independent measurement paths.

$$\sum_{p=1}^{|P|} u_p = m \tag{5}$$

If the shortest path p is selected as a measurement path, the endpoints of path p must be selected to place monitors.

$$v_k \cdot C \geq \sum_{p=1}^{|P|} u_p \cdot \delta_k^p, \quad \forall k = 1, 2, \cdots, n \tag{6}$$

The complexity of a MIP is known to be exponential, i.e., $O(2^N)$, where N is the number of integer variables. Thus the MIP model presented above has an exponential complexity with N in $O(|P|)$, which makes it computationally expensive and even infeasible in large networks. Hence, to solve the end-to-end shortest path based link metric identification problem efficiently in large network, we propose a heuristic algorithm in the next section.

III. The Monitor Placement and Measurement Path Selection Algorithm

In order to uniquely identify the link metrics in the IP network, we need to place some monitors on the network nodes and select m linear independent shortest measurement paths between these monitors. This section introduces our proposed Monitor Placement and Measurement Path Selection (MP-MPS) algorithm for shortest routing based IP networks.

A. Algorithm Description

MP-MPS places monitors and selects measurement paths simultaneously. For ease of description, we first introduce the following definition.

Definition 1 (identifiable and unidentifiable link) *Given a set of linear independent paths, if the metric of a link can be uniquely inferred by measuring the set of paths, the link is identifiable, otherwise, the link is unidentifiable.*

Algorithm 1 shows the detailed description of MP-MPS. Initially, MP-MPS uses MMP (Minimum Monitor Placement) algorithm [13] to place the minimum number of monitors needed to identify link metrics when explicit routing is allowed. However, under the shortest path routing constraint, it is generally impossible to identify all link metrics by only using the monitors placed by MMP, i.e., the number of linear independent shortest paths between these monitors is less than m. Therefore, MP-MPS will select more nodes to place monitors in the following steps. Firstly, MP-MPS selects the linear independent paths from the shortest paths between the placed monitors and adds these paths to linear independent shortest path set P (lines 2-4). To select a set of linear independent paths, we use the *Algorithm 2*, which is a variant of QR decomposition with column pivoting [6], [16]. In *Algorithm 2*, $\|\cdot\|_2^2$ denotes square of the 2-norm of a vector. *Algorithm 2* incrementally decomposes the routing matrix A into QR ($Q \in \mathbb{R}^{m\times h}, R \in \mathbb{R}^{h\times h}$), where Q is a matrix with orthonormal columns, R is an upper triangular matrix and h is the number of linear independent paths that have been selected. Then for an unidentifiable link (u, v) (the path only traversing link (u, v) is linear independent with the paths in set P) , MP-MPS will sequentially select nodes u and v as monitors (lines 8-18), and the shortest paths, which are starting from u or v to the selected monitors and linear independent with selected paths, will be added to set P (lines 19-21). MP-MPS terminates when the number of selected linear independent paths equals m (lines 6 and 23).

