2

A Greedy Scheme for Designing Delay Monitoring Systems of IP Networks

Yigal Bejerano[1] and Rajeev Rastogi[2]
[1]*Bell Laboratories, Alcatel-Lucent,*
[2]*Yahoo-Inc,*
[1]*USA*
[2]*India*

1. Introduction

The demand for sophisticated tools for monitoring network utilization and performance has been growing rapidly as Internet Service Providers (ISPs) offer their customers more services that require quality of service (QoS) guarantees and as ISP networks become increasingly complex. Tools for monitoring link delays and faults in an IP network are critical for numerous important network management tasks, including providing QoS guarantees to end applications (e.g., voice over IP), traffic engineering, ensuring service level agreement (SLA) compliance, fault and congestion detection and performance debugging. Consequently, there has been a recent flurry of both research and industrial activity in the area of developing novel tools and infrastructures for measuring network parameters.

Existing network monitoring tools can be divided into two categories. *Node-oriented* tools collect monitoring information from network devices (routers, switches and hosts) using SNMP/RMON probes [1] or the Cisco NetFlow tool [2]. These are useful for collecting statistical and billing information, and for measuring the performance of individual network devices (e.g., link bandwidth usage). However, in addition to the need for monitoring agents to be installed at every device, these tools cannot monitor network parameters that involve several components, like link or end-to-end path latency. The second category contains *path-oriented* tools for connectivity and latency measurement like `ping`, `traceroute` [3], `skitter` [4] and tools for bandwidth measurement such as `pathchar` [5], `Bing` [6], `Cprobe` [7], `Nettimer` [8] and `pathrate` [9]. As an example, `skitter` sends a sequence of probe messages to a set of destinations and measures the latency of a link as the difference in the round-trip times of the two probes to the endpoints of the link. A benefit of path-oriented tools is that they do not require special monitoring agents to be run at each node. However, a node with such a path-oriented monitoring tool, termed a *monitoring station*, is able to measure latencies and monitor faults for only a limited set of links in the node's routing tree, e.g., its *shortest path tree* (SPT). Thus, monitoring stations need to be deployed at a few strategic points in the ISP or Enterprise IP network so as to maximize network coverage, while minimizing hardware and software infrastructure cost, as well as maintenance cost for the stations. Consequently, any monitoring system needs to satisfy two basic requirements.

Source: Advances in Greedy Algorithms, Book edited by: Witold Bednorz,
ISBN 978-953-7619-27-5, pp. 586, November 2008, I-Tech, Vienna, Austria

1. *Coverage* - The system should accurately monitor all the links and paths in the network.
2. *Efficiency* - The systems should minimize the overhead imposed by monitoring on the underlying production network.

The chapter proposes an efficient *two-phased* approach for fully and efficiently monitoring the latencies of links and paths using path-oriented tools. Our scheme ensures complete coverage of measurements by selecting monitoring stations such that each network link is in the routing trees of some monitoring station. It also reduces the monitoring overhead which consists of two costs: the infrastructure and maintenance cost associated with the monitoring stations, as well as the additional network traffic due to probe packets. Minimizing the latter is especially important when information is collected frequently in order to continuously monitor the state and evolution of the network. In the *first phase*, the scheme addresses the *station selection problem*. This phase seeks for the locations of a minimal set of monitoring stations that are capable to perform all the required monitoring tasks, such as monitoring the delay of all the network links. Subsequently, in the *second phase*, the scheme deals with the *probe assignment problem*, which computes a minimal set of probe messages transmitted by each station for satisfying the monitoring requirements.

Although, the chapter focuses primarily on delay monitoring, the presented approach is more generally applicable and can also be used for other management tasks. We consider two variants of monitoring systems. A *link monitoring* (LM) system that guarantees that very link is monitored by a monitoring station. Such system is useful for delay monitoring, bottleneck links detection and fault isolation, as demonstrated in [10]. A *path monitoring* (PM) system that ensures the coverage of every routing path between any pair of nodes by a single station, which provides accurate delay monitoring.

For link monitoring we show that the problem of computing the minimum set of stations whose routing trees (e.g, its shortest path trees), cover all network links is NP-hard. Consequently, we map the station selection problem to the set cover problem [11], and we use a polynomial-time greedy algorithm that yields a solution within a logarithmic factor of the optimal one. For the probe assignment problem, we show that computing the optimal probe set for monitoring the latency of all the network links is also NP-hard. To this problem, we devise a polynomial-time greedy algorithm that computes a set of probes whose cost is within an factor of 2 of the optimal solution. Then, we extend our scheme to path monitoring. Initially, we show that even when the number of monitoring stations is small (in our example only two monitoring stations) every pair of adjacent links along a given routing path may be monitored by two different monitoring stations. This raises the need for a path monitoring system in which every path is monitored by a single station. For station selection we devise a set-cover-based greedy heuristic that computes solutions with logarithmic approximation ratio. Then, we propose a greed algorithm for probe assignment and leave the problem of constructing an efficient algorithm with low approximation ratio for future work.

The chapter is organized as follows. It starts with a brief survey of related work in Section 2. Section 3 presents the network model and a description of the network monitoring framework is given in Section 4. Section 5 describes our link monitoring system and Section 6 extends our scheme to path monitoring. Section 7 provides simulation results that demonstrate the efficiency of our scheme for link monitoring and Section 8 concludes the chapter.

2. Related work

The need for low-overhead network monitoring techniques has gained significant attention in the recent years and below we provide the most relevant studies to this chapter. The network proximity service project, SONAR [12], suggests to add a new client/server service that enables hosts to obtain fast estimations of their distance from different locations in the Internet. However, the problem of acquiring the latency information is not addressed. The IDmaps [13] project produces "*latency maps*" of the internet using special measurement servers called *tracers* that continuously probe each other to determine their distance. These times are subsequently used to approximate the latency of arbitrary network paths. Different methods for distributing tracers in the internet are described in [14], one of which is to place them such that the distance of each network node to the closest tracer is minimized. A drawback of the IDMaps approach is that latency measurements may not be accurate. Essentially, due to the small number of paths actually monitored, it is possible for errors to be introduced when round-trip times between tracers are used to approximate arbitrary path latencies. In [15], Breitbart *et al.* propose a monitoring scheme where a single *network operations center* (NOC) performs all the required measurements. In order to monitor links not in its routing tree, the NOC uses the IP source routing option to explicitly route probe packets along these links. The technique of using source routing for determining the probe routes has been used by other proposals as well for both fault detection [16] and delay monitoring [17]. Unfortunately, due to security problems, many routers frequently disable the IP source routing option. Further, routers usually process IP options separately in their CPU, which in addition to adversely impacting their performance, also causes packets to suffer unknown delays. Consequently, approaches that rely on explicitly routed probe packets for delay and fault monitoring may not be feasible in today's ISP and Enterprise environments. Another delay monitoring approach was presented by Shavit *et al.* in [18]. They propose to solve a linear system of equations to compute delays for smaller path segments from a given a set of end-to-end delay measurements for paths in the network.

The problem of station placement for delay monitoring has been addressed by several studies. In [19], Adler *et al.* focus on the problem of determining the minimum cost set of multicast trees that cover links of interest in a network, which is similar to the station selection problem tackled in this chapter. The two-phase scheme of station placement and probe assignment have been proposed in [10]. In this work, Bejerano and Rastogi show a combined approach for minimizing the cost of both the monitoring stations as well as the probe messages. Moreover, they extend their scheme for delay monitoring and fault isolation in the presence of multiple failures. In [20] Breitbart *et al.* consider two variants of the station placement problem assuming that the routing tree of the nodes are their shortest path trees (SPTs). In the first variant, termed A-Problem, the routing trees of a node may be any one of its SPT, while in the second variant, called E-Problem, the routing tree of a node can be selected among all the possible SPTs for minimizing the monitoring overhead. For both variant they have shown that the problems are NP-hard and they provided approximation algorithms. In [21] Nguyen and Thiran developed a technique for locating multiple failures in IP networks using active measurement. They also proposed a two-phased approach, but unlike the work in [10], they optimize first the probe selection and only then they compute the location of a minimal set of monitoring stations that can generate these probes. Moreover, by using techniques from a max-plus algebra theory, they show that the optimal set of probes can be determined in polynomial time. In [22], Suh *et al.*

propose a scheme for cost-effective placement of monitoring stations for passive monitoring of IP flows and controlling their sampling rate. Recently, Cantieni *et al.* [23], reformulate the monitoring placement problem. They assume that every node may be a monitoring station at any given time and then they ask the question which monitors should be activated and what should be their sampling to achieve a given measurement task? To this problem they provide optimal solution.

3. Network model

We model the Service Provider or Enterprise IP network by an undirected graph $G(V,E)$, where the graph nodes, V, denote the network routers and the edges, E, represent the communication links connecting them. The number of nodes and edges is denoted by $|V|$ and $|E|$, respectively. Further, we use $P_{s,t}$ to denote the path traversed by an IP packet from a source node s to a destination node t. In our model, we assume that packets are forwarded using standard IP forwarding, that is, each node relies exclusively on the destination address in the packet to determine the next hop. Thus, for every node $x \in P_{s,t}$, $P_{x,t}$ is included in $P_{s,t}$. In addition, we also assume that $P_{s,t}$ is the routing path in the opposite direction from node t to node s. This, in turn, implies that for every node $x \in P_{s,t}$, $P_{s,x}$ is a prefix of $P_{s,t}$. As a consequence, it follows that for every node $s \in V$, the subgraph obtained by merging all the paths $P_{s,t}$, for every $t \in V$, must have a tree topology. We refer to this tree for node s as the *routing tree* (RT) of node s and denote it by T_s. Note that tree T_s defines the routing paths from node s to all the other nodes in V and vice versa.

