NORTHWESTERN
UNIVERSITY

Electrical Engineering and Computer Science Department

Technical Report
NWU-EECS-08-06
August 15, 2008

Towards Efficient Large-Scale VPN Monitoring and Diagnosis under Operational Constraints

Yao Zhao, Zhaosheng Zhu, Yan Chen, Dan Pei and Jia Wang

Abstract

Continuous monitoring and diagnosis of network performance are of crucial importance for the Internet access service and virtual private network (VPN) service providers. Various operational constraints, which are crucial to the practice, are largely ignored in previous monitoring system designs, or are simply replaced with load balancing problems which do not work for real heterogeneous networks.

Given these real-world challenges, in this paper, we design a VScope monitoring system with the following contributions. First, we design a greedy-assisted linear programming algorithm to select as few monitors as possible that can monitor the whole network under the operational constraints. Secondly, VScope takes a multi-round measurement approach which gives a smooth tradeoff between measurement frequency and monitors deployment/management cost. We propose three algorithms to schedule the path measurements in different rounds obeying the operational constraints. Finally, we design a continuous monitoring and diagnosis mechanism which selects the minimal extra paths to measure to identify the faulty links after the discovery of faulty paths. Evaluations based on several real VPN topologies from a tier-1 ISP as well as some other synthetic topologies demonstrate that VScope is promising to solve the aforementioned challenges.

Keywords: VPN, monitoring and diagnosis, operation constraints

Towards Efficient Large-Scale VPN Monitoring and Diagnosis under Operational Constraints

Abstract— **Continuous monitoring and diagnosis of network performance are of crucial importance for the Internet access service and virtual private network (VPN) service providers. Various operational constraints, which are crucial to the practice, are largely ignored in previous monitoring system designs, or are simply replaced with load balancing problems which do not work for real heterogeneous networks.**

Given these real-world challenges, in this paper, we design a $VScope$ monitoring system with the following contributions. First, we design a greedy-assisted linear programming algorithm to select as few monitors as possible that can monitor the whole network under the operational constraints. Secondly, VScope takes a *multi-round* measurement approach which gives a smooth tradeoff between measurement frequency and monitors deployment/management cost. We propose three algorithms to schedule the path measurements in different rounds obeying the operational constraints. Finally, we design a continuous monitoring and diagnosis mechanism which selects the minimal extra paths to measure to identify the faulty links after the discovery of faulty paths. Evaluations based on several real VPN topologies from a tier-1 ISP as well as some other synthetic topologies demonstrate that VScope is promising to solve the aforementioned challenges.

1. Introduction

Recently the Internet has witnessed an unprecedented growth in terms of the scale of its infrastructure, the traffic load, as well as the abundant applications. More importantly, there is an exponential growth for MPLS-based IP Virtual Private Networks (VPN) recently. Large enterprise networks often have multiple sites that are at seperate geographical locations. For example, large corporations such as IBM and Nokia have offices/branches that locate in many countries. Another example is large retail stores such as Macys and Wal-Mart, which have thousands of stores globally. To connect sites (e.g., an office or a store location) within an enterprise network, one approach would be to deploy/lease physical lines between sites. Alternatively, connectivity between sites can be provided and managed by ISPs via MPLS/VPN. This approach has been adopted widely because of its low cost and great flexibility.

Because a VPN provider is often the sole provider of connectivity among a customer's sites, continuous monitoring and diagnosis of VPN performance are of crucial importance for the VPN service providers to ensure the reliability and quality of service, especially given that VPNs often carry important business applications, such as VoIP, realtime streaming video, and financial transactions that do not react well to even small traffic disruptions. In addition, from an ISP's perspective, continuous monitoring of network performance not only helps reporting and diagnosing possible service level agreements (SLAs) violations, but also provides useful input to many important network operations such as traffic engineering and network provisioning.

Today, ISPs heavily rely on the standard passive monitoring approach via SNMP, which usually polls the status of each router/switch periodically. However, there are several issues. First, an ISP usually provide VPN services to a large number of customers such as enterprise networks, all of which run on top of the same ISP infrastructure. As such, the ISP needs to monitor hundreds of thousands of routers. Therefore, it is infeasible to frequently poll every router due to the large bandwidth and management overhead. Secondly, SNMP based monitoring is unable to measure the path-level features such as latency.

Therefore, active measurements are important complement to the SNMP based monitoring approach and are also used by ISPs widely. However, most existing network monitoring and diagnosis designs [1]–[7] miss an important piece: various constraints that should be imposed on the monitors and links so that the measurement does not interfere with the normal operation or traffic, and meets the business requirement (in the case of VPN). For example, the capacity of access links that connects each site belong to a single VPN can be very limited. Note that the link capacity often is not the physical link capacity, but the maximal bandwidth allowed. For example, we find that vast majority of access links in thousands of VPNs managed by a tier-

1 ISP only have 1.54Mbps capacity. This is because customers often do not have incentive to pay for their providers to over provision the access link capacity. We define the *operational constraints* to be the set of constraints or rules that the monitoring system should comply to. For example, a typical constraint can be that all the measurement overhead over a link cannot exceed 1% of the link capacity.

In this paper, our goal is to design a monitoring and diagnosis system for the VPN infrastructure that ISPs deploy to host VPN services. Usually the backbone links are over-provisioned (*e.g.*, 10Gbps) while access links may have several orders of magnitude difference in terms of capacity (*e.g.*, 1.54Mbps mentioned above). Such high heterogeneity of link capacities and router capabilities makes it very likely to severely overload some monitors and links when selecting the monitors or paths for monitoring (as in [1, 4, 5, 7]) without considering these constraints. Taking the operational constraints into account makes this problem very challenging and unique from the existing work for the following reasons.

- Considering the operational constraints makes some well-known problems much harder. As we show in Section 3.2, minimizing the number of monitors under the constraints becomes harder than some notorious NP-hard problems.
- Most tomography work assumes that all the paths to be monitored will be measured simultaneously [1, 4, 5, 7]. However, this setup may not be true or efficient under the real-world constraints. Under tight constraints, the monitoring system may have to schedule the measurement in different time slots. We found this particularly crucial for the VPN topologies which exhibit star-like topologies where a backbone router connects to a large number of customer routers.

To address these challenges, in this paper, we propose $VScope$, a *continuous* monitoring and diagnosis system for VPN. While we mainly focus on VPN service in this paper, our system is general enough to work on any other network that its resources are limited and the operational constraints should be considered in its active monitoring system (*e.g.* IP network of a small Tier-3 ISP). VScope consists of two phases: 1) monitor setup phase which selects the candidate routers as monitors and schedules the paths to be measured by the monitors *in multiple rounds*, and 2) continuous monitoring and diagnosis phase. We make the following contributions in designing the VScope.

First, we design algorithms to select as few monitors as possible that can monitor the whole network under the operational constraints. The special case of our problem ignoring the operational constraints is shown to be NP-hard in [1]. Considering the operational constraints, we model our problem as a unique combination of the two-level nested Set Cover problem and constraint satisfaction problem. We found that no existing solutions such as those for variants of Set Cover problem [8] can be directly applied to solve this new problem. Thus we design a greedy-assisted linear programming algorithm for it. In addition, we develop a simple but scalable greedy algorithm for a smooth efficiency-optimality tradeoff.

Secondly, VScope takes a multi-round measurement approach which gives a smooth tradeoff between measurement frequency and monitors deployment/management cost. With the single-round measurement algorithms as the basis, we propose three algorithms to schedule the path measurements in different rounds obeying the operational constraints.

Finally, VScope not only detects the *existence* of some fault (*e.g.* large loss rates or latency) but also quickly identify exactly which links are faulty so that operators can take actions for mitigation. We design a continuous monitoring and diagnosis mechanism which aims to select the minimal number of paths to measure for diagnosis under operational constraints.

Besides some synthetic topologies, we mainly evaluate the VScope system with one IP network topology and two VPN topologies, *all with the real topologies, capacities, loss rates and constraints*, from a tier-1 ISP. The sizes of networks vary from hundreds to hundreds of thousands of routers. The results demonstrate that we are able to select about 5% of routers as monitors to cover all the links with all the operational constraints. When faulty [1] paths are discovered, in about 17 seconds, we identify only a small number of extra paths to be measured for locating the faulty links with small granularity close to that of the physical link. Both false positives and false negatives for locating the faulty links are less than 0.5%.

The rest of the paper is organized as follows. We introduce the problem and VScope architecture in Section 2. We present our design on monitor selection in Section 3 and our design on diagnosis in Section 4. The dynamics issues are discussed in Section 5. Then we show the evaluation methodology and results in Section 6. Finally, we present related work in Section 7 and conclude in

[1] Faulty means lossy in this paper as we currently focus on loss rates.