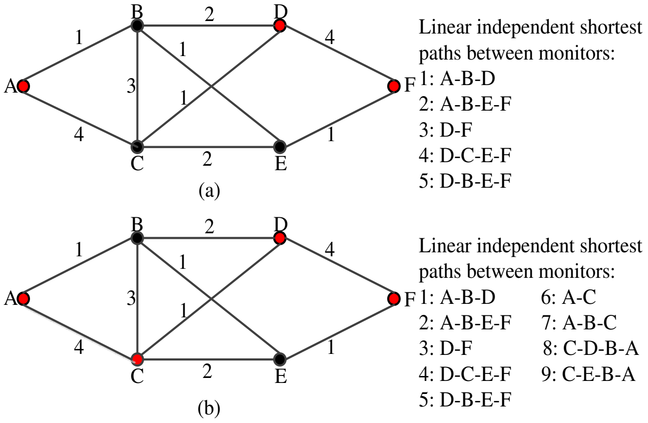

Fig. 1: An Illustrative example for MP-MPS

B. An Illustrative Example

We demonstrate how the MP-MPS algorithm works with an example shown in Fig. 1, where the numbers on the links represent the link weights used by shortest path routing protocols. MP-MPS first uses the MMP proposed in [13] to place monitors on nodes A, D and F (Fig. 1(a)). If the explicit routing is allowed, the link metrics can be uniquely identified by constructing nine linear independent paths starting and ending at nodes A, D and F [14]. However, if only the shortest measurement paths are allowed, there are only five linear independent shortest paths (paths 1-5 in Fig.1(a)) starting and ending at nodes A, D and F. So in order to uniquely determine the link metrics, MP-MPS needs to select more nodes to place monitors. In the following steps, MP-MPS checks the links sequentially to find an unidentifiable link. We assume the first unidentifiable link found by MP-MPS is (A, C). Thus, node C is selected as a monitor (MP-MPS ignores node A since it is already selected as a monitor), and four linear independent shortest paths (paths 6-9 in Fig.1(b)) between nodes A and C are added to the measurement path set. After that, MP-MPS terminates since the number of linear independent measurement paths equal to the number of links.

C. Complexity Analysis

In *Algorithm 1*, line 1 takes $O(n + m)$ time [13]. The complexity of checking whether a path is linear independent with a set of paths by using *Algorithm 2* is $O(m^2)$. In the worst case, MP-MPS will check all of the shortest paths in the network (lines 4, 10, 11 and 20). Therefore, the entire algorithm has time complexity of $O(K \cdot m^2)$, where K is the number of shortest paths in the network. It is notable that in real IP networks, the routing matrix A is very sparse. We can leverage this property to speed up the implementation of the algorithm.

Algorithm 1 Monitor Placement and Measurement Path Selection (MP-MPS)

```
Input: Network topology $G(V, L)$
Output: A subset $M$ of nodes in $V$ as monitors, and a set $P$ of $m$ linear independent shortest paths between monitors.
Invoke MMP algorithm [13] to select monitors and add these monitors to $M$
add all of the shortest paths between monitors in $M$ to set $SP$
$A \leftarrow NULL$, $R \leftarrow NULL$
$(P, A, R) \leftarrow$ Algorithm 2 $(SP, A, R)$
if $|P| == M$ then
    return $(M, P)$
end if
for each link $(u, v) \in L$ do
    Let path $p \leftarrow \{u, v\}$
    $R_{12} \leftarrow R^{-1^T} A a_p^T = Q^T a_p^T$
    $R_{22} \leftarrow \|a_p^T\|_2^2 - \|R_{12}\|_2^2$
    if $R_{12} \neq 0$ then
        Push nodes $u$ and $v$ to stack $S$
    end if
    while $S$ is not empty do
        $z \leftarrow pop(S)$
        if $z \notin M$ then
            Select $z$ as a monitor and put $z$ into $M$
            Let $SP \leftarrow \emptyset$ and add the shortest paths from node $u$ to the nodes in $M$ to set $SP$
            $(TP, A, R) \leftarrow$ Algorithm 2 $(SP, A, R)$
            $P \leftarrow P \bigcup TP$
            if $|P| == m$ then
                return $(M, P)$
            end if
        end if
    end while
end for
```

IV. Performance Evaluation

To evaluate the performance of MP-MPS, we conduct a set of simulations on randomly generated topologies and real network topologies obtained from the Rocketfuel project [17]. To demonstrate the optimality of MP-MPS, we first compare the performance of the MIP model and MP-MPS in small random topologies. Afterwards, similar to [13], we use the Random Monitor Placement and Measurement Path Selection (RMP-MPS) as a benchmark for MP-MPS in large random and real topologies, where the MIP model becomes intractable. Given the number of monitors k (k is greater or equal to the number of monitors needed when explicit routing is allowed), RMP-MPS uses the following steps to place monitors and select measurement paths:

Step 1: Use MMP [13] to place monitors in $G(V, L)$ and put the nodes with monitors into set M

Step 2: Randomly select $k - |M|$ nodes from set $V \setminus M$ to place monitors and append these nodes to set M.

Step 3: Select the linear independent measurement paths from the shortest paths between monitors.

Step 4: Remove the nodes that are not the termination nodes of the selected measurement paths in Step 3 from M.