Observe that for a Service Provider network consisting of a single OSPF area, the RT T_s of node s is its shortest path tree (SPT). However, for networks consisting of multiple OSPF areas or autonomous systems (that exchange routing information using BGP), packets between nodes may not necessarily follow shortest paths. In practice, the topology of RTs can be calculated by querying the routing tables of nodes. In our solution, the routing tree of node s may be its SPT but this is not an essential requirement.

We associate a positive cost $c_{u,v}$ with sending a message between any pair of nodes $u, v \in V$. For every intermediate node $w \in P_{u,v}$ both $c_{u,w}$ and $c_{v,w}$ are at most $c_{u,v}$ and $c_{u,w} + c_{v,w} \geq c_{u,v}$. Typical examples of this cost model are the fixed cost, where all messages have the same cost, and hop count, where the message cost is the number of hops in its route.

4. Network monitoring framework

In this section, we describe our methodology for complete IP network monitoring using path-oriented tools. Our primary focus is the measurement of round-trip latency of network links and paths. However, our methodology is also applicable for a wide range of monitoring tasks, like fault and bottleneck link detection, as presented in [10]. For monitoring the round-trip delay of a link $e \in E$, a node $s \in V$ such that e belongs to s's RT (that is, $e \in T_s$), must be selected as a monitoring station. Node s sends two probe messages[1] to the end-points of e, which travel almost identical routes except for the link e. On receiving a probe message, the receiver replies immediately by sending a probe reply message to the

[1] The probe messages are implemented by using "ICMP ECHO REQUEST/REPLY" messages similar to `ping`.

monitoring station. Thus, the monitoring station s can estimate the round-trip delay of the link by measuring the difference in the round-trip times of the two probe messages.

From the above description, it follows that a monitoring station can only measure the delays of links in its RT. Consequently, a monitoring system designated for measuring the delays of all network links has to find a set of *monitoring stations* $S \subseteq V$ and a *probe assignment* $A \subset S \times V$. A probe assignment is basically a set of pairs $\{(s, u) \mid s \in S, u \in V\}$ such that each pair (s, u) represents a probe message that is sent from the monitoring station s to node u. The set S and the probe assignment A are required to satisfy two constraints:

1. The *covering monitoring station set* constraint guarantees that all links are covered by the RTs of the nodes in S, *i.e.*, $\bigcup_{s \in S} T_s = E$.
2. The *covering probe assignment* constraint ensures that for every edge $e = (u, v) \in E$, there is a node $s \in S$ such that $e \in T_s$ and $\mathcal{A}$ contains the pairs[2] (s, u) and (s, v). In other words, every link is monitored by at least one monitoring station.

A pair $(S,\mathcal{A})$ that satisfies the above constraints is referred to as a *feasible solution*. In instances where the monitoring stations are selected from a subset $Y \subset V$, we assume that $\bigcup_{s \in Y} T_s = E$ which guarantees the existence of a feasible solution.

The overhead of a monitoring system is composed of two components, the overhead of installing and maintaining the monitoring stations and the communication cost of sending probe messages. In practice, it is preferable to have as few stations as possible since this reduces operational costs, and so we adopt a two-phased approach to optimizing monitoring overheads. In the first phase, we select an optimal set of monitoring stations, while in the second, we compute the optimal probes for the selected stations. Let w_v be the cost of selecting node $v \in V$ as a monitoring station. The *optimal station selection* S is the one that satisfies the covering monitoring station set requirement and minimizes the total cost of all the monitoring stations given be the sum $\sum_{s \in S} w_s$. After selecting the monitoring stations S, the *optimal probe assignment* $\mathcal{A}$ is one that satisfies the covering probe assignment constraint and minimizes the total probing cost defined by the sum $\sum_{(s,v) \in \mathcal{A}} c_{s,v}$. Note that choosing $c_{sv} = 1$ essentially results in an assignment $\mathcal{A}$ with the minimum number of probes, while choosing $c_{s,v}$ to be the minimum number of hops between s and v yields a set of probes that traverse the fewest possible network links.

A final component of our monitoring infrastructure is the *network operations center* (NOC) which is responsible for coordinating the actions of the set of monitoring stations S. The NOC queries the network nodes to determine their RTs, and subsequently uses these to compute a near-optimal set of monitoring stations and a probe assignment for them. In the following two sections, we develop approximation algorithms for the station selection and probe assignment problems. Section 5 considers the problem of monitoring links, while path monitoring is addressed in Section 6. Note that our proposed framework deals only with the aspect of efficient collection of monitoring information. It does not deal with the aspects of analyzing and distributing this information, which are application-dependent.

[2] If one of the end points of e is in S, let say $u \in S$, then $\mathcal{A}$ is only required to include the probe (u, v).

5. Link monitoring

We show in this section that for link monitoring both the station selection and probe assignment problems are NP-hard. Then, we present polynomialtime approximation algorithms for solving them. For station selection, we develop a $\ln(|V|)$-approximation algorithm where the lower bound is $1/2 \cdot \ln(|V|)$ and for probe assignment, we present a 2 approximation algorithm.

5.1 An efficient station selection algorithm

The problem addressed here is covering all the graph edges with a small number of RTs, and we consider both the un-weighted and the weighted versions of this problem.

Definition 1 (The Link Monitoring Problem - LM):

Given a graph $G(V,E)$ and a RT, T_v, for every node $v \in V$, find the smallest set $S \subseteq V$ such that $\bigcup_{v \in S} T_v = E$. □

Definition 2 (The Weighted LM Problem - WLM) :

Given a graph $G(V,E)$ with a non-negative weight w_v and a RT T_v for every node $v \in V$, find the set $S \subseteq V$ such that $\bigcup_{v \in S} T_v = E$ and the sum $\Sigma_{v \in S} w_v$ is minimum. □

We show a similarity between the link monitoring problem and the *set cover* (SC) problem, which is a well-known NP-hard problem [24]. An efficient algorithm for solving one of them can be also used to efficiently solve the other. Let us recall the SC problem. Consider an instance $I(Z, \mathcal{Q})$ of the SC problem, where $Z = \{z_1, z_2, \ldots, z_m\}$ is a universe of m elements and $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_n\}$ is a collection of n subsets of Z, (assume that $\bigcup_{Q \in \mathcal{Q}} Q = Z$). The SC problem seeks to find the smallest collection of subsets $\mathcal{S} \subseteq \mathcal{Q}$ such that their union contains all the elements in Z, *i.e.*, $\bigcup_{Q \in \mathcal{S}} Q = Z$. At the weighted version of the CS problem, each one of the subsets $Q \in \mathcal{Q}$ has a cost w_Q and the optimal solution is the *lowest-cost* collection of subsets $\mathcal{S} \subseteq \mathcal{Q}$, such that their union contains all the elements in Z. For SC problem the greedy heuristic [11] is a commonly used approximation algorithm and it achieves a tight approximation ratio of $\ln(k)$, where k is the size of the biggest set $Q \in$ Q. Note that in the worst case $k = m$.

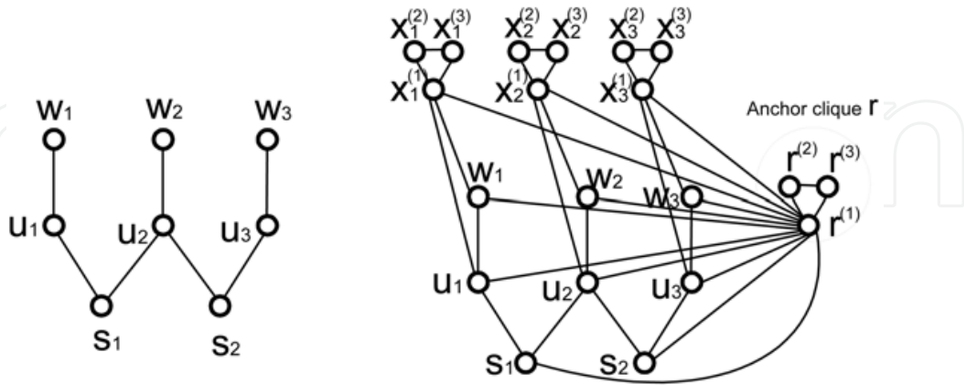


Fig. 1. The graph $G_{\mathcal{R}(I)}(V,E)$ for the given instance of the SC problem.

5.1.1 Hardness of the LM and WLM problems

Theorem 1 *The LM and WLM problems are NP-hard, even when the routing tree (RT) of each node is restricted to be its shortest path tree (SPT).*

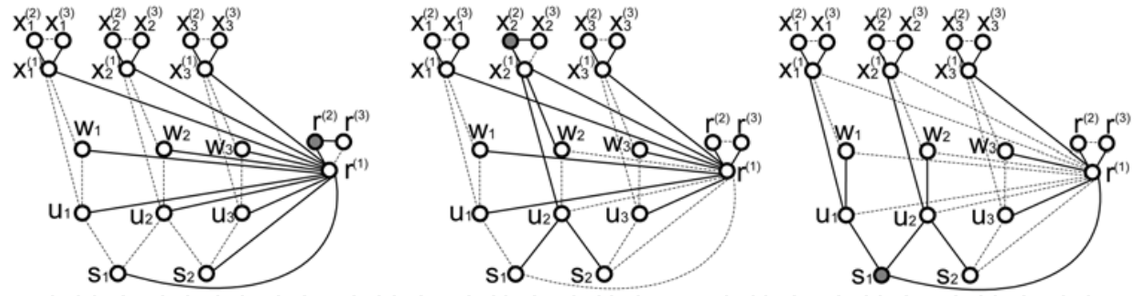


Fig. 2. The RTs of nodes $r^{(2)}$, $x_2^{(2)}$ and s_1.