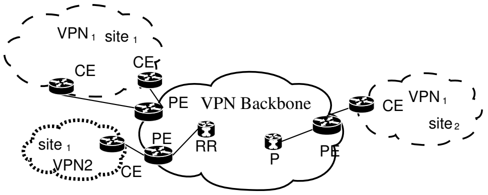

Fig. 1. Example of Layer-3 IP VPN infrastructure.

Section 8.

2. Problem Definition and VScope Architecture

1. Problem Definition

From the ISP operational perspective, the goals of network monitoring are two-fold. First, ISPs need to actively measure or infer the performance of all the possible paths through the VPN. Second, ISPs also need to quickly identify the root cause of the performance degradation or service disruption. The monitoring problem can be divided into two phases: setup phase for monitor selection, and continuous monitoring and fault diagnosis phase. In this section, we define each of the subproblems in the two phases. Note that the two phases are coupled tightly, because the goal of monitor selection is to optimize the second monitoring and diagnosis phase.

1) Background on ISP VPN Infrastructure: A layer-3 Virtual Private Network (VPN) refers to a set of sites among which communication takes place over a shared network infrastructure called a *VPN backbone*. Figure 1 shows a VPN backbone with two VPNs and three sites. *Customer Edge device routers (CE routers)* are connected to routers in the *Provider Edge device routers (PE routers)* via external BGP (eBGP). Other routers in the provider network are called *Provider's device routers (P routers)*. Each PE router maintains a Virtual Routing and Forwarding (VRF) table for each VPN so that routes from different VPN customers remain distinct and separate even if multiple VPN customers use the same IP address space. Internal BGP (iBGP) is used to distribute the VPN routes within the VPN backbone. Within the VPN backbone, *Multi-Protocol Label Switching (MPLS)* tunnels between PEs are used to forward packets. It is worth mention that the goal of the VScope system is to monitor and diagnosis the whole ISP VPN infrastructure including the shared VPN backbone and the customer routers, instead of a single VPN.

2) Measurement Constraints: One guideline of active measurements is to avoid interrupting the normal network traffic or overloading network or computation resources. After consulting network operators of a major tier-1 ISP, we consider the following realistic measurement constraints:

- **Monitor constraints.** Not all the routers can be selected as monitors for various business and hardware reasons. For example, some CE routers are not managed by the VPN provider. We define the routers that can be monitors as *candidate routers*. Each candidate monitor has limited probing ability (*e.g.*, 50 probes/second). Given a fixed measurement overhead on each measured path, a monitor thus can measure only a limited number of paths simultaneously. This constraint is called *monitor constraint* or *node constraint*.
- **Replier constraints.** The routers that can reply to the probes from the monitors are *repliers*. To avoid overloading the replier routers, we enforce the replier constraint, which specifies the number of probes that the replier can reply in a certain period. Note the operators may need to adjust the access list and rate limit of the router configuration to comply with the replier constraint without introducing security holes. For example, we can configure a router to allow 100 ICMP Echo Reply or ICMP Timestamp Reply (usually for latency measurement) per second from the senders in some IP prefix.
- **Link constraints.** Every link has its own bandwidth. The measurement overhead on a link should not exceed a certain portion of the link bandwidth (*e.g.*, 1%). We call such constraint *link bandwidth constraint* or *link constraint* in short. Generally, the link capacity in the backbone networks is pretty large, while the access links usually have much lower capacity. For example, we found that, among thousands enterprise VPN configurations that we have examined, more than 70% access links have capacity of only 1.54 Mbps, while the backbone links usually have capacity of 150 Mbps or more. Considering the number of access links are much more than that of backbone links, we can see in deed most of the links have low bandwidth.
- **Measurement path selection constraints.** VPN provides the traffic isolation between different customers. So only the sites/routers within the same VPN can communicate with each other. The path selected for measurement in VScope needs to satisfy this constraint too. Note the measured paths are round-trip paths because the non-monitor routers can only reply to probes.

3) Monitor Setup Phase: As introduced in Section 1, for existing Internet tomography works [4, 5, 7, 9], the experiment design problem is mainly to select a path set that satisfies some optimization goal to measure. For example, in [9], a minimal set of paths that covers all the links is the selection goal; while in [4] the goal path set corresponds to the basis of the path matrix in linear algebra. However, our VScope system design is unique due to the four challenges introduced in Section 1.

Note that the monitor setup problem comprises of the path selection problem because the ultimate goal is to monitor the networks by measuring some paths via the monitors. We realize that the operational constraints already result in a very challenging monitor (as well as path) selection problem in this paper, and hence we consider the simplest path selection goal (*i.e.*, covering all links), while leaving more sophisticated path selection goals to our future work.

Besides, it is worth mentioning that in previous works [4, 5, 7], all the selected paths are measured simultaneously. However, given the operational constraints and the special large star-like topology of the networks, we find that scheduling path measurements in multiple rounds is an efficient approach to save the monitor installation cost. Therefore, in a *measurement phase* not all the paths are measured at the same time and multi-round scheduling of the path measurement is adopted in our system (presented in Section 3.3).

Mathematically, the monitor selection problem can be abstracted and generalized as follows: Let $G(V, E, P)$ be a network where V is the vertex set, E is the edge set and P is the predefined set of paths. Assume Φ is a set of rules that determines if the selection of paths $P' \subset P$ is allowed or not. The problem is to select a path set P^* satisfying Φ and for each edge $e \in E$ there exists a path $p \in P^*$ with $e \in p$. Meanwhile, let V^* be the set of starting vertices of all paths in P^*, and the goal is to minimize the size of V^*.

4) Monitoring and Fault Diagnosis Phase: VScope monitoring involves periodically probing or inferring the path performance metrics, such as reachability, latency, loss rate, and so on. In this paper, we focus on the loss rate monitoring and diagnosis.

When the monitoring system detects a path that fails to meet the SLA with customers, it is always desirable to locate the faulty link which caused the violation. However, locating faulty links from path measurements is a hard problem. As shown in Section 7, a lot of algorithms [2, 6, 7] have already been designed for this purpose. In our paper we do not focus on the faulty link location algorithm, but leverage on and extend the existing approaches. More importantly, given the fact that link performance metrics usually have constancy [10], we consider the following problem in our VScope system: *when faulty paths are discovered in the path monitoring phase, how to quickly select some paths under the operational constraints to be further measured so that the faulty link(s) can be accurately identified?*

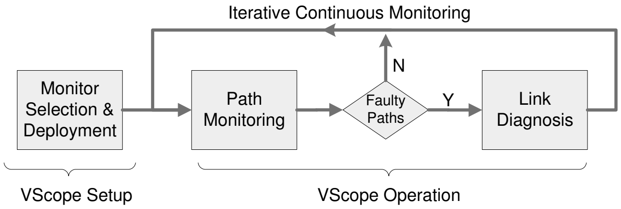

Fig. 2. VScope System Architecture.

2. Architecture

Figure 2 shows the architecture of our system. The architecture has two components: *monitor selection*, and *continuous monitoring and diagnosis*. First, a set of monitors are selected according to the algorithms introduced in Section 3, and measurement boxes/software are installed. Then the monitors probe paths and diagnose faulty links periodically. In each round, a set of paths is measured using active probing. Next, if some paths are found to be faulty, the diagnosis component will further locate the faulty links along the faulty paths. Additional path measurements are selected and conducted for this purpose. VScope has a centralized coordinator (like the network operation centers for many major ISPs) which assigns measurement tasks to monitors, collects the measurement results, and detects faulty paths and identifies faulty links.

In our current design, the diagnosis phase is also compliant with the operational constraints and takes an exclusive round. Alternatively, the diagnosis phase can be parallel with the next round of path monitoring if some extra budget is allowed for the diagnosis phase (which may be rare) by network operators. Both the options are supported in our system framework. Network operators can choose either one based on their preference.

3. VScope Setup: Monitor Selection

Monitor selection is the first component of our VScope system. As described in Section 2.1.3, the goal is to minimize the number of monitors selected to actively monitor links in the network satisfying some operation constraints.

The constraint satisfactory problems including our problem (See Section 3.2 for the hardness of our problem) is usually NP-hard, and even the best algorithms may not be able to achieved the satisfaction [11]. In our VScope system, we do not plan to struggle in the notorious satisfaction problem. Instead, we propose to schedule the path measurements into different rounds [2] to "reduce" the harsh constraints so that simple algorithms like the greedy algorithm can at least find a solution easily. Meanwhile we find multi-round can significantly cut down the number of monitors required to monitor the networks. In reality, as we studied in Section 6.2, typical ISP VPN have star-like topologies. Without multi-round measurement, the best algorithm still has to select tens of thousands of monitors [3]. Hence the emerging topologies further stress the necessity of the multi-rounding idea.

1. Overview of Multi-round Monitoring

The main idea of our multi-round monitoring is as follows: we consider R rounds of back-to-back measurements and in each measurement round different paths are measured by the selected monitors. Finally, all the links are covered by at least one of the R rounds of measurements. The multi-round monitor selection algorithm tries to minimize the number of monitors that can cover all the links in a certain number of rounds (R).