Evidently, RMP cannot guarantee to uniquely identify the metric of every link for arbitrary $G(V, L)$ and k. Therefore, similar to [13], we evaluate its performance by the fraction of random placements achieving full link metric identifiability over multiple Monte Carlo runs, which is referred as probability of achieving full identifiability.

Algorithm 2 Linear Independent Path Selection (LIPS)

```
Input: A set $CP$ of candidate paths, and matrices $A$ and $R$
Output: A set $LP$ of linear independent paths in $CP$, and the updated matrices $A$ and $R$
for each path $p \in CP$ do
    if $A == NULL$ then
        $A \leftarrow \left[a_p\right]$ and $R \leftarrow \left[\|a_p\|_2^2\right]$
    else
        $R_{12} \leftarrow R^{-T} A a_p^T = Q^T a_p^T$
        $R_{22} \leftarrow \|a_p^T\|^2 - \|R_{12}\|_2^2$
        if $R_{12} \neq 0$ then
            Add path $p$ to set $LP$
            Update $R \leftarrow \begin{bmatrix} R & R_{12} \\ 0 & R_{22} \end{bmatrix}$ and $A \leftarrow \begin{bmatrix} A \\ a_p \end{bmatrix}$
        end if
    end if
end for
return $(LP, A, R)$
```

A. Random Topologies

We use ERdős-Rényi (ER) and Barabási-Albert (BA) models to generate random topologies. In ERdős-Rényi model, a topology is constructed by independently connecting each pair of nodes by a link with a fixed probability p. BA model generate random topology by beginning with an initially connected topology of n_0 ($n_0 = 4$ in our simulations) nodes and adding new nodes sequentially. Each new node is connected to d_{min} existing nodes with a probability that is proportional to the degree of the existing nodes. For simplicity, we set the weight of every link to 1. In each simulation, we use the 100 randomly generated topologies to evaluate the performance of the algorithm.

(1) *Comparison of MP-MPS and MIP*

Since the complexity of MIP is exponential, MIP can only get optimal solutions within acceptable time in small topologies. We generate random topologies with 11 nodes. To evaluate the performance of the algorithms under different topology characteristics, we conduct simulations on two types of random topologies: sparsely connected topologies (for ER model, $p = 0.2$ and for BA model, $d_{min} = 2$) and densely connected topologies (for ER model, $p = 0.4$ and for BA model, $d_{min} = 3$). Fig. 2 shows the simulation results on sparsely connected random topologies. As shown in Fig. 2, the probability that MP-MPS and MIP are able to identify all the link metrics is quite close, e.g., the difference is less than 0.1 in both ER and BA topologies. It demonstrates that the monitors placement solutions found by MP-MPS are near-optimal in small topologies. We also can observe that when $k \leq 9$, RMP-MPS achieves identifiability on fewer than 30% of the topologies, whereas MP-MPS ensures identifiability on about 90% of the topologies. It means MP-MPS substantially outperforms RMP-MPS in small topologies. Fig. 3 shows the simulation results on densely connected small topologies. In

