Multi-layer Monitoring of Overlay Networks*

Mehmet Demirci[1], Samantha Lo[1], Srini Seetharaman[2], and Mostafa Ammar[1]

[1] School of Computer Science, Georgia Institute of Technology, Atlanta GA 30332
[2] Deutsche Telekom R&D Lab, Los Altos CA 94022, USA
{mdemirci,samantha,ammar}@cc.gatech.edu, srini.seetharaman@telekom.com

Abstract. Monitoring end-to-end paths in an overlay network is essential for evaluating end-system performance and for troubleshooting anomalous behavior. However, conducting measurements between all pairs of overlay nodes can be cumbersome and expensive, especially in a large network. In this paper, we take a different approach and explore an additional degree of freedom, namely, monitoring native links. We allow native link measurements, as well as end-to-end overlay measurements, in order to minimize the total cost of monitoring the network. We formulate an optimization problem that, when solved, identifies the optimal set of native and overlay links to monitor, and a feasible sequence of arithmetic operations to perform for inferring characteristics of the overlay links that are not monitored directly. We use simulations to investigate how various topological properties may affect the best monitoring strategy. We also conduct measurements over the PlanetLab network to quantify the accuracy of different monitoring strategies.

1 Introduction

Monitoring all links in infrastructure overlay networks with persistent nodes is necessary to assess the overall performance of the users and to detect anomalies. Since an *overlay link* is in reality an end-to-end native path spanning one or more native links, this full monitoring operation can constitute a significant overhead (in terms of bandwidth and processing) for large overlays, especially if the monitoring is performed by active measurements.

In this paper, we alleviate the overlay network monitoring problem by adopting a more flexible approach that allows certain native link measurements in addition to end-to-end measurements[1]. These native link measurements can be used to infer desired metrics for overlay links by suitable combinations of native layer metrics. We call this approach *multi-layer monitoring.* This framework allows for four different options:

1. **Monitor all overlay links:** With this strategy, all overlay links are monitored directly and individually.

* This work was supported in part by NSF grant CNS-0721559.

[1] Our work pertains to infrastructure overlays, rather than peer-to-peer networks.

S.B. Moon et al. (Eds.): PAM 2009, LNCS 5448, pp. 77–86, 2009.

2. **Monitor a *basis set* of overlay links:** The work in [8] introduces a method to select and monitor a minimal subset of overlay links called the *basis set.* The characteristics of the remaining overlay links are inferred from the measurements for the basis set.
3. **Monitor all native links:** Another option is to monitor all the underlying native links in the network. Afterwards observed native layer metrics are combined to produce the results for all the overlay links.
4. **Monitor a mix of native links and overlay links (Multi-layer Monitoring):** In this option proposed in this paper, we monitor some native links and a subset of the overlay links. We then infer the remaining overlay links by combining these observations.

Note that while options 2-4 have the potential to reduce the monitoring cost, they are also prone to *inference errors* when an overlay link measurement is inferred from measurements on native and/or other overlay links.

The multi-layer monitoring strategy (option 4) is the most general one and subsumes all others. It also affords significant flexibility in monitoring overlays. Our objective in this work is to minimize monitoring cost by determining the optimal mix between overlay and native layer monitoring. To this end we formulate this as an optimization problem and discuss some features of its solution.

Previous work has considered overlay network monitoring and developed various approaches for it. Chen et al. [8] propose an algebraic approach to efficiently monitor the end-to-end loss rates in an overlay network. They use linear algebraic techniques to find a minimal *basis set* of overlay links to monitor and then infer the loss rates of the remaining ones. iPlane [4] predicts end-to-end path performance from the measured performance of segments that compose the path. We generalize these techniques and allow measuring both end-to-end paths and underlying segments. Our approach in this paper requires a deep collaboration between the overlay network operator and the native network, similar to the design goals of the overlay-friendly native network[7].

The remainder of this paper is organized as follows: We describe the multi-layer monitoring problem in Section 2. Section 3 presents our linear program based solution. We present details from simulating the multi-layer monitoring framework in general topologies in Section 4. Section 5 describes PlanetLab experiments that we conducted to characterize the inference errors that can result from this multi-layer monitoring solution. We conclude the paper in Section 6.