An optimal solution should consider both the monitor/path selection and the schedule of the path measurements in multiple rounds at the same time, which is very hard involving the both monitor/path selection and scheduling problems. Therefore, we propose a two-step solution for the multi-round monitor selection problem. First we convert the multi-round selection problem to the "single-round" selection problem while relaxing the monitor's constraints and link bandwidth constraints by a factor of the round number R. In this step, we obtain the selected monitors as well as paths to be measured. In the second step, we schedule the paths to be measured in the R rounds appropriately, trying to satisfying the constraints of each round.

2. Monitor Selection

The monitor selection problem seems to be similar to the problem in [1], which is a simpler case of our problem without considering the operation constraints. And in [1] Bejerano *et al.* proved that this simplified case of our problem is NP-hard. The monitor selection problem resembles the well-known Minimum Set Cover problem [8, p. 118]. One can imagine each link as an element and each candidate router as corresponding to a set. We say a path *covers* a link if the link is on the path, and a link is *associated with* a router if the link is covered in at least one of the paths starting from the router. Hence a router's corresponding *set* contains all the links associated with the router. The Minimal Set Cover problem involves finding the smallest number of sets (or routers) that cover all the elements (or links). However, the existence of monitor/replier/link constraints makes our problem first a constraint satisfactory problem. Our problem is a generalization of the Exact Cover problem [8], a variant of the Set Cover problem. The Exact Cover problem requires the elements to be covered exactly *once* in all the selected sets.

Symbols	Meaning
N	Number of routers
S	Number of links
P_{ij}	The path from router i to router j
L_k	The kth link. $L_k \in P_{ij}$ if this link on path P_{ij}
x_i	1, if node i is a monitor, otherwise 0
y_{ij}	1, if path P_{ij} is measured, otherwise 0
z_k	1, if link k is covered, otherwise 0
c_i	The number of paths that node i can measure
r_i	The number of paths that node i can reply
b_k	Max number of measured paths that can pass link k
OPT	Number of monitors required in the best solution

TABLE I Notation used in the paper

Given complicated constraints, the classic approximation algorithms for the Set Cover problem and its variants [8] can not be directly applied to solve our problem. While in principle we still use the classic algorithms of approximation algorithm (*e.g.*, greedy algorithm and linear programming), there are substantial challenges to realize the algorithms for our realistic problem. Next, we present two algorithms, the greedy algorithm and the linear programming with random rounding algorithm to solve our monitor selection problem. Table I illustrates the notations used in the paper. Note that x_i, y_{ij}, and z_k are 0-1 variables, as a router or path can be either selected or not selected and a link can be either covered or not covered.

1) Greedy Monitor Selection Algorithm: Greedy algorithms are usually one of the most straightforward and to deal with some NP-hard problems. Especially in Minimum Set Cover problem, pure greedy algorithm turns out to be a $\log M$-approximation algorithm, where M is the number of elements to cover [8]. Besides, in the average case, greedy algorithm is much more efficient than what the theoretic bound says.

[2] Paths in the same round are measured simultaneously.

[3] Although the best result is unknown, but it can be bounded by linear programming results.

```
1  Let $L = \{l_1, l_2, \ldots, l_S\}$ be the set of links;
2  Let $C = \{r_1, r_2, \ldots, r_N\}$ be the set of candidate routers;
3  Let $T = \emptyset$ be the initial set of covered links;
4  Let $R = \emptyset$ be the output of selected monitors;
5  while $L - T \neq \emptyset$ do
6      $S^* = \emptyset$ and $r = \emptyset$;
7      foreach $r_i \in C - R$ do
8          Select the path set $S_i$ which covers the maximum number of the links in $L - T$ under link constraints;
9          if $|S_i| > |S^*|$ then
10             $S^* = S_i$, $r = r_i$;
           end
11     $R = R \cup \{r\}$, $T = T \cup S^*$;
12     Update the constraints of links;
   end
```

Algorithm 1: Greedy algorithm for monitor selection.

In this section, we introduce a simple greedy algorithm inspired by the greedy algorithm for Minimum Set Cover problem. Our monitor selection problem looks like a two-level nested Minimum Set Cover problem and Maximum k-Coverage problem [12] to some extent. Algorithm 1 describes the greedy algorithm for monitor selection. The basic idea is to greedily select one router at a time, which can monitor the largest number of links that have not been covered yet.

However, the problem of evaluating the gain of adding a router as a monitor is a variant of Maximum k-Coverage problem. The Maximum k-Coverage problem is to select k sets from certain candidate sets so that the maximum elements are covered in the union of the selected sets. Maximum k-Coverage problem is an NP-hard problem and the similar greedy algorithm which is used in Minimum Set Cover problem is an $\frac{e}{e-1}$-approximation algorithm. Considering the paths as sets and links as elements, it is a k-Coverage problem to find out the number of links covered by a fixed number of paths that a router can simultaneously monitor, if we do not consider link bandwidth constraints. Similarly, our greedy algorithm also selects iteratively the path that can cover most new links while complying with the link constraints. Because of space limit, line 8 in Algorithm 1 omits the details. Unfortunately, in our problem, the greedy algorithm can no longer be claimed to be an $\frac{e}{e-1}$-approximation algorithm because the link bandwidth constraints may prevent the greedy algorithm from selecting the best path in a greedy step.

2) Linear Programming based Monitor Selection Algorithm:

1) Integer Linear Programming: We first formulate our monitor minimization problem as an integer linear programming problem (ILP) as follows (See Table I for notations):

$$P: \quad \text{Minimize} \quad \textstyle\sum_i x_i \tag{1}$$

$$s.t. \quad y_{ij} \leq x_i, \ \forall i, \ \forall j \tag{2}$$

$$\textstyle\sum_j y_{ij} \leq c_i \cdot x_i, \ \forall i \tag{3}$$

$$\textstyle\sum_j y_{ji} \leq r_i, \ \forall i \tag{4}$$

$$\textstyle\sum_{\forall i, \ \forall j, \ L_k \in P_{ij}} y_{ij} \geq 1, \ \forall k \tag{5}$$

$$\textstyle\sum_{\forall i, \ \forall j, \ L_k \in P_{ij}} y_{ij} \leq b_k, \ \forall k \tag{6}$$

Formula 1 is the minimization goal of the ILP, *i.e.*, minimizing the number of monitors needed. Inequality (2) means a path can be measured if and only if the source router of the path is selected as a monitor. The monitor and replier constraints are formulated in Inequality (3) and (4). Inequality (5) shows that a link is covered when at least one of the paths containing the link is selected. Link bandwidth constraint is enforced by Inequality (6).

2) Relaxed Linear Programming: Integer linear programming is a NP-Complete problem [13], and thus solving it may not be feasible. We use the classic relaxation techniques to relax the $\{0,1\}$-ILP to a normal linear programming problems and then apply the random rounding scheme to achieve the optimality bound in terms of statistical expectation. To relax the integer linear programming, we simply add the following constraints and remove the $\{0,1\}$-solution requirement:

$$0 \leq x_i \leq 1, \ \forall i$$
$$0 \leq y_{ij} \leq 1, \ \forall i, \ \forall j$$

After relaxation both x and y are real numbers in the range [0,1], and the linear programming problem can be solved in polynomial time. Suppose the solution is x_i^*, y_{ij}^*. We do the random rounding in the following way:

$$X_i = \begin{cases} 1 & \text{with probability} \quad x_i^* \\ 0 & \text{with probability} \quad 1 - x_i^* \end{cases} \tag{7}$$

$$Y_{ij} = \begin{cases} 1 & \text{with probability } y_{ij}^*/x_i^*, \text{ if } X_i = 1 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

If X_i is rounded to 1, the corresponding router is selected as a monitor. Once a router is selected as a monitor, the paths starting from the router have some chance to be selected to measure with the probability y^*_{ij}/x^*_i. Then the value of z_k, *i.e.* whether a link is covered or not, is decided by the rounded Y_{ij}. Let random variables $X = \sum_i X_i$ and $Z = \sum_k z_k$. We have the following theorem:

Theorem 1: After applying random rounding to the solutions of the LP problem of the monitor selection, $E(X) \leq OPT$, and $E(Y_{ij}) = y^*_{ij}$.

The proof of Theorem 1 can be simply proved using the basic probability theory and we omit the details because of space limit. Theorem 1 shows that in expectation we select no more than OPT monitors. However, after rounding not all the links are covered. Note that in the standard LP algorithm for Minimum Set Cover problem, several random rounding results are combined together to obtain the 100% coverage of all the links. In our monitor selection problem, simply combining multiple results of random rounding will violate the monitor constraints and link bandwidth limitations. Therefore, we combine the LP-based algorithm with the greedy algorithm introduced in Section 3.2.1 to achieve 100% link coverage.

We apply the following Theorem 2 [14] to show that with pretty large probability, the random rounding results are not much larger than the expected results.