Proof: We show that the LM problem is NP-hard by presenting a polynomial reduction from the *set cover* problem to the LM problem. From this follows that also the WLM problem is NP-hard. Consider an instance $I(Z,\mathcal{Q})$ of the SC problem. Our reduction $\mathcal{R}(I)$ constructs the graph $G_{\mathcal{R}(I)}(V,E)$ where the RT of each node $v \in V$ is also its shortest path tree. For determining these RTs, each edge is associated with a weight[3], and the graph contains the following nodes and edges. For each element $z_i \in Z$, it contains two connected nodes u_i and w_i. For each set $Q_j \in \mathcal{Q}$, we add a node, labeled by s_j, and the edges (s_j, u_i) for each element $z_i \in Q_j$. In addition, we use an auxiliary structure, termed an *anchor clique* x, which is a clique with three nodes, labeled by $x^{(1)}$, $x^{(2)}$ and $x^{(3)}$, and only node $x^{(1)}$ has additional incident edges. For each element $z_i \in Z$, the graph $G_{\mathcal{R}(I)}$ contains one anchor clique x_i whose attachment point, $x_i^{(1)}$, is connected to the nodes u_i and w_i. The weights of all the edges described above is 1. Finally, the graph $G_{\mathcal{R}(I)}$ contains an additional anchor clique r that is connected to the remaining nodes and anchor cliques of the graph, and the weights of these edges is $1 + \epsilon$. An example of such a graph is depicted in Figure 1 for an instance of the SC problem with 3 elements $\{z_1, z_2, z_3\}$ and two sets $Q_1 = \{z_1, z_2\}$ and $Q_2 = \{z_2, z_3\}$.

We claim that there is a solution of size k to the given SC problem if and only if there is a solution of size $k+m+1$ to the LM instance defined by the graph $G_{\mathcal{R}(I)}(V,E)$. We begin by showing that if there is a solution to the SC problem of size k then there exists a set S of at most $k+m+1$ stations that covers all the edges in $G_{\mathcal{R}(I)}$. Let the solution of the SC problem consist of the sets $Q_{i_1}, \ldots, Q_{i_k}$. The set S of monitoring stations contains the nodes $r^{(2)}$, $x_i^{(2)}$ (for each element $z_i \in Z$) and $s_{i_1}, \ldots, s_{i_k}$. We show that the set S contains $k + m + 1$ nodes that cover all the graph edges. The tree $T_{r^{(2)}}$ covers edges $(r^{(1)}, r^{(2)})$, $(r^{(2)}, r^{(3)})$, all edges $(u_i, r^{(1)})$, $(w_i, r^{(1)})$, $(x_i^{(1)}, r^{(1)})$, $(x_i^{(1)}, x_i^{(2)})$, $(x_i^{(1)}, x_i^{(3)})$, for each element z_i, and the edges $(s_j, r^{(1)})$ for every set $Q_j \in \mathcal{Q}$. An example of such a $T_{r^{(2)}}$ is depicted in Figure 2-(a). Similarly, for every $z_i \in Z$, the RT $T_{x_i^{(2)}}$ covers edges $(x_i^{(2)}, x_i^{(3)})$, $(x_i^{(2)}, x_i^{(1)})$, $(x_i^{(1)}, u_i)$ and $(x_i^{(1)}, w_i)$. $T_{x_i^{(2)}}$ also covers all edges (s_j, u_i) for every set Q_j that contains element z_i, and edges $(r^{(1)}, r^{(2)})$ and $(r^{(1)}, r^{(3)})$. An example of the RT $T_{x_2^{(2)}}$ is depicted in Figure 2-(b). Thus, the only remaining uncovered edges are (u_i, w_i), for each element z_i. Since Q_{i_j}, $j = 1, \ldots, k$, is a solution to the SC problem, these edges are covered by the RTs $T_{s_{i_j}}$, as depicted in Figure 2- (c). Thus, S is a set of at most $k + m + 1$ stations that covers all the edges in the graph $G_{\mathcal{R}(I)}$.

[3] These weights do not represent communication costs.

Next, we show that if there is a set of at most k+m+1 stations that covers all the graph edges then there is a solution for the SC problem of size at most k. Note that there needs to be a monitoring station in each anchor clique and suppose w.l.o.g that the selected stations are $r^{(2)}$ and $x_i^{(2)}$ for each element z_i. None of these m + 1 stations covers edges (u_i,w_i) for elements $z_i \in Z$. The other k monitoring stations are placed in the nodes u_i,w_i and s_j. In order to cover edge (u_i,w_i), there needs to have a station at one of the nodes u_i, w_i or s_j for some set Q_j containing element z_i. Also, observe that the RTs of u_i and w_i cover only edge (u_i,w_i) for element z_i and no other element edges. Similarly, the RT of s_j covers only edges (u_i,w_i) for elements z_i contained in set Q_j . Let $\mathcal{S}$ be a collection of sets defined as follows. For every monitoring station at any node s_j add the set $Q_j \in \mathcal{Q}$ to $\mathcal{S}$, and for every monitoring station at any node u_i or w_i we add to $\mathcal{S}$ an arbitrary set $Q_j \in \mathcal{Q}$ such that $z_i \in Q_j$. Since the set of monitoring stations cover all the element edges, the collection $\mathcal{S}$ covers all the elements of Z, and is a solution to the SC problem of size at most k. □

The above reduction $\mathcal{R}(I)$ can be extended to derive a lower bound for the best approximation ratio achievable by any algorithm. This reduction and the proof of Theorem 2 are given in [25].

Theorem 2 *The lower bound of any approximation algorithm for the LM problem is* $\frac{1}{2} \cdot \ln(|V|)$.

5.1.2 A greedy algorithm for the LM and WLM problems

We turn to present an efficient algorithm for solving the LM and the WLM problems. The algorithm maps the given instance of LM or WLM problem to an instance of the SC problem and uses a greedy heuristic for solving the SC instance, which achieves a near tight upper bound for the LM and WLM problems.

$\mathcal{S} \leftarrow \emptyset$, $C \leftarrow Z$
While $C \neq \emptyset$ do
    For every $Q_v \in \mathcal{Q} - \mathcal{S}$ do
        $n_v \leftarrow |Q_v \bigcap C|$
    $Q \leftarrow arg\min_{Q_v, Q_v \notin \mathcal{S}} \frac{w_v}{n_v}$
    $\mathcal{S} \leftarrow \mathcal{S} \bigcup \{Q\}$
    $C \leftarrow C - Q$
End While
Return $\mathcal{S}$

Fig. 3. A formal description of the Greedy Heuristic for Set-Cover.

For a given WLM problem involving the graph $G(V,E)$ we define an instance of the SC problem as follows. The set of edges, E, defines the universe of elements, Z. The collection of sets $\mathcal{Q}$ includes the subsets $Q_v = \{e \mid e \in T_v\}$ for every node $v \in V$, where the weight of each subset Q_v is equal to w_v, the weight of the corresponding node v. The greedy heuristic is an iterative algorithm that selects in each iteration the most cost-effective set. Let $C \subseteq Z$ be the set of uncovered elements. In addition, let $n_v = \{Q_v \cap C\}$ be the number of uncovered elements in the set Q_v, for every $v \in V$, at the beginning of each iteration. The algorithm works as follows. It initializes $C \leftarrow Z$. Then, in each iteration, it selects the set Q_v with the

minimum $\frac{w_v}{n_v}$ ratio and removes all its elements from the set C. This step is done until C becomes empty. A formal description of the algorithm is presented in Figure 3.

Theorem 3 *The greedy algorithm computes a* $\ln(|V|)$*-approximation for the LM and WLM problems.*

Proof: According to [11], the greedy algorithm is a H(d)-approximation algorithm for the SC problem, where d is the size of the biggest subset and $\mathcal{H}(d) = \sum_{i=1}^{d} \frac{1}{i}$, is the harmonic sequence. For the LM and WLM problems, every subset includes all the edges of the corresponding RT and its size is exactly $|V| - 1$. Hence, the approximation ratio of the greedy algorithm is $H(|V| - 1) \leq \ln(|V|)$.

Note that the worst-case time complexity of the greedy algorithm can be shown to be $O(|V|^3)$.

5.2 An efficient probe assignment algorithm

Once we have selected a set S of monitoring stations, we need to compute a probe assignment $\mathcal{A}$ for measuring the latency of the network links. Recall from Section 4 that a *feasible probe assignment* is a set of pairs $\{(s, u) \mid s \in S, u \in V\}$. Each pair (s, u) represents a probe message that is sent from station s to node u and for every edge $e = (u, v) \in E$, there is a station $s \in S$ such that $e \in T_s$ and $\mathcal{A}$ contains the pairs (s, u) and (s, v). The cost of a probe assignment $\mathcal{A}$ is $COST_{\mathcal{A}} = \sum_{(s,u) \in \mathcal{A}} c_{s,u}$ and the *optimal probe assignment* is the one with the minimum cost.

5.2.1 Hardness of the probe assignment problem

In the following, we show that computing the optimal probe assignment is NP-hard even if we choose all $c_{s,u} = 1$ that minimizes the number of probes in $\mathcal{A}$. A similar proof can be used to show that the problem is NP-hard for the case when $c_{s,u}$ equals the minimum number of hops between s and u (this results in a set of probes traversing the fewest possible network links).