2 The Multi-layer Monitoring Problem

We model the native network as a directed graph $G = (V, E)$, where V is the set of vertices and E is the set of directed edges connecting these vertices. Next, we model the overlay network as a directed graph $G' = (V', E')$, with $V' \subseteq V$ being the set of overlay nodes and E' being the set of overlay links. In a multi-layer network, each overlay link spans one or more native links. Thus, the following relation holds: $e' \in E'$ is a set $\{e_{1e'}, e_{2e'}, ..., e_{ne'}\}$, where $e_i \in E$ and $e_{ke'}$ denotes the k^{th} native edge in e'.

Link monitoring incurs a certain cost, typically in the form of resource overhead (e.g., processor utilization, bandwidth), at each layer. We use $C(e)$ and $C'(e')$ as the cost of monitoring a native link and an overlay link respectively. Since $C(e)$ and $C'(e')$ are variables, the cost structure is flexible and can accommodate various scenarios. For instance, if it is not possible to monitor certain native links directly, the cost variables for those links can be set to infinity.

Let $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_\mathrm{N}\}$ represent the desired set of monitoring operations we would like to get results for, which in our case is the set of desired overlay link measurements. Let $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_\mathrm{R}\}$ represent the set of monitoring operations that are actually performed. This set can contain a mixture of native and overlay link measurements. Let *composition rule* $\mathcal{F}(\mathcal{P}, \mathcal{M}_\mathrm{i})$ represent a function that combines the results from available native and overlay link measurements to infer the desired measurement of the overlay link $\mathcal{M}_\mathrm{i}$. In this work, we use the composition rule of the latency metric.

We say that a certain $\mathcal{M}$ is *feasible with respect to* $\mathcal{P}$, if all values in $\mathcal{M}$ can be computed from $\mathcal{P}$. Clearly, if $\mathcal{M} \subseteq \mathcal{P}$, then the monitoring problem is *feasible.* In cases when $\mathcal{M} \nsubseteq \mathcal{P}$, feasibility is not always assured.

The optimization problem can thus be stated as, "Given a monitoring objective $\mathcal{M}$, find the $\mathcal{P}$ such that $\mathcal{M}$ is feasible with respect to $\mathcal{P}$ and $cost(\mathcal{P}) = \sum_{i=1}^{R} cost(\mathcal{P}_i)$ is minimal."

Assumptions and Limitations. In this paper, we assume that the best-effort routing at the native layer treats measurement probes in the same manner as other data packets, so as to obtain an accurate estimate of the user experience. We restrict our work to the metric of latency, although it has been shown that the logarithm of link loss rates are additive metrics that can be composed in a manner similar to link latencies[8]. Furthermore, the linear programming formulation in

Table 1. Notations used

E	Edges in the native layer
E'	Edges in the overlay layer
$C(e)$	Cost to monitor native link e
$C'(e')$	Cost to monitor overlay link e'
$X_m(e)$	1 if native link e is monitored, 0 otherwise*
$X_i(e)$	1 if native link e is inferred, 0 otherwise*
$Y_m(e')$	1 if overlay link e' is monitored, 0 otherwise**
$Y_i(e')$	1 if overlay link e' is inferred, 0 otherwise**
$f(e, e')$	1 if overlay link e' is routed over native link e, 0 otherwise
$x_i(e, e')$	1 if native link e is inferred from overlay link e', 0 otherwise
$l_i(e)$	Integer representing the inference dependency between native links to resolve inference loops

* A native link can be monitored or inferred but never both. Some are neither monitored nor inferred if they are not needed in inferring overlay link measurements.

** An overlay link is either monitored or inferred, but never both.

the subsequent section cannot be applied for multi-path routing at the native layer: The overlay link latency composition rule needs revision for handling multi-path routing. We reserve these extensions to the model for future study.