Theorem 2: Let V be the sum of independent $\{0, 1\}$ random variables, and $\mu > 0$ be the expected value of V. Then for $\forall \epsilon > 0$,

$$P_r(V \geq (1+\epsilon)\mu) < e^{-\mu \min\{\epsilon,\epsilon^2\}/3}.$$

For example, let $\mu = 12$ and $\epsilon = 1$, then $P_r(V > 24) < 0.018$. According to Theorem 2, we can see that the probability of large violation of the node constraint and link constraint is small. For example, inequality 3 enforces the node constraint in the linear programming and after random rounding we have $E[\sum_j Y_{ij}] \leq \sum_j y^*_{ij} \leq c_i$. In our setup, usually one monitor can measure 12 paths simultaneously (*i.e.*, $c_i = 12$), hence we have $P_r(\sum_j Y_{ij} > 2c_i) < 0.018$. To further reduce this violation, we can run random rounding several times to find the one which has minimal violations. The result shows that there are no violations to the constraints in our experiments on real topologies (See Section 6.3).

3) Greedy-assisted Relaxed Linear Programming: We take the LP results as a good starting point, which selects a certain number of monitors and paths associated with the monitors already. After removing the already covered links, we continue to use the greedy algorithm to add more and more monitors until all the links are covered.

Although it is hard to prove the bound for the greedy-assisted LP algorithm, we expect it to be more efficient compared to the pure greedy algorithm because of the good starting point. As shown in our experimental results (See Section 6), this hybrid approach is better than the pure greedy algorithm in terms of minimizing the number of monitors. Additionally, the greedy algorithm sometimes fails to select monitors that cover all the links under the operational constraints simply because it does not try to balance the loads on nodes and links.

3. Multi-round Path Scheduling

We now introduce the path scheduling algorithm. It is worth mentioning that the path scheduling problem itself is also an NP-hard problem. We can reduce the well-known minimum graph coloring problem (which is NP-hard [15]) to our path scheduling problem [4] and hence finding the optimal schedule of the path measurements is not feasible. In this paper, we propose an integer linear programming (ILP) with relaxation to solve the scheduling problem. Meanwhile, we also include two other straightforward and simpler scheduling algorithms for comparison, a simple randomized algorithm and a greedy algorithm. The simple randomized algorithm and the ILP-based algorithm have nice theoretical stochastic bounds on the results, and the greedy algorithm clearly has the optimization goal as the ILP-based algorithm. Although theoretically we cannot prove the ILP-based algorithm with relaxation is the best of the three, our simulation results on practical scenarios shows the advantages of the ILP-based algorithm.

Note that node constraints are easy to satisfy because monitors are independent in terms of the node constraints. However in some extreme cases, there may be some link constraint violations in some rounds even if we have the optimal scheduling algorithm. Therefore, in such cases our scheduling algorithm tries to minimize the constraint violations. We define the link violation degree of a link as $\frac{n}{b} - 1(n > b)$ where n is the scheduled number of paths over the link and b is the link constraint of the link. We consider two metrics that quantify the violation degree: 1) maximum link violation degree (MLVD); 2) total link violation degree (TLVD).

1) Simple Randomized Algorithms: For any path p to be measured, we simply randomly select a round of the R rounds and schedule to measure the path p in this round. To do the random scheduling for a path, we just use a random function which generates a number t within $[0, R]$ with uniform distribution. Suppose the

[4] Imagine a round as a color, a path as a vertex and let two paths share a link if the corresponding vertices have an edge. Details are omitted for space limit.

integer number k satisfies $k-1 \leq t < k$, then we put the path to be measured in the kth round.

In the sense of expectation, the randomized scheduling results comply to the monitor constraints and link bandwidth constraints in each round. For example, the monitor i will monitor no more than $N \times c_i$ paths in total, hence in every round at most c_i paths from the monitor i are expected to be measured. However, for example, in a randomized instance, a monitor may monitor paths more than expected and hence the node constraint is violated. Similarly, we can apply Theorem 2 to quantify the violation degree and possibility for node constraints and link constraints.

2) Greedy Algorithm: The second algorithm we propose is a greedy algorithm. Basically, the greedy algorithm adds paths to the possible rounds of measurement, trying to minimize the violations of the system's constraints. It is easy for a greedy algorithm to schedule the path measurement so that monitor's constraints are all satisfied. However, link constraint violations may happen in some cases. Therefore, we let the object function of our greedy algorithm to minimize the maximum link violation degree or the total link violation degree of all the links. In each step, the greedy algorithm picks a path (randomly) in the measurement set and put the path to a certain round so that monitor constraints are not violated and the maximum (or total) link violation degree so far is minimized.

3) LP based Randomized Algorithm: The third algorithm we propose is to use integer linear programming first, and then use the relaxation and random rounding algorithm described in Section 3.2.2 to convert it to linear programming. The objective function is minimizing the maximum link violation degree or the total link violation degree, which is the same as the greedy algorithm (See Section 3.3.2). Let $y_{ijr} = 1$ if path P_{ij} is scheduled to be measured in round r, and $y_{ijr} = 0$ otherwise. The integer linear programming is formulated to minimize the maximum link violation degree:

$$
\begin{array}{lll}
P: & \text{Minimize} & v \\
 & s.t. & \sum_r y_{ijr} = 1, \ \forall i, j \\
 & & \sum_j y_{ijr} \leq c_i, \ \forall i \\
 & & \displaystyle\sum_{\forall i,\ \forall j,\ L_k \in P_{ij}} y_{ijr} - b_k \leq v \times b_k, \ \forall k, r \\
 & & y_{ijr} \in \{0, 1\}
\end{array}
\tag{9}
$$

Minimizing the total link violation degree is very similar so we omit the formula for the interest of space. Also we can apply Theorem 2 to quantify the violation degree and possibility for node constraints and link constraints after random rounding.

4. VScope Operation: Continuous Monitoring and Diagnosis

1. Overview

In Section 3, we introduced the algorithms for selecting routers to install monitors. After monitors are installed, VScope continuously monitors the performance of the networks round by round. Each round contains the following two stages:

Stage 1: Path monitoring. Our monitor selection algorithm gives the set of monitors and paths to measure in order to cover all (or majority of) the links under the operational constraints. In the first stage, VScope just instruments these monitors to measure the selected path and collect the measurement information.

Stage 2: Faulty link diagnosis. If paths are identified as faulty in the first stage, there must be faulty links on those paths. In the second stage, VScope diagnoses which links are faulty. Although we can try to infer the lossy links solely based on the measurement results of the first stage with existing approaches [2, 6, 16]–[18], the measurements are often insufficient to give the best diagnosis granularity or accuracy for the specific faulty paths.

Based on the observation that Internet congestions usually have some constancy [10], we assume that the faulty link discovered in the first stage will remain faulty in the second stage, which starts right after the first stage. Therefore, in the second stage, VScope selects a minimal extra set of paths to measure which, when combines with the first stage measurement results, gives the best diagnosis granularity and accuracy. For diagnosis, we focus on loss rate inference in this paper. But our techniques can also be extended to other metrics such as delay.

2. Path Monitoring Stage

In the path monitoring stage, monitors send out probes on the pre-selected paths to measure path properties. Measurements from different monitors are expected to be executed during the same period. In VScope, the coordinator first assigns the measurement tasks to all the monitors (not necessary for it to be done simultaneously). Then at the beginning of the path monitoring stage, the coordinator sends a START command to all the monitors at nearly the same time. This ensures that all the monitors start the measurements within a short period. In case there are network dynamics, VScope may need to re-select the paths to ensure link coverage. We discuss more details about path re-selection in Section 5.2.

3. Faulty Link Diagnosis Stage

After faulty paths are reported in the first stage, we need to select the minimal number of extra paths to measure in order to locate the faulty links. We develop a linear algebra based approach to select the minimal number of paths which, when combined with the paths measured in stage 1, can give us the complete loss information about the networks and consequently the best diagnosis granularity and accuracy. Next, we will first give the background on the linear algebra model, and then introduce the algorithms.

1) Background on the Linear Algebra Model: Suppose that a network consists of s IP links. In the linear algebra model, a path is represented by a column vector $v \in \{0,1\}^s$, where the jth entry v_j is 1 if link j is on the path and 0 otherwise. Suppose link j drops packets with probability l_j. Then the loss rate p of a path represented by v is given by

$$1 - p = \prod_{j=1}^{s} (1 - l_j)^{v_j} \tag{10}$$

By taking logarithms on both sides of (10), we have

$$\log(1-p) = \sum_{j=1}^{s} v_j \log(1 - l_j) = \sum_{j=1}^{s} v_j x_j = v^T x \tag{11}$$

where $x \in \mathbb{R}^s$ is a column vector with elements $x_j = \log(1 - l_j)$ and v^T is the transpose of the row vector v.