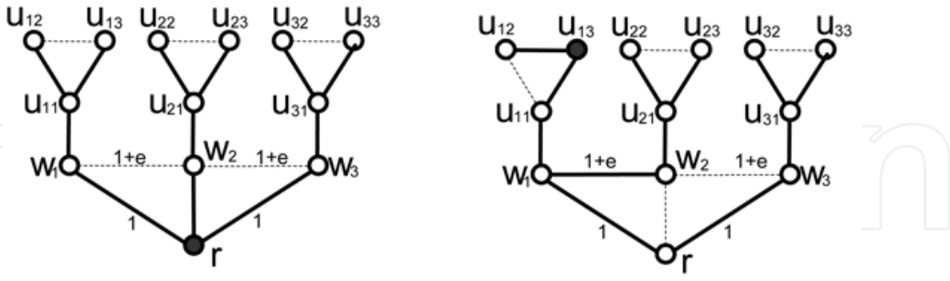


Fig. 4. The RTs of nodes r and u_{13}.

Theorem 4 *Given a set of stations S, the problem of computing the optimal probe assignment is NP-hard.*

Proof: We show a reduction from the vertex cover (VC) problem [24], which is as follows: Given k and a graph $\hat{G} = (\hat{V}, \hat{E})$, does there exist a subset $V' \subseteq \hat{V}$ containing at most k vertices such that each edge in $\hat{E}$ is incident on some node in V'. For a graph $\hat{G}$, we define an instance of the probe assignment problem, and show that there is a vertex cover of size at most k for $\hat{G}$ if and only if there is a feasible probe assignment $\mathcal{A}$ with cost no more than

$COST_{\mathcal{A}} = 5 \cdot |\hat{V}| + |\hat{E}| + k$. We assume that all $c_{s,u} = 1$ (thus, $COST_{\mathcal{A}}$ is the number of probes in $\mathcal{A}$).

For a graph $\hat{G}$, we construct the network graph $G(V,E)$ and a set of stations S for the probe assignment problem as follows. In addition to a root node r, the graph G contains, for each node $\hat{v}_i$ in $\hat{G}$, four nodes denoted by w_i, u_{i1}, u_{i2} and u_{i3}. These nodes are connected with the following edges (w_i, r), (w_i,u_{i1}), (u_{i1}, u_{i2}), (u_{i1}, u_{i3}) and (u_{i2}, u_{i3}). Also, for edge $(\hat{v}_i,\hat{v}_j)$ in $\hat{G}$, we add the edge (w_i,w_j) to G. For instance, the graph G for $\hat{G}$ containing nodes $\hat{v}_1$, $\hat{v}_2$ and $\hat{v}_3$, and edges $(\hat{v}_1,\hat{v}_2)$ and $(\hat{v}_2,\hat{v}_3)$ is shown in Figure 4. The weight of each edge (w_i,w_j) in G is set to $1 + \epsilon$, while the remaining edges have a weight of 1. Finally, we assume that there are monitoring stations at node r and nodes u_{i3} for each vertex $\hat{v}_i \in \hat{G}$. Figure 4 illustrates the RTs of nodes r and u_{13}. Note that edge (w_i,w_j) is only contained in the RTs of u_{i3} and u_{j3}, and (u_{i1}, u_{i2}) is not contained in the RT of u_{i3}.

We first show that if there exists a vertex cover V' of size at most k for $\hat{G}$, then there exists a feasible assignment $\mathcal{A}$ containing no more than $5 \cdot |\hat{V}| + |\hat{E}| + k$ probes. For measuring the latency of the five edges corresponding to $\hat{v}_i \in \hat{V}$, $\mathcal{A}$ contains five probe messages: (r,w_i), (r, u_{i1}), (r, u_{i2}), (u_{i3}, u_{i1}) and (u_{i3}, u_{i2}). So (w_i,w_j) (corresponding to edges $(\hat{v}_i,\hat{v}_j)$ in $\hat{E}$) are the only edges in G whose latency still remains to be measured. Since V' is a vertex cover of $\hat{G}$, it must contain one of $\hat{v}_i$ or $\hat{v}_j$. Suppose $\hat{v}_i \in V'$. Then, $\mathcal{A}$ contains the following two probes (u_{i3},w_i) and (u_{i3},w_j) for each edge (w_i,w_j). Since the probe message (u_{i3},w_i) is common to the measurement of all edges (w_i,w_j) corresponding to edges covered by $\hat{v}_i \in V'$ in $\hat{G}$, and size of V' is at most k, $\mathcal{A}$ contains at most $5 \cdot |\hat{V}| + |\hat{E}| + k$ probes.

We next show that if there exists a feasible probe assignment $\mathcal{A}$ containing at most $5 \cdot |\hat{V}| + |\hat{E}| + k$ probes, then there exists a vertex cover of size at most k for $\hat{G}$. Let V' consist of all nodes $\hat{v}_i$ such that $\mathcal{A}$ contains the probe (u_{i3},w_i). Since each edge (w_i,w_j) is in the RT of only u_{i3} or u_{j3}, $\mathcal{A}$ must contain one of (u_{i3},w_i) or (u_{j3},w_j), and thus V' must be a vertex cover of $\hat{G}$. Further, we can show that V' contains at most k nodes. Suppose that this is not the case and V' contains more than k nodes. Then, $\mathcal{A}$ must contain greater than k probes (u_{i3},w_i) for $\hat{v}_i \in \hat{V}$. Further, in order to measure the latencies of all edges in E, $\mathcal{A}$ must contain $5 \cdot |\hat{V}| + |\hat{E}|$ additional probes. Of these, $|\hat{E}|$ are needed for edges (w_i,w_j), $3 \cdot |\hat{V}|$ for edges (u_{i3}, u_{i1}), (u_{i3}, u_{i2}) and (r,w_i), and $2 \cdot |\hat{V}|$ for edges (u_{i1}, u_{i2}). $\mathcal{A}$ contains 2 probe messages for each edge (u_{i1}, u_{i2}) because the edge does not belong to the RT of u_{i3} and thus 2 probe messages (v, u_{i2}) and (v, u_{i1}), $v \neq u_{i3}$ are needed to measure the latency of edge (u_{i1}, u_{i2}). This, however, leads to a contradiction since $\mathcal{A}$ would contain more than $5 \cdot |\hat{V}| + |\hat{E}| + k$ probes. Thus V' must be a vertex cover of size no greater than k. □

5.2.2 Probe assignment algorithms

We first describe a *simple probe assignment* algorithm that computes an assignment $\mathcal{A}$ whose cost is within a factor of 2 of the optimal. Consider a set of monitoring stations S and for every edge $e \in E$, let $S_e = \{s \mid s \in S \wedge e \in T_s\}$ be the set of stations that can monitor e. For each $e = (u, v) \in E$, select the station $s_e \in S_e$ for which the cost $c_{s_e,u} + c_{s_e,v}$ is minimum. Then add the pairs (s_e, u) and (s_e, v) to $\mathcal{A}$. As a result, the returned assignment is,

$$\mathcal{A} = \{(s_e, u), (s_e, v) | e = (u, v) \in E \wedge s_e \in S_e \wedge s_e = arg \min_{s \in S_e}(c_{s,u} + c_{s,v})\}$$

Theorem 5 *The approximation ratio of the simple probe assignment algorithm is* 2.

Proof: For monitoring the delay of any edge $e \in E$, at least one station $s \in S$ must send two probe messages, one to each endpoint of e. As a result, in any feasible probe assignment at least one probe message can be associated with each edge e. Let it be the message that is sent to the farthest endpoint of e from the monitoring station. Let $\mathcal{A}^*$ be the optimal probe assignment and let s_e^* be the station that monitors edge e in $\mathcal{A}^*$. So, in $\mathcal{A}^*$, the cost of monitoring edge $e = (u, v)$ is at least $\max\{c_{s_e^*,u}, c_{s_e^*,v}\}$. Let s_e be the selected station for monitoring edge e in the assignment $\mathcal{A}$ returned by the simple probe assignment algorithm. s_e minimizes the cost $c_{s,u} + c_{s,v}$, for every $s \in S_e$. Thus, $c_{s_e,u} + c_{s_e,v} \le c_{s_e^*,u} + c_{s_e^*,v} \le 2 \cdot \max\{c_{s_e^*,u}, c_{s_e^*,v}\}$. Thus, $COST_{\mathcal{A}} \le 2 \cdot COST_{\mathcal{A}^*}$. □

Note that the time complexity of the simple probe assignment algorithm can be shown to be $O(|S| \cdot |V|^2)$.

Example 1 This example shows that the simple probe assignment algorithm has a tight approximation ratio of 2. Suppose that the cost of sending any message is 1 and consider the graph depicted in Figure 5. Let the monitoring stations be $S = \{s_1, s_2\}$ and consider the following message assignment, $\mathcal{A}$, that may be calculated by the simple algorithm. The edges (s_1, s_2), (s_1, v_1) and (v_i, v_{i+1}) of every odd i are assigned to station s_1. The edges (s_2, v_1) and (v_i, v_{i+1}) of every even i are assigned to station s_2. In this message assignment both s_1 and s_2 send probe messages to every node v_i and in additional s_1 send probe message to s_2. Hence, $COST_{\mathcal{A}} = 1 + 2 \cdot n$. At the optimal assignment, $\mathcal{A}^*$, all the edges (v_i, v_{i+1}) are assigned to a single station either s_1 or s_2. Here, s_1 sends messages to s_2 and v_1, station s_2 also sends message to v_1, and one message is sent to every node v_i, $i > 1$ either from s_1 or s_2. Hence, $COST_{\mathcal{A}^*} = 2 + n$, and the limit $\lim_{n\to\infty} \frac{COST_{\mathcal{A}}}{COST_{\mathcal{A}^*}} = \lim_{n\to\infty} \frac{1+2\cdot n}{2+n} = 2.$

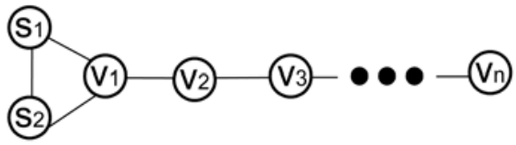


Fig. 5. An example of a probe assignment that cost twice than the optimal.