3 Linear Programming Formulation

Using the notation presented in Table 1, we formulate the optimization problem as the following Integer Linear Program (ILP):

$$\textbf{minimize}\ \text{Total Cost} = \sum_{e \in E} X_m(e) \cdot C(e) + \sum_{e' \in E'} Y_m(e') \cdot C'(e') \tag{1}$$

subject to the following constraints

$$\forall\ e' \in E', e \in e' : \quad X_m(e) + X_i(e) = 1,\ \text{if}\ (Y_m(e') + Y_i(e')) = 0\ . \tag{2}$$

$$\forall\ e' {\in} E', e \in e', d \in (e' - e) : \quad x_i(e, e') \leq (X_m(d) + X_i(d))\ . \tag{3}$$

$$\forall\ e' \in E' : \quad \sum_{e \in e'} x_i(e, e') \leq (Y_m(e') + Y_i(e'))\ . \tag{4}$$

$$\forall\ e \in E : \quad X_i(e) \leq \sum_{e' \in E'} x_i(e, e') \leq 1\ . \tag{5}$$

$$\forall\ e' \in E', e \in e', d \in (e' - e) : \quad x_i(e, e') = \begin{cases} 1, & \text{if } l_i(e) > l_i(d), \\ 0, & \text{otherwise}\ . \end{cases} \tag{6}$$

$$\forall\ e' \in E' : \quad Y_i(e') = 1, \text{if } e' \text{ can be inferred from other overlay links in } \mathcal{P}\ . \tag{7}$$

$$\begin{aligned} \forall\ e \in E, e' \in E' : \quad & X_m(e) \in \{0,1\}, X_i(e) \in \{0,1\}, x_i(e, e') \in \{0,1\}, \\ & Y_m(e) \in \{0,1\}, Y_i(e) \in \{0,1\}\ . \end{aligned} \tag{8}$$

Constraints (2) to (8) assure the feasibility of the solution. These constraints can be explained as follows:

(2) This constraint, applied to all overlay links, determines the exact layer at which each overlay link is to be monitored. If the overlay link is not already monitored or inferred, then monitor, or infer, all native links it spans. Furthermore, this constraint will ensure that we only monitor or infer, and never both. This condition also prevents an overlay link from being monitored, if all its constituent native link measurements are already known.

(3) We enforce the constraint that a native link e is inferred from an overlay link e' only if all other native links in that overlay link are already monitored or inferred. This insures that the inferred native link can be appropriately calculated from other link measurements.

(4) This constraint insures that a native link e is inferred from an overlay link e' only if the overlay link latency is already monitored, or inferred, at the overlay layer (i.e., $Y_m(e') + Y_i(e') = 1$) . Furthermore, we place the constraint that no more than 1 native link can be inferred from each overlay link. This is typically achieved in an ILP by setting the sum of individual variables $x_i(e, e')$ to be less than or equal to 1.

(5) This is a complex constraint which achieves three sub-goals: (a) Mark a native link as *inferred* if it is inferred on any of the overlay links that span it, (b) Mark a native link as *not inferred* if it is not inferred on any of the overlay links that span it, and (c) Insure that a native link is inferred only from 1 overlay link, so as to reduce wasting resources on performing multiple inferences. These three constraints ensure that we accurately mark a native link as *inferred.*

(6) This constraint is crucial to remove any *circular inference*, which can happen if we infer one native link measurement through an arithmetic operation on the measurement of another. We achieve this by assigning integer inference levels (denoted by variable l_i), such that a native link must be inferred only from other native links that have a lower inference level.

(7) We use this constraint to implement the basis set computation and infer some overlay link measurements from other known overlay link measurements.

(8) Lastly, we specify the binary constraints for all variables used. This constraint makes the problem hard.

We apply the above ILP to any given topology and solve it using the GNU linear programming kit[3], which uses the branch-and-bound approximation technique. The optimal solution for a given topology identifies the overlay links that can be inferred from other native and overlay links, and describes how these inferences should be done. Using this information, we infer the latency of all overlay links ($\mathcal{M}$) from available measurements ($\mathcal{P}$) in our database.

4 Examples Using Multi-layer Monitoring

In this section, we present various simulation experiments to demonstrate the types of results obtainable from our optimization approach and how it is affected by various network features. Although we only simulate intra-domain topologies, our model and ILP are equally applicable to multi-domain topologies.

Random Placement. In the first experiment we consider five native link topologies derived from Rocketfuel [6] data. For each network we generate an overlay network using approximately 20% the number of nodes in the native topology as overlay nodes. These nodes are placed randomly among the native nodes and fully-connected to form the overlay network. In this case, we define the cost of monitoring as the total number of native and overlay measurements needed. We consider the following four monitoring strategies:

- *Monitoring all overlay links:* The total cost is the cost of monitoring all $N.(N-1)$ overlay links, where N is the number of overlay nodes.