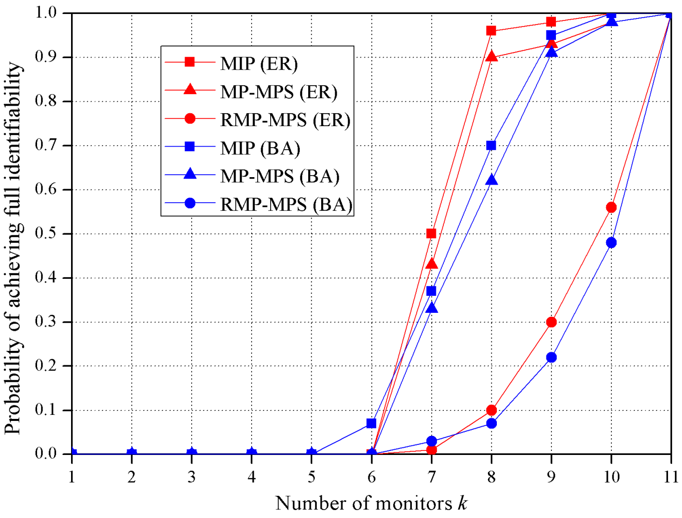

Fig. 2: Comparison of MP-MPS, MIP and RMP-MPS on small sparse random topologies

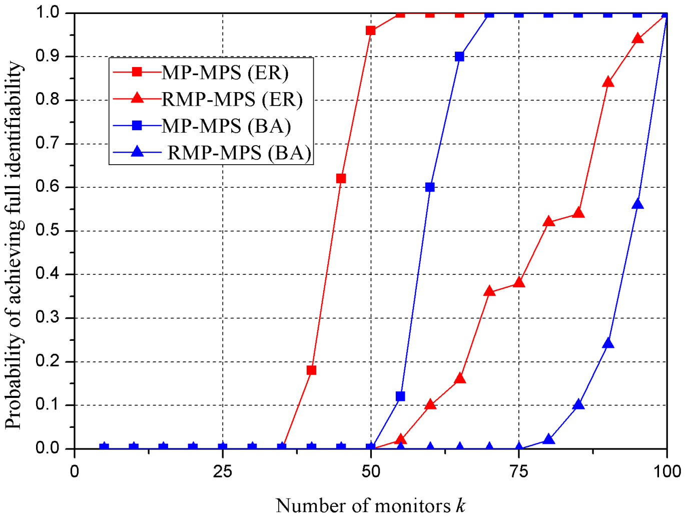

Fig. 4: Comparison of MP-MPS and RMP-MPS on large sparse random topologies

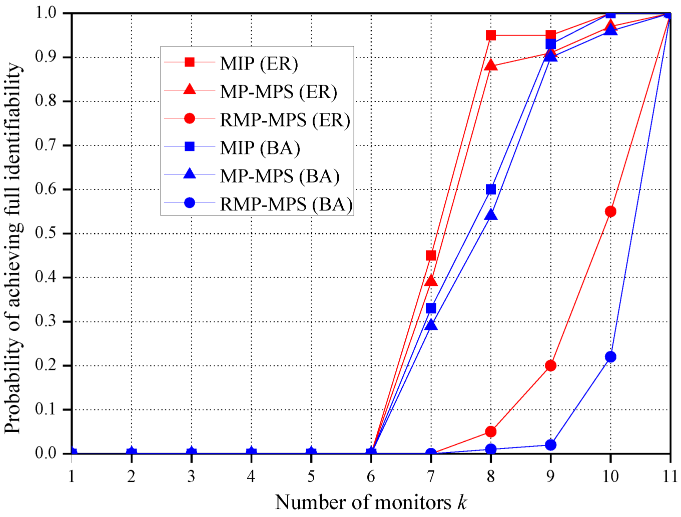

Fig. 3: Comparison of MP-MPS, MIP and RMP-MPS on small dense random topologies

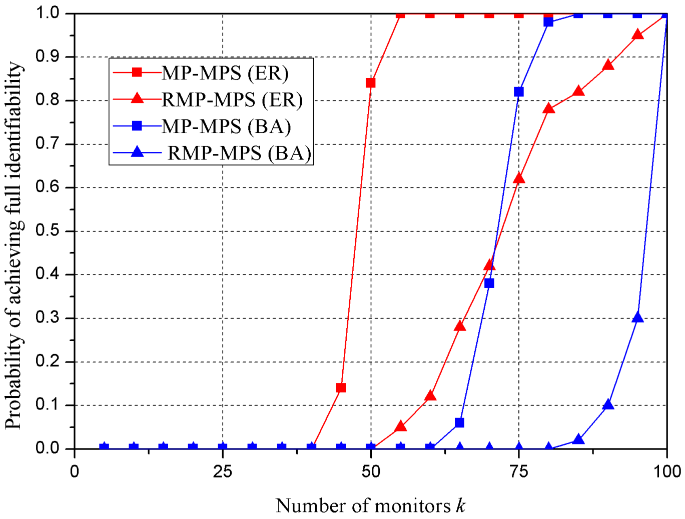

Fig. 5: Comparison of MP-MPS and RMP-MPS on large dense random topologies

general, we can see that the results in Fig. 3 and Fig. 2 exhibit the same trends. However, It is notable that MIP, MP-MPS and RMP-MPS perform worse in densely connected small topologies. This is because the number of monitors required to achieve identifiability always increases with the link and shortest path ratio r, which is defined as $r = m/|P|$ ($|P|$ is the number of shortest paths in the network). And in our simulations, densely connected topologies have higher r than sparsely connected topologies.