We turn now to describe a *greedy probe assignment* algorithm that also guarantees a cost within a factor of 2 of the optimal, but yields better results than the simple algorithm in the average case. It is based on the observation that a pair of probe messages is needed for monitoring a link, however, a single message may appear in several such pairs. It attempts to maximize the usage of each message for monitoring the delay of several adjacent links. This is an iterative algorithm that keeps for each station-edge pair (s, e), $e \in T_s$, the current cost, $w_{s,e}$, of monitoring edge e by station s. At each iteration the algorithm select the pair (s', e') with the minimal cost and add the required messages to the message assignment $\mathcal{A}$. If several pairs have the same cost the one with minimal number of hopes between the station and the edge is selected. Probe messages in $\mathcal{A}$ are considered as already been paid and the algorithm update the cost of monitoring the adjacent edges of e' by station s'. This operation is done until all the edges are monitored. A formal description of the algorithm is given in Figure 6, where L is the set of unassigned edges and the initial value of $w_{s,e} \leftarrow c_{s,u} + c_{s,v}$, for every $e = (u, v) \in E$ and $s \in S_e$.

$\mathcal{A} \leftarrow \emptyset$
$L \leftarrow E$
For every $s \in S$ and for every $e = (u, v) \in T_s$ do
    $w_{s,e} \leftarrow c_{s,u} + c_{s,v}$
While $L \neq \emptyset$ do
    Let (s', e') be the pair with minimal $w_{s,e}$ value
    $\mathcal{A} \leftarrow \mathcal{A} \bigcup \{(s', u'), (s', v') | e' = (u', v')\}$
    $L \leftarrow L - \{e'\}$
    For every $\tilde{e} = (x, y) \in T_{s'} \ \wedge y \in \{u', v'\}$ do
        $w_{s',\tilde{e}} \leftarrow c_{s',x}$
    End For
End While
Return $\mathcal{A}$

Fig. 6. The Greedy Probe Assignment Algorithm.

Recall that the algorithm assigns links to the monitoring stations from near to far. First it assigns to each station its adjacent links. Then it continues by assigning links, which are adjacent to the already assigned links. In this way it attempts to avoid the situation where two adjacent links, that should be assigned to the same station, eventually are assigned to two different monitoring stations. The greedy algorithm yields the optimal probe assignment for the graph in Example 1.

Theorem 6 *The approximation ratio of the greedy probe assignment algorithm is* 2.

Proof: Each link $e = (u, v) \in E$ is monitored by the station that minimize the cost $w_{s,e}$. This cost is at most $\min_{s \in S_e}(c_{s,u} + c_{s,v})$. As we have shown in Theorem 5 this guarantees a solution with in a factor of 2 from the optimal. □

6. Path monitoring algorithms

In this section, we address the problem of designing an accurate *path monitoring* system that guarantees that every routing path is monitored by a single monitoring station. First, we present the need for path monitoring and then we provide greedy algorithms for station selection and probe assignment.

6.1 The need for path monitoring

A delay-monitoring system should be able to provide accurate estimates of the end-to-end delay of the routing paths between arbitrary nodes in the network. In the monitoring framework described in the previous section, each link is associated with a single monitoring station that monitors its delay. Thus, the end-to-end delay of any path can be estimated by accumulating the delays of all the links along the path. A drawback of this approach is that the accuracy of a path delay estimation decreases as the number of links that compose the path increases. A better estimate can be achieved by partitioning each path into a few contiguous segments. Each segment is then required to be in the RT of a single monitoring station, which estimates its delay by sending two probe messages to the segment's end-points. Of course, the best estimate of delay is obtained when every path consists of a single segment. Unfortunately, the link monitoring scheme presented in Section

5.1 cannot guarantee an upper bound on the number of segments in a path. In fact, this number may be as high as the number of links in the path, even when the number of monitoring stations is small, as illustrated by the following example.

Example 2 Consider a graph that consists of a grid of size $k \times k$ and two additional nodes, a and b, as depicted in Figure 7-(a). The weight of each grid edge is 1 except for edges along the main diagonal from node c to d whose weights are 1-ϵ. Also, the weights of edges incident on nodes a and b vary between 1* and k* as shown in Figure 7-(a), where $n^* = n \cdot (1 - \epsilon)$. Monitoring stations are located at nodes a and b, and their RTs are along their SPTs, as shown in Figures 7-(b) and 7-(c), respectively. In this graph, the shortest path from node c to d is composed of the edges along the main diagonal of the grid, as shown in Figure 7-(d). Note that any pair of adjacent edges along this path are monitored by two different stations. Thus, each edge in this path represents a separate segment and the number of segments that cover this path is $2 \cdot (k\text{-}1)$, even though the number of stations is only two. □

In this section, we address the problem of designing an accurate *path monitoring* system that guarantees that every routing path is monitored by a single station. Thus, for every path $P_{u,v}$ there is a monitoring station $s \in S$ such that $P_{u,v} \in T_s$. In such case, the end-to-end delay of the path can be estimated by sending at most three probe messages, as described later in Sub-Section 6.3.

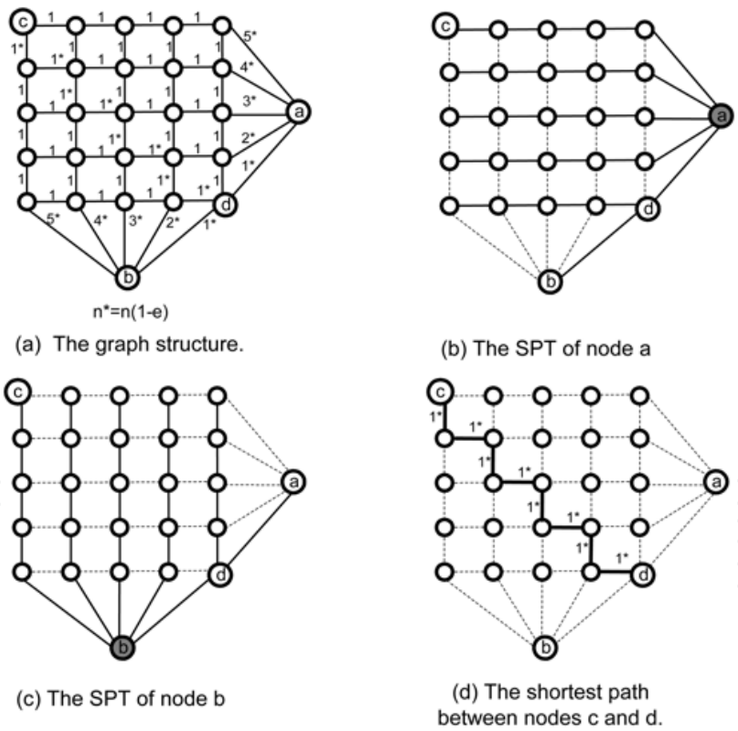


Fig. 7. An example where each edge along a given path is included in a separate segment.

6.2 An efficient station selection algorithm

The station selection problem for path monitoring is defined as follows.

Definition 3 (The Weighted Path Monitoring Problem - WPM): Given a graph $G(V,E)$, with a weight w_v and a RT T_v for every node $v \in V$, and a routing path $P_{u,v}$ between any pair of

nodes $u, v \in V$, find the set $S \subseteq V$ that minimizes the sum $\Sigma_{v \in S} w_v$ such that for every pair $u, v \in V$ there is a station $s \in S$ such that $P_{u,v} \subseteq T_s$. □

In the un-weighted version of the WPM problem, termed the *path monitoring* (PM) problem, the weight of every node is 1.

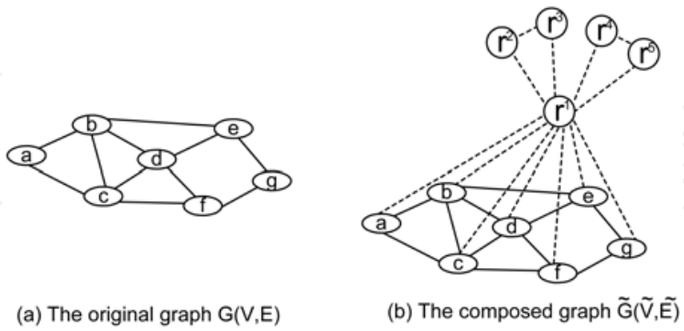


Fig. 8. An example of a graph $G(V,E)$ and the corresponding graph $\tilde{G}$ ($\tilde{V}$, $\tilde{E}$).

Theorem 7 The PM and WPM problems are both NP-Hard.

Proof: We show that the PM and WPM problems are NP-hard by presenting a polynomial reduction from the *vertex cover* (VC) problem[4] to the PM problem. Since the VC problem is a well-known NP-complete problem this proves that the PM and the WPM problems are also NP-hard.

Consider the following reduction from the VC problem to the PM problem. For a given graph $G(V,E)$ we construct a graph $\tilde{G}$ ($\tilde{V}$, $\tilde{E}$) that contains the following nodes and edges. $\tilde{V} = V \bigcup \{r_1, r_2, r_3, r_4, r_5\}$ and the edges $\tilde{E} = E \bigcup \{(v, r_1) \mid v \in V\} \bigcup \{(r_1, r_2), (r_1, r_3), (r_1, r_4), (r_1, r_5), (r_2, r_3), (r_4, r_5)\}$. The weight of every edge $e \in E$ is 3 and the weight of any edge $e \notin E$ is 2. In the following $R = \{r_1, r_2, r_3, r_4, r_5\}$. An example of such graph is given in Figure 8.