- *Monitoring all native links:* The total cost is the number of distinct native links spanned by all the overlay links.
- *Monitoring a basis set of overlay links:* To obtain this solution, we set the cost of monitoring a native link very high in our ILP so that the solution selects only overlay links for monitoring.
- *Monitoring a combination of native and overlay links:* We set the cost of monitoring a native link equal to the cost of monitoring an overlay link in the ILP. (From here on, we refer to these costs as $unitNativeCost$ and $unitOverlayCost$, respectively.) The ILP then produces a solution that minimizes the total cost, which is the same as minimizing the number of measurements in this case.

Table 2 demonstrates the lowest total monitoring cost that can be achieved by the above monitoring strategies for each topology. In addition, the cost that results from monitoring native links and the cost that results from monitoring overlay links are reported separately for the multi-layer combination strategy in the last column. In all topologies, monitoring a combination of native and overlay links provides the lowest-cost option. On average, this lowest cost is 71% lower than the cost for the naive all-overlay approach and 11% lower than the all-native solution. This represents significant saving, while being flexible enough to accommodate other constraints.

Table 2. The lowest cost for each strategy when $unitNativeCost = unitOverlayCost$

AS #	Number of overlay nodes	All overlay	All native	Basis set	Combination (n: native, o: overlay)
1221	21	420	102	198	98 (66 n, 32 o)
1755	17	272	112	98	92 (42 n, 50 o)
3257	32	992	240	500	222 (142 n, 80 o)
3967	15	210	98	138	78 (46 n, 32 o)
6461	28	756	224	394	210 (146 n, 64 o)

Amount of link-level overlap. In this section, we study the effect of overlap between overlay links over the optimal monitoring solution. As a measure, we use the average number of overlay links that span a native link in the network. We call this value the *overlap coefficient*. For this analysis we use the results from the first experiment.

Table 3 demonstrates how the lowest cost solution, as given by our ILP, varies with the amount of link-level overlap. In the table, *Cost per overlay link* represents the total monitoring cost divided by the number of overlay links. The rows are sorted by increasing overlap coefficient. We observe that in general, the monitoring cost per overlay link decreases as overlap increases. However, the cost per link value for AS 1221 is slightly higher than that of AS 3257 although the former has a higher overlap coefficient. This may suggest that increasing overlap can only decrease the cost per link by a limited amount.

Table 3. Effect of link-level overlap on the lowest total monitoring cost

AS	Overlap coefficient	Lowest total cost	# of overlay links	Cost per link
3967	8.59	78	210	0.37
1755	9.21	92	272	0.34
6461	12.80	210	756	0.28
3257	17.08	222	992	0.22
1221	18.33	98	420	0.23

Percentage of overlay nodes. In this experiment, we vary the fraction of overlay nodes among all nodes in the network. We call this fraction *overlay node density*. We examine two Rocketfuel topologies using five different density values from 0.1 to 0.5, and random overlay node placement. Our ILP gives the results in Table 4 when $unitNativeCost = unitOverlayCost$. This result is consistent with the effect of link-level overlap. As the overlay node density increases, link-level overlap also increases, and the cost per overlay link decreases.

Table 4. Effect of overlay node density on the optimal monitoring solution

Overlay node density	Cost per link for AS 1755	Cost per link for AS 3967
0.1	0.75	0.71
0.2	0.34	0.37
0.3	0.25	0.28
0.4	0.15	0.17
0.5	0.12	0.13

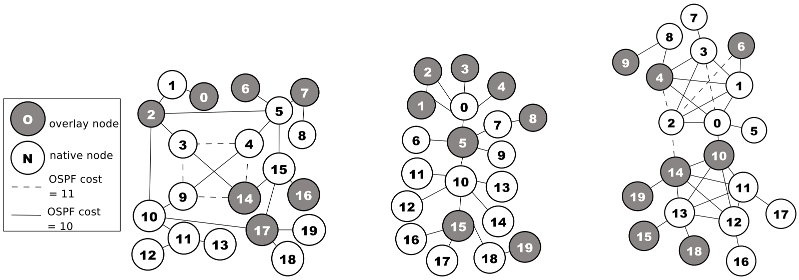
5 Experimental Evaluation of Inference Errors