(2) *Comparison of MP-MPS and RMP-MPS*

We compare the performance of MP-MPS and RMP-MPS in large random topologies with 100 nodes. In the simulations, we also generate sparsely connected random topologies (for ER model, $p = 0.05$ and for BA model, $d_{min} = 2$) and densely connected random topologies(for ER model, $p = 0.08$ and for BA model, $d_{min} = 3$). Fig. 4 shows the simulation results on sparsely connected random topologies. We can observe that as expected, the probability that MP-MPS and RMP are able to identify all the link metrics increases with the number of placed monitors. However, when $k \leq 60$ in ER topologies and $k \leq 85$ in BA topologies, RMP-MPS achieves full link metrics identification only on about 10% of the topologies, whereas when $k \leq 55$ in ER topologies and $k \leq 70$ in BA topologies, MP-MPS achieves full link metrics identification on all of the topologies. It is demonstrated that MP-MPS significantly outperforms the RMP-MPS in most cases.

Fig. 5 shows the simulation results on densely connected topologies. Similar to the simulation results in small random topologies, both MP-MPS and RMP-MPS perform worse in densely connected topologies, requiring more monitors to

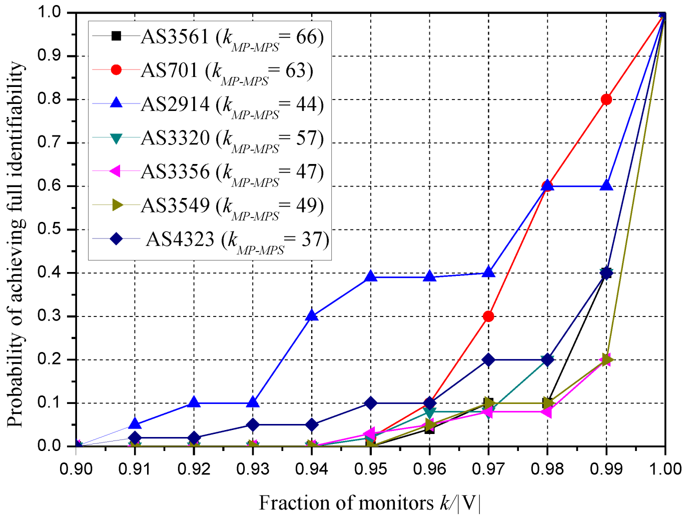

Fig. 6: Comparison of MP-MPS and RMP-MPS on real topologies

achieve the same probability of full link metrics identification. We also can see from Fig. 4 and Fig. 5 that both MP-MPS and RMP need more monitors to achieve full link metrics identification in BA topologies. This is because ER topologies have more equal cost shortest paths than BA topologies.

B. Real Topologies

We also conduct a set of simulations on seven ISP topologies with link weights derived by the Rocketfuel project [17]. Table I summarizes the number of nodes and links in each topology. We repeat RMP-MPS 1000 times in each ISP topology. Let k_{MP-MPS} denote the number of monitors required by MP-MPS to fully identify the ISP networks. For ease of comparison, we mark k_{MP-MPS} in the legends of the Fig. 6. As shown in Fig. 6, MP-MPS needs a significant fraction of nodes to be monitors in ISP topologies, roughly ranging from 70% to 80%. One reason is that in these real topologies a large number of nodes have degree than less than 3, which have to be selected as monitors. It is verified in [13] that in some real topologies, more than 60% nodes need to be monitor even if the explicit routing is allowed. However, we also can observe that in real topologies, MP-MPS also needed much less monitors than RMP-MPS.

At last, Table II depicts the average length (i.e., the number of hops) of measurement paths and the running time of MP-MPS in ISP topologies. In all of the ISP topologies, the average length of measurement paths selected by MP-MPS is about 3 hops, and the running time of MP-MPS on a desktop with $3GHz$ CPU and $4GB$ memory is less $10s$, which indicates that MP-MPS is time efficient.

Topology	AS4323	AS3549	AS3356	AS3320	AS2914	AS701	AS3561
Nodes	51	61	63	70	70	83	92
Links	161	486	285	355	111	219	329

TABLE I: Real Topologies Used in Simulation.

V. Conclusion

We studied the problem of identifying link metrics using end-to-end measurements along shortest paths between monitors. In this paper, we first formulated the problem as a MIP, and then we proposed an efficient algorithm named MP-MPS for placing monitors and selecting shortest measurement paths. Extensive simulation results shown that MP-MPS gets near-optimal solutions in small topologies, and MP-MPS also significantly outperforms a baseline algorithm in both large random topologies and real topologies.

Acknowledgment

We would also like to thank the anonymous reviewers for their valuable comments. This work is partially supported by NSF CNS-1321115, 973 Program (2013CB329103), NSFC Fund (61301153, 61271165, 61201128).

References

[1] F. Lo Presti, N. Duffield, J. Horowitz, and D. Towsley, Multicast-based inference of network-internal delay distributions, *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, pp. 761775, 2002.

[2] M. Coates, A. O. Hero, R. Nowak, and B. Yu, Internet tomography, *IEEE Signal Processing Magzine*, vol. 19, pp. 4765, 2002.

[3] E. Lawrence and G. Michailidis, Network tomography: A review and recent developments, *Frontiers and Statistics*, vol. 54, 2006.

[4] F. Lo Presti, N. Duffield, J. Horowitz, and D. Towsley, Multicast- based inference of network-internal delay distributions, *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, pp. 761-775, 2002.

[5] Y. Xia and D. Tse, Inference of link delay in communication networks, *IEEE Journal of Selected Areas in Communications*,vol. 24, no. 12, pp.2235-2248, 2006.

[6] Y. Chen, D. Bindel, and R. H. Katz, An algebraic approach to practical and scalable overlay network monitoring, in *ACM SIGCOMM*, 2004

[7] O. Gurewitz and M. Sidi, Estimating one-way delays from cyclic-path delay measurements, in *IEEE INFOCOM*, 2001.

[8] A. Adams, T. Bu, T. Friedman, J. Horowitz, D. Towsley, R. Caceres, N. Duffield, F. Presti, and V. Paxson, The use of end-to-end multicast measurements for characterizing internal network behavior, *IEEE Communications Magazine*, vol. 38, no. 5, pp. 152159, 2000.

[9] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, Network tomography: recent developments, *Statistical Science*, 2004.

[10] M. F. Shih and A. Hero, Unicast inference of network link delay distributions from edge measurements, in *IEEE ICASSP*, 2001.

[11] A. Gopalan and S. Ramasubramanian, On identifying additive link metrics using linearly independent cycles and paths, *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, 906-916, 2012.

[12] A. Gopalan and S. Ramasubramanian, On the maximum number of linearly independent cycles and paths in a network," to appear on *IEEE/ACM Transactions on Networking*, 2013.

[13] L. Ma, T. He, K. K. Leung, A. Swami and D. Towsley, Identifiablity of link metrics based on end-to-end path measurements, in *IMC*, 2013.

[14] L. Ma, T. He, K. K. Leung, D. Towsley and A. Swami, Efficient identification of additive link metrics via network tomography in *IEEE ICDCS*, 2013

[15] J. Sommers, B. Eriksson and P. Barford, "On the prevalence and characteristics of MPLS deployments in the open Internet," in *IMC*, 2011.

[16] G.H. Golub and C.F. Van Loan, "Matrix Computations," The Johns Hopkins University Press, 1989.

[17] R. Sherwood, A. Bender, and N. Spring, Measuring ISP Topologies with Rocketfuel, in *ACM SIGCOMM*, 2002.

Topology	AS4323	AS3549	AS3356	AS3320	AS2914	AS701	AS3561
Average length	3.2	3	3	2.9	4.1	3	3.5
Running Time	0.86s	9.4s	2.9s	7.4s	0.87s	2.7s	6.8s

TABLE II: Average length of measurement paths and running time of MP-MPS.