Now we will show that the given VC instance, graph $G(V,E)$, has a solution, S of size k if and only if the PM instance, graph $\tilde{G}$ ($\tilde{V}$, $\tilde{E}$) has a solution, $\tilde{S}$ of size $k + 2$. In this proof we assume without lose of generality that the routing tree (RT) of every node is its shortest path tree (SPT). First, let considered the auxiliary structure defined by the nodes in R. The edge (r^2, r^3) is covered only by the SPTs T_{r^2} and T_{r^3}. Therefore, one of these nodes must be included in $\tilde{S}$. Similarly, one of the nodes r^4 or r^5 must be included in $\tilde{S}$ for covering the edge (r^4, r^5). Suppose without lose of generality that the selected nodes are r^2 and r^4.

Let us turn to describe the different SPTs of the nodes in $\tilde{G}$ ($\tilde{V}$, $\tilde{E}$). The SPTs T_{r^2} and T_{r^4} are very similar. The SPT T_{r^2} contains the edge (r^2, r^3) and all the incident edges of node r^1 except edge (r^1, r^3). The SPT T_{r^4} contains the edge (r^4, r^5) and all the incident edges of node r^1 except edge (r^1, r^5). These two SPTs together guarantee that any shortest path that one of its end-node is in the set R is covered. They also cover the shortest path between every pair of nodes $u, v \in V$ such that $(u, v) \notin E$. The only shortest paths that are not covered by the two SPTs T_{r^2} and T_{r^4} are the one-edge paths defined by E. Let N_v be the set of adjacent nodes to node v in the graph $\tilde{G}$ ($\tilde{V}$, $\tilde{E}$). The SPT T_v of every node $v \in V$ contains of the set of edges, $T_v = \{(v, u) \mid u \in N_v\} \bigcup \{(r^1, u) \mid u \notin \tilde{V} - N_v\}$.

[4] Definition of the vertex cover problem is given in the proof of Theorem 4.

Consider a solution S of size k to the VC problem defined by graph $G(V,E)$. Then $\tilde{S} = S \bigcup \{r^2, r^4\}$ is a solution of size k+2 to the corresponding PM instance $\tilde{G}(\tilde{V}, \tilde{E})$. At least one end-point of every edge $e \in E$ is a node in S. Therefore, $\bigcup_{s \in S} T_s$ covers all the paths with only one edge between any pairs of nodes in V. The rest of the paths are covered by the SPTs T_{r^2} and T_{r^4}. Hence, $\tilde{S}$ is a solution of size k + 2 to the PM problem.

Let $\tilde{S}$ be a solution of size k + 2 to the PM problem defined by the graph $\tilde{G}(\tilde{V}, \tilde{E})$. Then $S = \tilde{S} \cap V$ is a solution of size at most k to the VC instance $G(V,E)$. The set $\tilde{S}$ must include at least two nodes from the set R. Thus, $|S| \leq k$. The SPTs of the nodes in R do not contain any edge in E. Therefore the edges of E are covered only by SPTs of nodes in S. Since for every node $v \in V$ holds that $T_v \cap E = \{(v, u) \mid u \in N_v - \{r^1\}$, The set S is a solution to the instance $G(V,E)$ of the given VC problem. □

6.2.1 Lower bounds for the PM and WPM problems

We now turn our attention to computing the lower bounds on the best approximations for the PM and WPM problems. In the sequel, we limit our discussion to cases where monitoring stations are selected from a given subset of nodes $Y \subseteq V$. In our lower bound proof, we use a polynomial reduction, $\hat{\mathcal{R}}(I)$, from any instance $I(Z,\mathcal{Q})$ of the SC problem to a corresponding PM instance. The graph $\hat{G}_{\hat{\mathcal{R}}(I)}(V,E)$ computed by the reduction $\hat{\mathcal{R}}(I)$ contains the following nodes. The nodes u_i and s_j for every element $z_i \in Z$ and set $Q_j \in \mathcal{Q}$, respectively, and three additional nodes u_0, t and r. The node u_0 corresponds to a dummy element z_0 that is included in every set $Q_j \in \mathcal{Q}$, and each one of the nodes t and r is the hub of a star that is connected to the rest of the nodes. The weight of all the graph edges is 1. An example of such a graph $\hat{G}_{\hat{\mathcal{R}}(I)}(V,E)$ is depicted in Figure 9-(a), for the SC instance with four elements $\{z_1, z_2, z_3, z_4\}$ and three sets $Q_1 = \{z_1, z_2\}$, $Q_2 = \{z_2, z_4\}$, $Q_3 = \{z_3, z_4\}$.

We next describe the routing paths between every pair of nodes, which are along their shortest paths. The nodes t and r have a direct edge to every other node. The shortest path between every pair of nodes s_j and s_k is through node t, and between every pair u_i and u_l, it is through node r. Between every pair of nodes s_j and u_i for a set $Q_j \in \mathcal{Q}$ and an element $z_i \in Z$, the shortest path traverses through node t if $z_i \in Q_j$, otherwise it passes through node r. The RTs of the various nodes for of the given example above are depicted in Figure 9. Moreover, for the proof, let assume that the set of possible monitoring stations is $Y = \{s_j \mid \forall Q_j \in \mathcal{Q}\} \bigcup \{r\}$.

Lemma 1 *Consider an instance $I(Z,\mathcal{Q})$ of the SC problem and the corresponding graph $\hat{G}_{\hat{\mathcal{R}}(I)}(V,E)$ and set Y. Then there is a solution of size k to the SC problem if and only if there is a solution of size k + 1 to the corresponding PM problem defined by $\hat{G}_{\hat{\mathcal{R}}(I)}(V,E)$ and Y.*

Proof: Let $\mathcal{S}$ be a solution of size k to the given SC instance. Let set $\hat{S} = \{s_j \mid Q_j \in \mathcal{S}\} \bigcup \{r\}$. We claim that $\hat{S}$ is a feasible solution of size k + 1 to the given PM problem. First, observe that $\hat{S} \subseteq Y$. Further, the RT T_r covers all the shortest paths that pass through node r. Also, the RT of any node $s_j \in \hat{S}$ for a set $Q_j \in \mathcal{Q}$ covers all the shortest paths between arbitrary pairs of nodes s_j and s_k. Thus, we only need to show that all the shortest paths between pairs of nodes s_k and u_i that pass through node t are also covered. This is satisfied since for every u_i, there is a $Q_j \in \mathcal{S}$ such that $z_i \in Q_j$. Thus, $s_j \in \hat{S}$ and T_{s_j} contains all such paths between u_i and s_k through t.

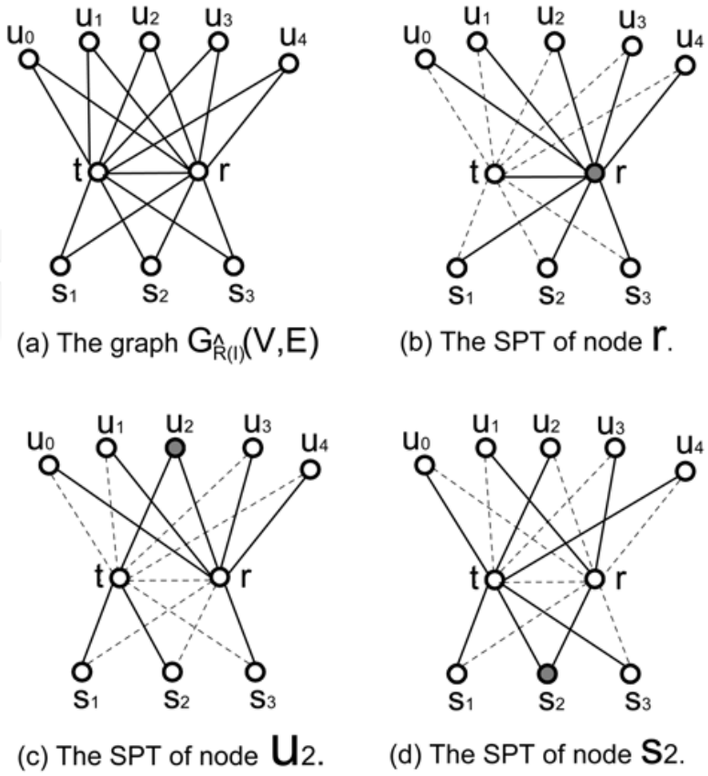


Fig. 9. The graph $G_{\hat{\mathcal{R}}}$ (V,E) and the RTs of the nodes.

Now consider a solution $\hat{S}$ of size $k + 1$ to the PM problem and let $\mathcal{S} = \{Q_j \mid s_j \in \hat{S}\}$. We claim that $\mathcal{S}$ is a solution of size k to the given SC problem. Note that $r \in \hat{S}$ for covering all the shortest paths between node u_0 and any node u_i. This is because element z_0 is contained in all sets of $\mathcal{Q}$ and thus edge (u_0, r) is not contained in any RT T_{s_j}. Hence, $\mathcal{S}$ is of size k. The set $\hat{S} - \{r\}$ covers all the shortest paths that pass through node t. Every element $z_i \in Z$ appears in at least one set $Q_k \in \mathcal{Q}$. Thus, there is a shortest path through t between every node u_i and at least one node s_k. This path is covered by at least one node $s_j \in \hat{S}$ for whom the shortest path to u_i also passes through t. As a result, for every element $z_i \in Z$ there is a set $Q_j \in \mathcal{S}$ such that $z_i \in Q_j$. □

Theorem 8 *The lower bound of any approximation algorithm for the PM and WPM problems is* $\ln(|V|)$.