Composing an end-to-end measurement from other measurements can introduce an error in the result. We refer to this as *inference error*. One source of error may be packets traversing different sequences of router functions. For example, an end-to-end latency measurement probe may be forwarded along the fast path of a router, while probes that measure the latency of native links may be forwarded along the slow path. This makes the latter probe packets susceptible to processor delays, thereby introducing additional latency. Furthermore, some native link measurements may be inferred from overlay link measurements using arithmetic operations. This too introduces estimation error.

We represent the inference error for overlay links by computing the *absolute relative estimation error*. We compute this error value as a percentage:

$$\text{Abs. Rel. Est. Error Percentage}(e') = \frac{|\widehat{\rho}(e') - \rho(e')|}{\rho(e')} \times 100 \tag{9}$$

where $\rho(e')$ is the actual measurement result for e' and $\widehat{\rho}(e')$ is the inferred result obtained through combining a different set of measurements.



(a) Topology 1 (b) Topology 2 (c) Topology 3

Fig. 1. Three PlanetLab topologies we use. (a) represents a general AS topology. (b) has a tree-like structure which can be found on some campus-wide networks such as [2]. (b) can be interpreted as a graph of two interconnected ASes. Native links are assigned with different OSPF costs to avoid multiple shortest paths.

To assess the extent of inference errors, we conducted experiments on PlanetLab [5] using three different overlay topologies shown in Fig. 1. We implemented these topologies as virtual networks on PlanetLab using PL-VINI, the VINI [1] prototype running on PlanetLab. In each experiment, we picked 20 PlanetLab nodes from different ASes as our native network and ran OSPF on this network with PL-VINI. Note that we cannot control the inter AS routing of these PlanetLab nodes. We treated the edges between these nodes on the PL-VINI network as native links. We picked 8 nodes out of the 20 as our overlay nodes, and assumed that these 8 nodes are fully connected to form an overlay network.

For each topology, we ran 4 rounds of measurements at different times. In each round, we measured the delay on all native and all overlay links by simultaneously running 100 pings on every link at a frequency of 1 per second. We calculated the delay from node a to node b as the average round-trip time over all ping results for native or overlay link $a - b$.

In order to find the optimal combination of links to monitor for these topologies, we ran our ILP on each of them with the objective of minimizing the total number of measurements. The output of the ILP gave us a set of overlay and native links to monitor. Using this output and the measurement results for the corresponding topology, we first inferred the measurements of the links that are not monitored, and then calculated the errors in these inferences using Eq. 9. The errors for all-native and basis set solutions are calculated in a similar manner.

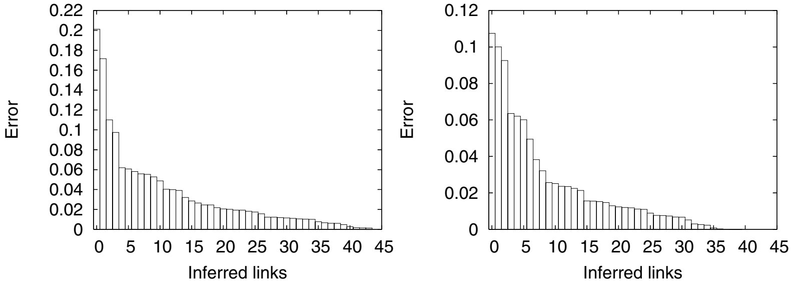
Table 5 summarizes the results for all three topologies. The *Cost* column represents the lowest possible monitoring cost that can be achieved by each strategy. *Max* is the largest inference error observed in a certain strategy. Mn_i is the inference error averaged over all *inferred* overlay links, while Mn_a is the error averaged over *all* the overlay links in the network, with the difference being that direct overlay link measurements have no errors. Averaging over all overlay links does not reduce the error in the case of all-native monitoring because in this case all overlay links are inferred and none are measured directly. However,

Table 5. Costs and inference errors for different monitoring strategies

	Topology 1				Topology 2				Topology 3			
	$Cost$	Mn_i	Mn_a	Max	$Cost$	Mn_i	Mn_a	Max	$Cost$	Mn_i	Mn_a	Max
All-overlay	56	0	0	0	56	0	0	0	56	0	0	0
All-native	34	5.01	5.01	21.18	24	1.43	1.43	4.30	30	3.54	3.54	10.75
Basis set	38	2.68	0.86	20.29	26	0.96	0.51	2.79	26	1.13	0.61	4.95
Combination	26	3.43	2.70	20.12	18	1.58	1.35	3.17	24	2.35	1.68	10.75

$Mn_a < Mn_i$ in the basis set and lowest-cost combination strategies because some overlay links are directly measured and these zero errors bring down Mn_a.

Among the last three strategies, monitoring a combination of native and overlay links achieved the lowest cost, and monitoring a basis set of overlay links resulted in the smallest error. However, we should note that if we use a different cost definition, such as the total number of native links carrying probe traffic, these results may change significantly. For instance in topology 3, the last strategy uses a combination of 8 native and 16 overlay links, spanning a total of 42 native links , while the all-native solution spans 30 links and the basis set solution spans 52 native links. Our insight from these experiments suggests that in general, all-native solutions minimize bandwidth consumption, basis overlay set solutions minimize error, and using a combination of native and overlay links allows reducing the total number of measurements with comparable errors.



(a) Topology 1

(b) Topology 3

Fig. 2. Error rates of inferred overlay links

For the two topologies whose maximum errors are above 10%, we examine the error distribution among the inferred overlay links as shown in Fig. 2. We sort the inference errors from high to low and place them on the graphs from left to right. It can be seen that in both cases a few inferred links produce high errors that dominate the rest, increasing the mean error. If the ILP is aware of the overlay links that incur a high error when they are inferred, it can choose to monitor them directly and avoid these errors. Thus, adding certain error constraints to the ILP is a plausible step to improve its performance.

6 Conclusions

In this work we have proposed *multi-layer monitoring* as a flexible approach for overlay network measurement. We focused on the specific issue of determining the optimal mix of native and overlay link monitoring. We show that the overall cost of monitoring the network is the least when we allow native link measurements, as well as end-to-end measurements. We present a novel ILP formulation that when solved minimizes the cost of network monitoring with the appropriate combination of end-to-end and native network measurements. Through simulation studies, we observe that the optimal monitoring solution, i.e. the set of native and overlay links that minimizes the total monitoring cost while supplying sufficient information, depends on unit monitoring costs as well as the selection and placement of overlay nodes. We also find that the average monitoring cost per overlay link is lower for topologies where there is a high overlap between overlay links. Furthermore, we evaluate our approach through PlanetLab experiments with a focus on the question of *inference errors*.

Future work in this area should include: 1) applying our approach to multi-domain scenarios, 2) consideration of monitoring for metrics other than latency, 3) including error minimization as an objective in the optimization problem, 4) extending multi-layer monitoring to include Layer 2, 5) considering problems of dynamic monitoring which would allow changes in the monitoring mix over time in response to changing network conditions or changes in overlay topology.

References

1. Bavier, A., et al.: In VINI veritas: realistic and controlled network experimentation. In: Proceedings of ACM SIGCOMM, pp. 3–14 (2006)
2. CPR: Campus Wide Network Performance Monitoring and Recovery, `http://www.rnoc.gatech.edu/cpr`
3. GNU Linear Programming Kit (GLPK), `http://www.gnu.org/software/glpk`
4. Madhyastha, H.V., et al.: iPlane: An Information Plane for Distributed Services. In: OSDI, pp. 367–380 (2006)
5. Planetlab, `http://www.planet-lab.org`
6. Rocketfuel: An ISP Topology Mapping Engine, `http://www.cs.washington.edu/research/networking/rocketfuel/`
7. Seetharaman, S., Ammar, M.: Overlay-friendly Native Network: A Contradiction in Terms?. In: Proceedings of ACM HotNets-IV (November 2005)
8. Chen, Y., et al.: Algebra-based scalable overlay network monitoring: algorithms, evaluation, and applications. IEEE/ACM Trans. Netw. 15(5), 1084–1097 (2007)