Given the installed monitors and traffic isolation constraints, if there are r measurable paths in the network, we can form a rectangular matrix $G \in \{0,1\}^{r \times s}$. Each row of G represents a measurable path in the network: $G_{ij} = 1$ if path i contains link j, and $G_{ij} = 0$ otherwise. Let p_i be the end-to-end loss rate of the ith path, and $b \in \mathbb{R}^r$ be a column vector with elements $b_i = \log(1 - p_i)$. Then we have

$$Gx = b \tag{12}$$

The above linear algebraic model is also applicable for any additive metric, such as delay.

2) Incrementally Selecting Paths for Diagnosis: Given the measurement results of the path monitoring phase, we first apply the good path algorithm [6] to find out potential lossy links. The good path algorithm simply considers that all the links on non-lossy paths are also non-lossy and hence removes these good links and paths. Next, we obtain a path set which include all the paths that contain at least one potential lossy link. We call the path matrix of these paths G'. As used in [6], the basis of the matrix G', $\bar{G}'$, contains the same amount of information as the whole G' matrix for inferring the link level loss rates. Thus we just need to get the paths corresponding to a basis for the diagnosis purpose. It is desirable to measure all the paths simultaneously so that the faulty link(s) can be located quickly before the faulty link properties change remarkably. Hence we do not consider multi-round diagnosis in our current design. Meanwhile, the additional selected paths should also satisfy the node/link measurement constraint.

The constrained basis selection problem is NP-hard [19], and sometimes it may not have a solution. We designed a greedy algorithm and found that it works very well in practice. The algorithm works as follows:

For each unmeasured path, we first obtain its *path measurement capacity* by taking the minimum of the node constraints of the source and destination nodes, and the link constraints of all the links on the path. For example, if the source node can measure 10 paths, the destination node can measure 20 paths, and there are two links on the paths whose constraints translates to 12 and 8 paths respectively, then the measurement capacity of the path is 8.

We sort these paths by the path measurement capacity (denoted as c_i for path i). $\bar{G}'$ is set to be empty at the beginning. Then starting from the path with the largest c_i, we iteratively try to add the path (denoted as a vector v) to $\bar{G}'$ if v can expand the basis of $\bar{G}'$. If so, we select path v, update the remaining capacity of the nodes and links, and then pick the next path with the largest path measurement capacity.

We stop the iteration when the rank of Q is the same as the rank of G, or we run out of paths, *i.e.*, the greedy algorithm does not find the extra paths which can constitute a basis with Q, under these constraints. Though theoretically possible, we found such cases very rare with real topologies and reasonable measurement constraints.

For implementation, we use the basis expanding algorithm introduced in [4], but extend it with path selection priority and constraint satisfactory inspection. As in [4], the computational complexity is $O(rk^2)$ where r is the number of paths in G' and k is the rank of G'. In practice, our experiment shows that the algorithm finishes in less than 20s for dealing with G' of thousands of paths.

Alternatively, we also consider the Bayesian experimental design introduced in [7] for path selection in the faulty path diagnosis. The Bayesian experimental design can potentially give the best results under certain total measurement budgets. Listed as an open problem in [7], Bayesian experimental design with operation constraints will also be our future work.

3) Locating Faulty Links: After collecting measurement results of the newly selected paths in this stage, we next locate the faulty links. There are several existing works on diagnosis analysis [2, 6, 7] which can be applied in VScope. Among them, the Minimal Identifiable Link Sequence (MILS) [6] needs the least statistical assumptions, compared with most existing network tomography work [2, 16]–[18]. Therefore, in the current implementation, we adopt the MILS approach [6] and obtain good enough evaluation results (See Section 6.6), but other tomography approaches can be easily adopted in our framework.

5. Robustness and Adaptivity in Dynamic Scenarios

In previous sections, we have assumed that the network topology and routing are static. In reality, the networks are dynamic. For example, the network topology may change as the network expands, and routing changes may happen when routers or links fail. Therefore, our VScope system needs to be robust against the temporary or permanent changes, and be adaptive to the dynamics in the network.

1. Redundancy in Monitor Selection

Selecting redundant monitors are necessary to assure that VScope handles well the various dynamics in the network for the following reasons. First of all, a monitor or the router that the monitor is attached to may fail. As a result, some previously covered links might not be covered by any path of the remaining monitors. Secondly, new routers or links can be added into the network after the monitor selection has been done. Installing new monitors to cover the newly added links every time is costly and annoying.

A straightforward way to introduce redundancy is to require each link to be covered by multiple paths. Therefore, a small number of routing changes may not break the full coverage of links. To achieve such redundancy, we can simply change the greedy algorithm on calculating the progress of the new paths. As for the LP based algorithm, we change the Inequality (5) such that each link is covered by at least a certain number of paths.

Furthermore, considering the possibility of the failure of monitors, we can require that the multiple paths covering the same link are from two or more different monitors if possible. Again, greedy algorithm can be extended easily to achieve such redundancy, however, LP based algorithm may not be able to assure the monitor redundancy. It will be our future work to design and evaluate the detailed algorithms for supporting redundancy in VScope.

2. Reselecting Paths for Path Monitoring Stage

When the set of monitors change, or the set of paths of a monitor changes as the result of routing changes (such as OSPF weight change), the coordinator has to re-select the paths to measure for the path monitoring stage and redistribute the task assignment to all the monitors.

First, the measurement path selection is a simpler problem than the monitor selection because it is a special case of the monitor selection problem when the monitors are fixed. Naively, we can use the monitor selection algorithms presented in previous sections for this purpose. However, an incremental adjustment is clearly more desirable because incremental algorithms usually have less computational complexity and introduce less communications overhead. The communication overhead is due to the communication messages through which the coordinator distributes the measurement tasks to all the monitors. Since some incremental update can be used for the task distribution, the communication overhead is proportional to the change in the measurement tasks. As we mentioned in Section 5.1, some monitors have unused measurement capacity for redundancy purpose. Therefore, when a link is no longer covered due to a routing change, VScope first applies a simple heuristic algorithm on all paths containing the target link. If monitor M^* has the ability to measure one more path P^* containing the target link, then VScope adds P^* into M^*'s measurement task. On the other hand, if the heuristic algorithm fails, this indicates that some large-scale adjustments are necessary and VScope will re-select the paths to measure from scratch. We also apply this heuristic algorithm in the case of monitor failure.

6. Evaluation

In this section, we will first describe the evaluation methodology. After a study on the real topologies from a large tier-1 ISP for evaluation, we present the results of the baseline monitor selection, multi-round monitor selection, and path scheduling. Finally we show the diagnosis accuracy and the computation speed results.

Statistics	V1-EX	V2-EX	IP-BB	IP-EX
# of PE routers	100s	100s	100s	100s
# of P routers	100s	100s	100s	100s
# of CE routers	100000s	10000s	N/A	10000s
# of Links	100000s	10000s	1000s	10000s
# of VPNs	1000s	1000s	N/A	N/A

TABLE II Statistics of the IP and VPN Topologies

1. Evaluation Methodology

1) Topology Dataset: The tier-1 ISP we studied has one IP backbone in US (named *IP-BB* in the rest of the paper). Some customer routers connected to the IP-BB are managed by the ISP and available for attaching monitors. We call the extended IP backbone (IP-BB plus the customer routers) *IP-EX*. We call two VPN backbones of the tier-1 ISP *V1* and *V2*, and the two VPN extended networks with customer routers *V1-EX* and *V2-EX*, respectively. We evaluate VScope's performance for IP-BB, IP-EX, V1-EX, and V2-EX. Table II gives the orders of magnitude for the number of routers, links and VPNs in these topologies.

Table III describes the basic configuration and constraints we select for the baseline experiments. We use them as the default setup unless specified otherwise. After consulting with the ISP management team, the rule of thumb is to have one monitor send about 3000 probes per minute, and usually the probing frequency of one path is four probes per second. We set our constraints accordingly, *e.g.*, we set the monitor constraint as 12 paths. This means the monitor can measure 12 paths simultaneously. The link capacity in the backbone networks is pretty large, while the access links usually have low capacity. For example, for V1 topology, almost all backbone links have more than 150 Mbps capacity, and a good portion of links have 1 Gbps or larger, but in V1-EX most access link capacities are 1.54 Mbps.

2) Link Loss Rates Collection: We use the following approach to obtain realistic link loss rates, which will be used to guide our simulation of continuous monitoring and diagnosis. We obtain 12-day of link loss rate data by collecting the SNMP link loss data from the routers. The collector sends SNMP inquires to the routers every five minutes and retrieves the link loss rate data. Figure 13 shows the cumulative distribution function of the loss rates of all the time slots of the links. We define *lossy link ratio* as the percentage of links (out of all the links) that are simultaneously lossy in the same 5-min time slot. Intuitively, the lossy link ratio will be quite different during different time slots because of the diurnal and weekly pattern of network traffics. In Figure 14, we use two thresholds (1% and 0%) to classify "lossy" and "non-lossy" links. It shows that when only a link with loss rate no less than 1% is said to be lossy, in most (about 80%) time slots the lossy link ratios are less than 1%.

Number of paths a monitor can measure	12/round
Number of paths a replier can respond	24/round
Packet probing rates per path	4 pkt/s
Bandwidth consumed by each path measurement	1.6 Kbps
Percent of link bandwidth allowed for probing	1%

TABLE III Basic configuration and constraints

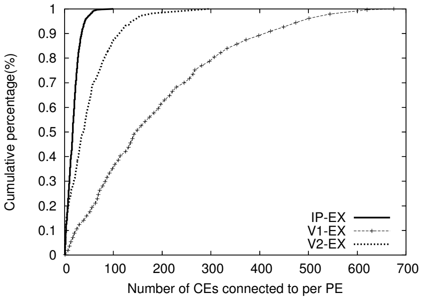

Fig. 3. Distribution of core router connections to customer routers

3) Evaluation Metrics: Our metrics include 1) the number of selected monitors in monitor setup phase; 2) maximum link violation degree or average link violation degree (See Section 3.3) in multi-round path scheduling ; 3) diagnosis granularity and accuracy in monitoring and diagnosis phase; and 4) running speed of the algorithms for monitor setup and faulty diagnosis.

Due to the anonymity requirement from the tier-1 ISP, we cannot provide the number of monitors or links in the studied topologies. So we only show the percentage of monitors selected and the percentage of links covered, *etc.*.

The diagnosis granularity is the average length of MILSes. To compare the inferred loss rate $\hat{p}$ with the real loss rate p of MILSes, we analyze both the absolute error and the error factor. The absolute error is $|p - \hat{p}|$. We use the error factor $F_\varepsilon(p, \hat{p})$ defined in [20] as $F_\varepsilon(p, \hat{p}) = \max\left\{\frac{p(\varepsilon)}{\hat{p}(\varepsilon)}, \frac{\hat{p}(\varepsilon)}{p(\varepsilon)}\right\}$ where $p(\varepsilon) = \max(\varepsilon, p)$ and $\hat{p}(\varepsilon) = \max(\varepsilon, \hat{p})$. Thus, p and $\hat{p}$ are treated as no less than ε. In our simulations we use the value ε = 0.003. $F_\varepsilon(p, \hat{p}) = 1$ indicates our estimation is accurate.

2. Star-like Topology

Interestingly, we found all the three backbone extended topologies studied in our experiments have a star-like topology.

- The backbone network is relatively small compared to the entire extended network. For example, the backbone network usually has hundreds of routers and thousands of links, while the number for the whole network is 1 to 2 orders of magnitude higher.

- There are a large number (tens or hundreds of thousands) of customer routers connecting to the PE routers with one access link each. On average, tens or even hundreds of customer routers connect to a single provider edge router.

Figure 3 shows the CDF of the degrees of PE routers that connect the customer routers in three real topologies. The average degree of PE routers in the IP backbone network is about 30, while in one VPN network the average degree of PE routers reaches 300. A nature question hence is: Is such star-like topology general in all or most ISP networks?

As shown in [21], typically an ISP's topology is designed based on technological and economic constraints. On one hand, a router can have a few high bandwidth connections or many low bandwidth connections or some combination in between. On the other hand, because it is cheaper to install and operate less number of links, traffic is aggregated at all levels of an ISP's network hierarchy, from its periphery all the way to its core. Meanwhile, there is a wide variability in customer's demand for network bandwidth and relatively low bandwidth is still widely needed. And the best place to deal with diverse user traffic is at the edge of the ISP network (*i.e.*, provider edge or PE routers). As a result PE routers tend to have high degrees. Therefore, we believe the star-like topology is very generic and prevalent in large ISP networks. This thought is also consistent with the large real and simulated topologies studied in [21].

3. *Baseline Monitor Selection Results*

In this section, we present the results of the single-round monitor selection algorithms of both the LP+Greedy algorithm and the pure greedy algorithms. We first present the baseline experiment results with the IP-BB backbone topology. And then we run more extensive experiments, varying the constraints and the topologies.

1) Results of Baseline Setup: We use the default configuration in Table III and run the two monitor selection algorithms (the LP+Greedy algorithm and pure greedy algorithm) on the IP-BB topologies. The LP+Greedy algorithm selects about 13% candidate routers as monitors while the pure greedy algorithm selects 14% routers as monitors. And both algorithms can cover all the links in the network. In the default configuration, we can see that the LP+Greedy algorithm performs a little bit better than the pure greedy algorithm.

2) Varying Monitor Constraints: Intuitively under certain monitor and link bandwidth constraints, the monitor selection algorithm may not be able to achieve 100% link coverage. Fortunately in our simulations, the algorithms can always achieve full link coverage and hence we only need to consider the number of selected monitors.

Figure 4 shows the percentage of routers that are selected as monitors given different monitor constraints. Clearly, for the LP+Greedy algorithm, the higher monitor constraint, the fewer monitors are required. However, there are some exceptions in the pure greedy algorithm. We believe this instability problem of the pure greedy algorithm lies in the nature of missing global optimization in the resource allocation. Overall, the LP+Greedy algorithm outperforms the pure greedy algorithm by selecting fewer monitors. In some cases, the greedy algorithm selects about 30% more monitors than the LP+Greedy algorithm (*e.g.* when a monitor can measure 16 paths simultaneously).

3) Varying Link Bandwidth Constraints: In this section, we vary the link bandwidth constraints with the IP-BB topology in the simulation. Usually the more link bandwidth is allowed for measurement, the larger flexibility for monitors to select paths to measure.

Figure 5 demonstrates how many routers are selected as monitors by the two monitor selection algorithms. Again, we find the LP+Greedy algorithm is better than the pure greedy algorithm, as the latter always selects more monitors. For example, when link constraint is 4% of link capacity, the LP+Greedy algorithm selects about 25% less monitors than the pure greedy algorithm. Interestingly, looser link constraints do not always result in fewer monitors for both algorithms. Again, locally optimized feature of the greedy algorithm may play an important role for such results.

4) Varying Topologies: We present the monitor selection results on different topologies in the following paragraphs. Note we only show the result of the pure greedy monitor selection algorithm. The linear programming based algorithm cannot scale to the extremely large network topologies which has hundreds of thousands of nodes and hundreds of millions of paths.

Figure 6 shows the number of monitors selected in different topologies while varying the monitor constraint. As we expected, for all topologies the percentage of routers selected as monitors drops as each monitor can measure more paths. Meanwhile the dropping rates become flat as monitor constraints increase.

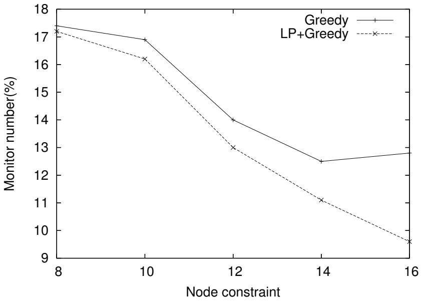

Fig. 4. Number of monitors as a function of node constraints in IP-BB

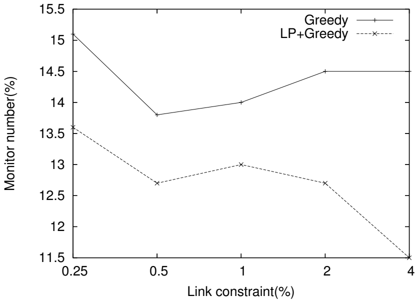

Fig. 5. Number of monitors as a function of link constraints in IP-BB

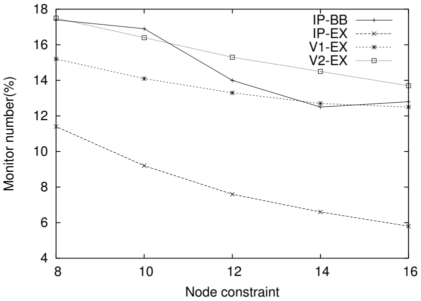

Fig. 6. Number of monitors as a function of node constraints in four topologies

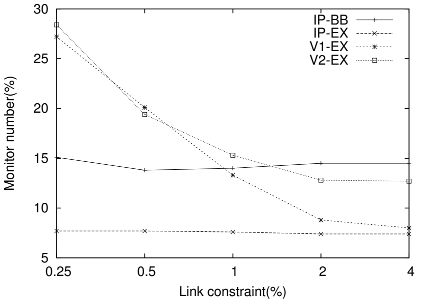

Fig. 7. Number of monitors as a function of link constraints in four topologies

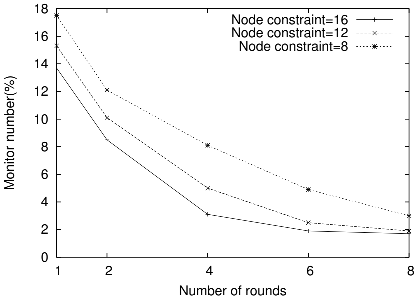

Fig. 8. Multi-round monitor selection with different node constraints

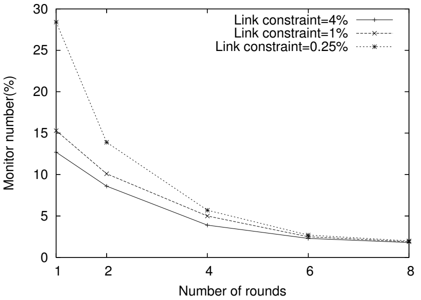

Fig. 9. Multi-round monitor selection with different link constraints

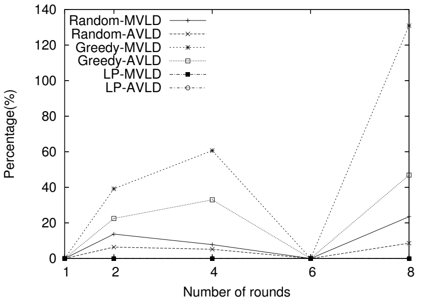

Fig. 10. Link violation degree of different algorithms

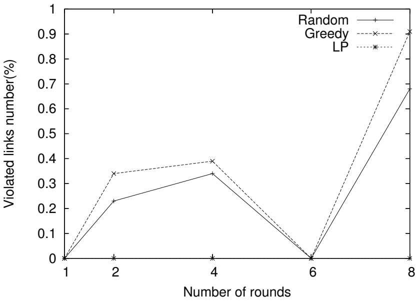

Fig. 11. Number of link violations of different algorithms

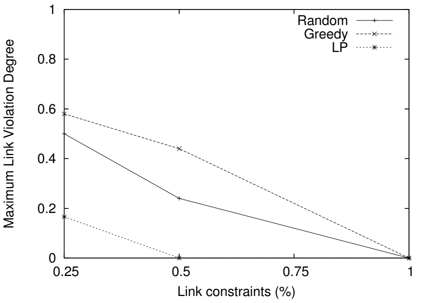

Fig. 12. Maximum link violation degree of different link constraints

Figure 7 shows the effect of link bandwidth constraints on the monitor selection. In the V1-EX and V2-EX topologies, link bandwidth constraints play a very important role. For example, in the V1-EX topology, less than 15% routers are selected as monitors if 1% link bandwidth is used for measurement; while the percentage of monitors increases to about 27% when only we use 0.25% link bandwidths for measurement. On the contrary, the IP-EX topology may have large link bandwidth and the monitor selection is not affected by the link bandwidth constraints at all. Since the configurations of the ISP measurement are also flexible (*e.g.* changing the probe rate on a path to vary the node constraints), it is reasonable to select a practical constraint configuration to achieve a good tradeoff between the deployment cost and monitoring performance.

4. Multi-round Monitor Selection Results

In this section, we present the simulation results of the multi-round monitor selection algorithm and the three multi-round scheduling algorithms on the V1-EX topology and omit the similar results of other topologies. As described in Section 3.3, there can be two different optimization goals of the greedy and LP-based scheduling algorithm: minimizing the Maximum Link Violation Degree (MLVD) and minimizing the Total Link Violation Degree (TLVD). We present the simulation results of the both goals in the following simulations.

We simulate the three multi-round monitor selection algorithms under the baseline setup (See Table III) first, and then vary the configurations such as link bandwidth constraints. We also vary the number of rounds from one to eight to show the efficiency of the multi-round monitor selection algorithm.

1) Monitor Selection Results: Figures 8 and 9 show the number of monitors selected under different simulation setups. Clearly, the percentage of routers selected as monitors decreases as the number of rounds increases. For example, in the baseline setup (*i.e.*, monitor constraint is 12), with round number as four we select only 6.2% routers as monitors, which is half of that selected by the single-round algorithm. We also run our multi-round algorithm for synthetic random graphs(*e.g.*, same number of node size, *etc.*) of the Barabasi-Albert model and Waxman model generated by the tool BRITE [22]. The result also proves multi-round can save lots of monitors. For example, when the number of round is 4, the system only needs about half of the monitors in single-round selection. However, Figures 8 and 9 also show that more rounds do not save many monitors when the number of rounds is more than four. Actually, the multi-round approach is a way of relaxing the constraints of the monitoring, and there is a minimum number of required monitors even without any constraints. In our topologies, we find the round number of four is a good trade-off between the cost of monitors (*i.e.*, number of monitors) and the measurement frequency.

5. Multi-round Scheduling Algorithm Results

1) Comparing different scheduling algorithms: We first compare the three scheduling algorithms, simple random algorithm, greedy algorithm and LP-based algorithm using the maximum link violation degree as the optimization goal. Note in the baseline setup, link violation is always zero for all the three algorithms, so we show the comparison results under a tighter constraint setup for comparison where only 0.25% link bandwidth can be used for measurements.

Figure 10 shows the maximum link violation degree (MLVD) and average link violation degree (ALVD) of the three algorithms while varying the number of rounds. Clearly, LP-based algorithm works the best, as it always has no violation in every setup. Surprisingly, simple random algorithm outperforms the greedy algorithm. Note for the simple random algorithm, we run the algorithm with different random seeds for several times and pick the best randomized result. So this suggests that randomization is quite helpful in our cases, while the simple greedy algorithm may be far from global optimization. Figure 11 shows percentage of links that link constraint violation happen after scheduling. The figure shows that the violation chances are very rare, *e.g.*, even in the worst case less than 1% links have constraint violation after scheduling. These results show that in practice the scheduling algorithms work very well and make no or acceptable link constraint violations.

2) Different optimization goals: For the greedy and LP-based algorithms, we can choose to minimize the maximum link violation degree or to minimize the total link violation degree. Generally speaking, optimizing the worst case and the total violations may be conflicting with each other, however, we find that in our simulations the violation results (maximum and average link violation degree) are nearly the same, no matter which optimization goal is chosen. One possible reason is that the violations are very rare, and hence the two goals are nearly equivalent.

3) Varying link bandwidth constraints: Figure 12 shows the maximum link violation degree of the three scheduling algorithms under different link bandwidth constraints. We fix the number of rounds to be four. Clearly, when the link bandwidth constraints become tighter, the scheduling algorithm tends to have more violations. This is reasonable as the scheduling problem becomes harder when the resources are more limited. Figure 12 also shows even when the link constraints are set to be unreasonably small, the maximum link violations of the three algorithms are still acceptable.

6. Continuous Monitoring and Diagnosis

1) Extra Overhead for Diagnosis: As described in Section 4.3, in diagnosis stage, we measure some extra paths than the paths measured in the path monitoring stage to get accurate diagnosis results.

We run our simulations 40 times and get the number of extra paths we need to measure in the diagnosis stage. As the loss rate assignments in every simulation are random, we obtain different number of extra paths in different simulations.

Figure 15 shows that in about 90% of cases, the number of paths monitored in diagnosis stage is only 3-10% of the paths that are measured in the path monitoring stage. Therefore, in the diagnosis stage, the overhead is very low. Furthermore, we can always find the extra measurement paths to get the basis of G matrix even with these constraints. In our future work, we will consider how to utilize the available measurement ability in the diagnosis stage for the path monitoring stage of the next round measurement.

2) Diagnosis Granularity: Figure 16 shows the diagnosis granularity of MILS with 40 sets of different loss rates. The MILS is very “short”, *e.g.*, less than 1.15 on average.

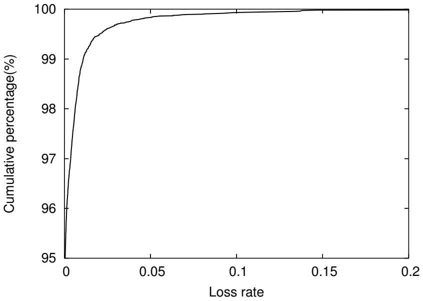

Fig. 13. Distribution of loss rates from real SNMP data

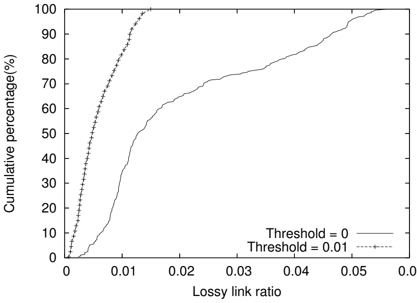

Fig. 14. Lossy link ratio from real SNMP data

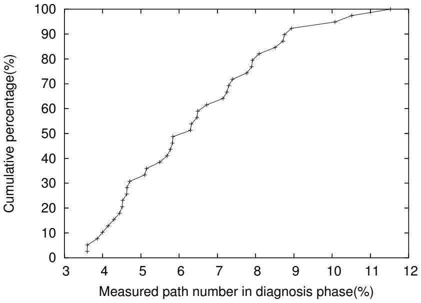

Fig. 15. Number of paths to measure in diagnosis phase

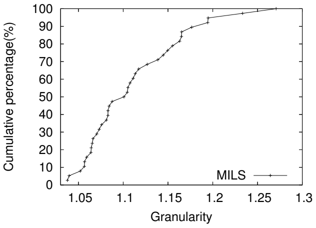

Fig. 16. Diagnosis granularity of MILS

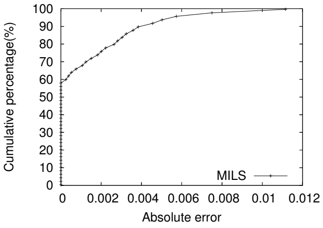

Fig. 17. Absolute inference error of MILS

3) Accuracy: In this section, we study the accuracy of the loss rate estimation of MILS. Basically, the loss rates of MILS are computed using the algorithm introduced in [6].

Figures 17 show the absolute error (defined in Section 6) of loss rate estimations of MILS. Meanwhile, the relative error is also very small and we omit the result for the interest of space.

We also checked the false negative and false positive of using MILS to locate lossy links. In our simulations, about 0.1% lossy links evade from the inferred lossy MILSes, and about 0.5% lossy MILSes actually do not contain any lossy links. Combining the granularity and accuracy results of MILS, we believe our diagnosis system serves good enough for the diagnosis purpose.

7. Computation Speed Results

In this section we present the speed for monitor selection phase and diagnosis phase. The experiments described above were conducted on a machine with Intel(R) Xeon(TM) 2.80GHz CPU. For IP-BB, LP+Greedy costs about 10 hours to choose the monitors, while the greedy algorithm needs about 5 minutes to finish this process. For the other three backoneExt topologies, the monitor selection phase costs about 4 hours using greedy algorithm for single round. And for the scheduling problem, LP based algorithm needs most time, *e.g.*, half an hour, while the simple random and greedy algorithms need only several seconds and minutes, respectively.

In diagnosis phase, after path monitoring stage, we need to figure out which paths to measure for diagnosis use. This path selection should be done in real time for quick diagnosis. With our linear-algebra based algorithm this process costs about 17.6 seconds on average, which satisfies the time requirement. In faulty link diagnosis stage, we locate lossy links using MILS. In average, calculating MILS needs about 14.7 seconds.

7. Related Work

A network monitoring and diagnosis system usually consists of two general components: experimental design and network inference [7]. The experimental design refers to the design on what or how to measure. For example, the experimental design selects the probe type (multicast or unicast) or the paths to measure. The network inference component specifies the algorithms to infer the unknown metrics based on the measurement results. The two components are tightly coupled, as the goal of experimental design is to favor the network inference.

The most related experimental designs in the literature are those monitor placement algorithms for tomography [1, 23]–[25]. For example, Bejerano *et al.* attempted to solve a simpler case of our monitoring selection problem [1], determining the smallest set of monitors whose probes can cover all traverse all links in the network. Most important constraints such as monitor, replier and link constraints are not considered, and this problem is still proven to be NP-hard in [1]. In [24] and [26], robustness problem is further considered to tolerate the routing dynamics. Previous work [4, 5, 7, 9] aimed at designing scalable measurement systems which select a subset of the paths and achieve the same or similar accuracy as if all the paths are measured. For example, Song *et al.* [7] introduced the Bayesian experimental design framework into network measurement. In [7], the problem is to choose the best set of paths to monitor in order to achieve the highest expected estimation accuracy given the constraint on the total number of monitored paths.

As mentioned in Section 3, the experimental design problem in our system is mainly to select the monitors as well as the paths under certain operation constraints. Compared to the previous works, our experimental design problem is unique in two aspects: 1) considering the installation cost of monitors, our approaches target to minimize the number of monitor; 2) to avoid interfering with the normal network traffic and take into account the monitor's ability, operation constraints are enforced on the path selection. These practical constraints make the monitor and path selection problem very challenging, even in this stage the selection goal is the simplest one, *i.e.*, to cover all the links. It will be our future work to study other selection goals in [4, 5, 7], which are more challenging under operational constraints.

The monitor selection problem with monitors' constraints may seem to be very common and related to many classic research topics such as placement of web cache replica or intrusion detection monitors. But these problems usually only have the monitor constraints (*e.g.* the load that monitors can take), while our monitor selection problem faces many other complex constraints such as link bandwidth constraints. We found the classic network (call) admission control problem [27] is somewhat related to our problem in terms of the link bandwidth constraints. But unlike our problem, the admission control problem does not include any monitor selection optimization problem.

8. Conclusion

In this paper, we propose VScope for continuously monitoring and diagnosis of VPN system under various operational constraints. VScope has two phases: monitor selection phase, and continuous monitoring and diagnosis phase. For the former, we develop several algorithms, in particular, a pure greedy algorithm and a greedy-assisted relaxed linear programming method, to select small number of monitors to cover all links spanning over the network under the constraints. For the large-scale and star-like VPN system, we design three multi-round scheduling algorithms to further reduce the number of monitors. Upon faulty paths are identified, a minimal number of extra paths are carefully chosen to further locate faulty links with high accuracy and precision. Evaluation based on data obtained from real IP and VPN networks managed by a large tier-1 ISP demonstrate the efficiency and effectiveness of VScope.

References

[1] Y. Bejerano and R. Rastogi, "Robust monitoring of link delays and faults in ip networks," in *IEEE INFOCOM*, 2003.

[2] V. Padmanabhan, L. Qiu, and H. Wang, "Server-based inference of Internet link lossiness," in *IEEE INFOCOM*, 2003.

[3] N. Duffield, "Simple network performance tomography," in *ACM SIGCOMM Internet Measurement Conference (IMC)*, 2003.

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

[5] D. B. Chua, E. D. Kolaczyk, and M. Crovella, "Efficient monitoring of end-to-end network properties," in *IEEE INFOCOM*, 2005.

[6] Y. Zhao, Y. Chen, and D. Bindel, "Towards unbiased end-to-end network diagnosis," in *ACM SIGCOMM*, 2006.

[7] H. Song, L. Qiu, and Y. Zhang, "Netquest: A flexible framework for lange-scale netork measurement," in *ACM SIGMETRICS*, June 2006.

[8] V. V. Vazirani, *Approximation Algorithms*, Springer-Verlag, 2001.

[9] C. Tang and P. McKinley, "On the cost-quality tradeoff in topology-aware overlay path probing," in *IEEE ICNP*, 2003.

[10] Y. Zhang et al., "On the constancy of Internet path properties," in *Proc. of SIGCOMM IMW*, 2001.

[11] E. Tsang, *Foundations of Constraint Satisfaction*, Academic Press, 1993.

[12] D. S. Hochbaum and A. Pathria, "Analysis of the greedy approach in problems of maximum k-coverage," *Naval Research Logistics*, vol. 45, 1998.

[13] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, Wiley-Interscience (3rd edition), 2004.

[14] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995.

[15] T. R. Jensen and B. Toft, *Graph coloring problems*, Wiley-Interscience, New York, 1995.

[16] R. Caceres, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal loss characteristics," *IEEE Transactions in Information Theory*, vol. 45, 1999.

[17] M. Coates, A. Hero, R. Nowak, and B. Yu, "Internet Tomography," *IEEE Signal Processing Magazine*, vol. 19, no. 3, pp. 47–65, 2002.

[18] N.G. Duffield, F.L. Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," in *IEEE INFOCOM*, 2001.

[19] G. W. Stewart, *Matrix Algorithms: Basic Decompositions*, Society for Industrial and Applied Mathematics, 1998.

[20] T. Bu, N. Duffield, F. Presti, and D. Towsley, "Network tomography on general topologies," in *ACM SIGMETRICS*, 2002.

[21] L. Li, D. Alderson, W. Willinger, and J. Doyle, "A first-principles approach to understanding the internet's router-level topology," in *ACM SIGCOMM*, 2004.

[22] A. Medina, I. Matta, and J. Byers, "On the origin of power laws in Internet topologies," in *ACM Computer Communication Review*, Apr. 2000.

[23] J.D. Horton and A. Lopez-Ortiz, "On the number of distributed measurement points for network tomography," in *ACM SIGCOMM Internet Measurement Conference (IMC)*, 2003.

[24] R. Kumar and J. Kaur, "Efficient beacon placement for network tomography," in *ACM SIGCOMM Internet Measurement Conference (IMC)*, 2004.

[25] G. R. Cantieni and et. al., "Reformulating the monitor placement problem: Optimal network-wide sampling," in *Conference on Information Sciences and Systems (CISS)*, 2006.

[26] Y. Breitbart, F. Dragan, and H. Gobjuka, "Effective network monitoring," in *International Conference on Computer Communications and Networks (ICCCN)*, 2004.

[27] N. Alon, S. Gutner, and Y. Azar, "Admission control to minimize rejections and online set cover with repetitions," in *Proc. of SPAA*, 2005.