Proof: Let J be a bad SC instance with m elements and $n \simeq c_1 \cdot (\ln\ m)^{c_2}$ subsets, as constructed in [26] for proving the $\ln(m)$ lower bound for the SC problem. Let $\hat{G}_{\hat{\mathcal{R}}(J)}$ be the graph calculated by $\hat{\mathcal{R}}$. The lower bound for the PM problem, $\delta_{PM}(|V|)$, satisfies,

$$\delta_{PM}(|V|) \geq \frac{COST_{PM}(\hat{G}_{\hat{\mathcal{R}}(J)})}{OPT_{PM}(\hat{G}_{\hat{\mathcal{R}}(J)})} \simeq \frac{\ln(m) \cdot OPT_{SC}(J) + 1}{OPT_{SC}(J) + 1} \simeq \ln(m)$$

The number of nodes in the graph $\hat{G}_{\hat{\mathcal{R}}(J)}$ is $|V| = 3+m+n \simeq m+c_1 \cdot (\ln\ m)^{c_2}$. Consequently, for a large m we assume that $|V| \simeq m$ and thus, $\delta_{PM}(|V|) \geq \ln(|V|)$. □

6.2.2 A greedy algorithm for station selection

Similar to the WLM problem, an efficient solution to a WPM instance is obtained by mapping it to an instance of the CS problem and using the greedy heuristic given in Figure 3 to solve this problem. Consider a graph $G(V,E)$, a weight w_v and an RT T_v for every node $v \in V$ and let $P_{u,v}$ be the routing path between any nodes $u, v \in V$. The corresponding SC instance in as follows. Every shortest path $P_{u,v}$ is represented by an element, denoted by $[u, v]$. Thus, the universe of elements, Z, contains $\binom{|V|}{2} = \frac{|V|\cdot(|V|-1)}{2}$ elements. For every node $v \in V$ we define a set Q_v with weight w_v that contains all the routing paths covered by the RT of node v, *i.e.*, $Q_v = \{[x, y] \mid x, y \in V, x \notin y, P_{x,y} \subseteq T_v\}$. Now consider a feasible solution $S = \{Q_{v_1}, Q_{v_2}, \cdots, Q_{v_k}\}$ to the defined SC problem. Then, $S = \{v \mid Q_v \in S\}$ defines a feasible solution to the WPM problem and for every path $P_{u,v}$, $u, v \in V$ there is a monitoring station $s \in S$ such that $P_{u,v} \subseteq T_s$. As a result, an efficient solution to the CS problem defines also an efficient solution of the WPM problem.

Theorem 9 *The greedy algorithm computes a* $2 \cdot \ln(|V|)$*-approximation for the PM and WPM problems.*

Proof: Similar to the proof of Theorem 3. □

6.3 An efficient probe assignment algorithm

Suppose that the greedy algorithm selects a set of monitoring stations S. A monitoring station $s \in S$ can monitor any path $P_{u,v} \subseteq T_s$ by sending at most three probe messages to nodes u, v and w, where $w \in P_{u,v}$ is the common ancestor of nodes u, v in the tree T_s. Let $delay(y, x)$ be the estimated delay of the path between nodes y and x. Since, s can estimate its delay to the nodes u, v and w, the delay of the path $P_{u,v}$ can be computed as follows:

$$delay(u, v) = delay(s, u) + delay(s, v) - 2 \cdot delay(s, w)$$

Theorem 10 *Given a set of stations S, the problem of computing the optimal probe assignment is NP-hard.*

Proof: We show a similar reduction to the one given in the proof of Theorem 4 from the vertex cover (VC) problem. For a given graph $\hat{G}$ $((\hat{V}), \hat{E})$, we define an instance of the probe assignment problem and show that there exists a vertex cover of size at most k for $\hat{G}$ if and only if there exists a feasible probe assignment $\mathcal{A}$ with cost no more than $COST_{\mathcal{A}} = 2 \cdot |\hat{V}| + 2 \cdot |\hat{E}| + k + 1$. We assume that all $c_{s,u} = 1$ (thus, $COST_{\mathcal{A}}$ is the number of probes in $\mathcal{A}$).

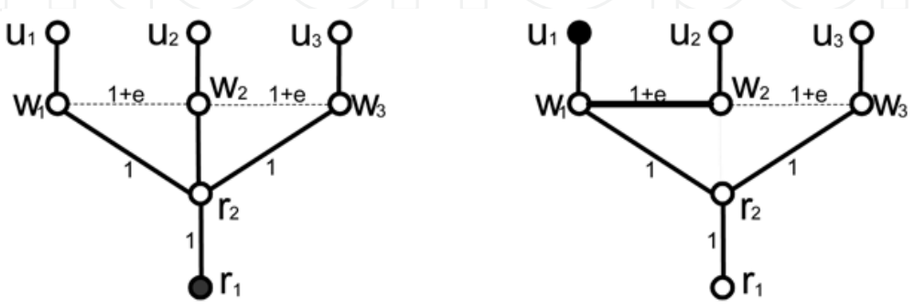


Fig. 10. The RTs of nodes r and u_{13}.

For a graph $\hat{G}$, we construct the network graph $G(V,E)$ and set of stations S for the probe assignment problem as follows. The graph G contains two root nodes, denoted by r_1, r_2, and for each node $\hat{v}_i$ in $\hat{G}$ it contains two additional nodes denoted by w_i and u_i. The set E of edges of the graph G consists of the following edges. The edge (r_1, r_2) and for each node $\hat{v}_i$ in G, the edges (r_2,w_i) and (w_i, u_i). Also, for every edge ($\hat{v}_i$,$\hat{v}_j$) in $\hat{G}$, we add the edge (w_i,w_j) to G. The weight of each edge (w_i,w_j) in G is set to $1 + \epsilon$, while the remaining edges have a weight of 1. Finally, we assume that there are monitoring stations at node r_1 and nodes u_i for each vertex $\hat{v}_i \in \hat{G}$. For example, consider the graph $\hat{G}(\hat{V}, \hat{E})$ that contains nodes $\hat{V} = \{\hat{v}_1, \hat{v}_2, \hat{v}_3\}$ and edges $(\hat{E}) = \{(\hat{v}_1, \hat{v}_2), (\hat{v}_2,\hat{v}_3)\}$. Figure 10 shows the corresponding graph G as well as the routing trees of the nodes r_1 and u_1. Note that edge (w_i,w_j) is only contained in the RTs of u_i and u_j.

We first show that if there exists a vertex cover V' of size at most k for $\hat{G}$, then there exists a feasible assignment $\mathcal{A}$ containing no more than $2 \cdot \mid \hat{V} \mid + 2 \cdot \mid \hat{E} \mid + k + 1$ probes. Recall that by sending a single probe from r_1 to every other node We can calculate the path delay of every path that traverses through node r_2 and every sub-paths of these paths. This requires $2 \cdot \mid \hat{V} \mid +1$ probes assign to r_1. Thus the only paths that are not monitored by r_2 are the ones that contain an edge (w_i,w_j), corresponding to edges ($\hat{v}_i$,$\hat{v}_j$) in $\hat{E}$. These include the paths P_{u_i,u_j}= {u_i,w_i,,w_j , u_j}, P_{w_i,u_j}= {w_i,w_j , u_j}, P_{u_i,w_j}= {u_i,w_i,w_j} and P_{w_i,w_j}= {w_i,w_j}. Consider such path P_{u_i,u_j} = {u_i,w_i,w_j , u_j}. This path can be monitored only by u_i or u_j . Let assume without the lose of generality that it is monitored by u_i. This is done by sending a single probe from u_i to u_j . Similarly the path P_{u_i,w_j}= {u_i,w_i,w_j} can be monitored by sending a single probe from u_i to w_j . From this follows that $2 \cdot \mid \hat{E} \mid$ are required for each edge in $\hat{E}$. Yet, we still need to monitor the paths P_{w_i,u_j}= {w_i,w_j , u_j} and P_{w_i,w_j}= {w_i,w_j}. This can be done by sending a single message from u_i to w_i. Recall that this probe message can be used for path monitoring of every path P_{w_i,u_j} and P_{w_i,w_j}such that ($\hat{v}_i$,$\hat{v}_j$) in $\hat{E}$. Since V' is a vertex cover of $\hat{G}$, it must contain one of $\hat{v}_i$ or $\hat{v}_j$. Let assume the node $\hat{v}_i$. So by selecting node u_i as the monitoring station of the path P_{u_i,u_j} and the corresponding sub-paths that contains the edge (w_i,w_j), only $2 \cdot \mid \hat{E} \mid + k$ additional probes are required.

We next show that if there exists a feasible probe assignment $\mathcal{A}$ containing at most $2 \cdot \mid \hat{V} \mid + 2 \cdot \mid \hat{E} \mid + k + 1$ probes, then there exists a vertex cover of size at most k for $\hat{G}$. As mentioned above, at least $2 \cdot \mid \hat{V} \mid + 1$ probes are required to monitors all the paths that traverse through node r_2 and any sub path of them. Now, let V' consists of all nodes $\hat{v}_i$ such that $\mathcal{A}$ contains the probe (u_i,w_i). Since each edge (w_i,w_j) is in the RT of only u_i or u_j, $\mathcal{A}$ must contain one of (u_i,w_i) or (u_j ,w_j), and thus V' must be a vertex cover of $\hat{G}$. Further, we can show that V' contains at most k nodes. Suppose that this is not the case and V' contains more than k nodes. Then, $\mathcal{A}$ must contain more than k probes (u_i,w_i) for $\hat{v}_i \in \hat{V}$. However, as mentioned above at least $2 \cdot \mid \hat{E} \mid$ probes are required to measure any path $P_{u_i,u_j}$and one of the paths P_{w_i,u_j} or P_{u_i,w_j}. This contradict the assumption that there are $2 \cdot \mid \hat{V} \mid + 2 \cdot \mid \hat{E} \mid + k + 1$ probes. □

Finding a low-cost optimal probe assignment for monitoring path delays is more challenging than computing the probe assignment for link monitoring (see Section 5.2).

Unlike the link monitoring case, we cannot claim that the optimal solution for path monitoring must contain at least one probe for every routing path[5], which makes the problem for paths more dificult to approximate. We believe that a greedy algorithm similar to the one described in Section 5.2 will achieves good results. Yet, finding an algorithm with a low approximation ratio is still an open problem.

7. Experimental results

In this section, we present experimental results for the number of monitoring stations and probe messages required to measure the latency of every link in the network. The experiments are based on network topologies generated using the Waxman Model [27], which is a popular model used by the networking research community (e.g., [15]). Different network topologies are generated by varying three parameters: (1) n, the number of nodes in the network graph; (2) α, a parameter that controls the density of short edges in the network; and (3) β, a parameter that controls the average node degree.

For computing the locations of monitoring stations that cover all the links in the network graph, we use the greedy algorithm described in Section 5.1.2. These monitoring stations are then used to compute probe messages for measuring the latency of every network link using the greedy algorithm from Sub-Section 5.2.2. We consider two measures for the cost of a probe: $c_{s,v} = 1$ and $c_{s,v}$ is the number of links in the shortest path between s and v. The former optimizes the number of probe messages generated, while the latter optimizes total number of links traversed by all the probe messages.

Num Edges	Num Monitors	Num Probes $c_{sv} = 1$	Num Links c_{sv} = hop count
2189	10	2277	7936
5540	24	5651	14104
8947	41	9150	19649
11178	55	11363	22860
16566	93	16699	30755

Table 1. Number of Monitoring Stations/Probes, $n = 1000$, $\alpha = 0.2$, $\beta \in \{0.02, 0.05, 0.08, 0.1, 0.15\}$.

Table 1 depicts the results of our experiments for networks containing 1000 nodes. We vary the number of edges in the graph by increasing the β parameter and leaving α fixed. From the tables, it follows that as the number of edges in the graph increases, we need more monitoring stations to cover all the links in the network. However, even for large networks, a few monitoring stations suffice for monitoring all the links in the network. For instance, for n = 1000 and 2200 edges, only 10 monitoring stations cover all the network links.

[5] We only know that it must contain one probe for every edge.

In terms of probe messages, the number of probe messages generated by the greedy algorithm closely tracks the number of edges, which is good since this implies a high degree of sharing among the probe messages. Note that this is almost optimal since the number of probe messages needed to measure all the network links is at least the number of edges in the graph. Finally, observe that the average number of links traversed by each probe message is fairly low, ranging between 2 and 4 in most cases.

8. Summary

This chapter introduces a greedy approach for delay monitoring of IP networks. It proposed two-phased monitoring scheme that ensures complete coverage of the network from both links and paths point of views, and it minimizes the monitoring overhead on the underlying production network. In the first phase it computes the locations of monitoring stations such that all network links or paths are covered by the minimal number of stations. Subsequently, in the second phase, it computes the minimal set of probe messages to be sent by each station such that the latency of every routing path can be measured. Unfortunately, both the station selection and the probe assignment problems are NP-hard. However, by using greedy approximation algorithms the scheme finds solutions close to the best possible approximations to both the station selection and the probe assignment problems. Further, the experimental results demonstrate the effectiveness of the presented algorithms for accurately monitoring large networks with very few monitoring stations and probe messages close to the number of network links.

9. References

[1] William Stallings. “SNMP, SNMPv2, SNMPv3, and RMON 1 and 2”. *Addison-Wesley Longman* Inc., 1999. (Third Edition).

[2] “NetFlow Services and Applications”. Cisco Systems White Paper, 1999.

[3] S. R. Stevens, “TCP/IP illustrated”, *Addison-Wesley Publishing Company*, 1994.

[4] Cooperative Association for Internet Data Analysis (CAIDA). http://www.caida.org/.

[5] V. Jacobsen. “Pathchar - A Tool to Infer Characteristics of Internet Paths”, April 1997. ftp://ftp.ee.lbl.gov/pathchar.

[6] P. Beyssac, “Bing - Bandwidth Ping”, URL: http://www.freenix.fr/freenix/logiciels/bing.html.

[7] R. L. Carter and M. E. Crovella. “Server Selection Using Dynamic Path Characterization in Wide-Area Networks”, In *Proceedings of IEEE INFOCOM'99*, Kobe, Japan, April 1997.

[8] K. Lai and M. Baker. “Measuring Bandwidth”. In *Proceedings of IEEE INFOCOM’99*, New York City, New York, March 1999.

[9] C. Dovrolis, P. Ramanathan and D. Moore. “What Do Packet Dispersion Techniques Measure?”. In *Proceedings of IEEE INFOCOM’2001*, Anchorage, Alaska, April 2001.

[10] Y. Bejerano abd R. Rastogi, "Robust monitoring of link delays and faults in IP networks". In *Proceedings of the IEEE INFOCOM'2003*, San Francisco, CA, USA, April 2003.

[11] V. Chavatel, "A Greedy Heuristic for the Set-Covering Problem", *Math. of Operation Research*, Vol. 4, No. 3, pp 233-235, 1979.

[12] K. Moore, "SONAR - A Network Proximity Service, Version 1". Internet-Draft, http://www.netlib.org/utk/projects/sonar/ August 1998.

[13] P. Francis, S. Jamin, V. Paxson, L. Zhang, D. F. Gryniewicz, and Y. Jin, "An Architecture for a Global Internet Host Distance Estimation Service", In *Proceedings of IEEE INFOCOM'99*, New York City, New York, March 1999.

[14] S. Jamin, C. Jin, Y. Jin, Y. Raz, Y. Shavitt, and L. Zhang, "On the Placement of Internet Instrumentation", In *Proceedings of IEEE INFOCOM'2000*, Tel Aviv, Israel, March 2000.

[15] Y. Breitbart, C-Y. Chan, M. Garofalakis, R. Rastogi and A. Silberschatz, "Efficiently Monitoring Bandwidth and Latency in IP Networks", In *Proceedings of the IEEE INFOCOM'2000*, Tel-Aviv, Israel, March 2000,

[16] M. Brodie, I. Rish and S. Ma, "Intelligent probing: A cost-effective approach to fault diagnosis in computer networks", In *IBM Systems Journal*, Vol. 31, No. 3, pp 372-385, 2002.

[17] F. Li, M. Thottan, End-to-End Service Quality Measurement Using Source-Routed Probes In *Proceedings of the IEEE INFOCOM'2006*, Barcelona, Spain, April 2006.

[18] Y. Shavitt, X. Sun, A. Wool and B. Yener, "Computing the Unmeasured: An Algebraic Approach to Internet Mapping", In *Proceedings of IEEE INFOCOM'2001*, Alaska, April 2001.

[19] M. Adler, T. Bu, R. Sitaraman and D. Towsley, "Tree Layout for Internal Network Characterizations in Multicast Networks", In *Proceedings of NGC'01*, London, UK, November 2001.

[20] Y. Breitbar, F. Dragan and H. Gobjuka, " Effective network monitoring", In *Proceedings of ICCCN'04*, Rosemont, IL, USA, October 2004.

[21] H. X. Nguyen and P. Thiran, "Active Measurement for Multiple Link Failures Diagnosis in IP Networks, In *Proceedings of PAM'04*, Antibes, France, 2004.

[22] K. Suh, Y. Guo, J. F. Kurose, D. F. Towsley, Locating network monitors: Complexity, heuristics, and coverage, In *Proceedings of Infocom'05*, Miami, FL, USA, March, 2004.

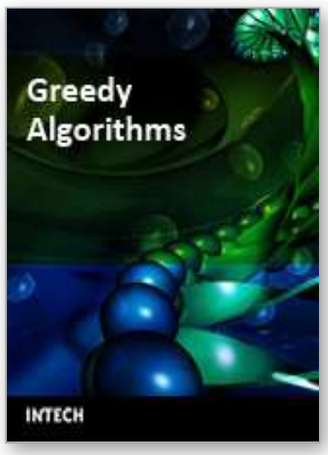
[23] G. R. Cantieni, G. Iannaccone, C. Barakat, C. Diot and P. Thiran, "Reformulating the monitor placement problem: optimal network-wide sampling, In *Proceedings of CoNEXT06*, Lisboa, Portugal, December, 2006.

[24] M. R. Garey and D. S. Johnson. "Computers and Intractability: A Guide to the Theory of NP-Completeness". *W.H. Freeman Publishing Company*, 1979.

[25] Y. Bejerano abd R. Rastogi, "Robust monitoring of link delays and faults in IP networks". IEEE/ACM Trans. on Networking, Vol. 14, No. 5, pp 1092-1103, 2006.

[26] U. Feige, "A threshold of ln *n* for approximating set-cover", *Proceedings of the 28th Annual ACM Symposium on Theory of Computing, 314-318*, 1996.

[27] B. M. Waxman. "Routing of Multipoint Connections". *IEEE Journal on Selected Areas in Communications*, 6(9):1617-1622, December 1988.



Greedy Algorithms

Edited by Witold Bednorz

ISBN 978-953-7619-27-5
Hard cover, 586 pages
Publisher InTech
Published online 01, November, 2008
Published in print edition November, 2008

Each chapter comprises a separate study on some optimization problem giving both an introductory look into the theory the problem comes from and some new developments invented by author(s). Usually some elementary knowledge is assumed, yet all the required facts are quoted mostly in examples, remarks or theorems.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yigal Bejerano and Rajeev Rastogi (2008). A Greedy Scheme for Designing Delay Monitoring Systems of IP Networks, Greedy Algorithms, Witold Bednorz (Ed.), ISBN: 978-953-7619-27-5, InTech, Available from: http://www.intechopen.com/books/greedy_algorithms/a_greedy_scheme_for_designing_delay_monitoring_systems_of_ip_networks

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen