Optimal positioning of active and passive monitoring devices

Claude Chaudet, Eric Fleury, Isabelle Guérin-Lassous

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Optimal positioning of active and passive monitoring devices

Claude Chaudet — Éric Fleury — Isabelle Guérin Lassous

N° 5273 – version 2

version initiale Juillet 2004 – version révisée Février 2005

Thème COM

ISSN 0249-6399 ISRN INRIA/RR--5494--FR+ENG

Optimal positioning of active and passive monitoring devices

Claude Chaudet , Éric Fleury , Isabelle Guérin Lassous

Thème COM — Systèmes communicants
Projet Ares

Rapport de recherche n° 5273 – version 2*— version initiale Juillet 2004 — version révisée Février 2005 19 pages

Abstract: Network measurements are essential for assessing performance issues, identifying and locating problems. Two common strategies are: the passive approach that attaches specific devices to links in order to monitor the traffic that passes by the network and the active approach that generates explicit control packets in the network for measurements. One of the key point is to minimize the overhead in terms of hardware, software and maintenance cost and of additional traffic.

In this paper, we study the problem of assigning tap devices for passive monitoring and beacons for active monitoring. Minimizing the number of devices and finding optimal strategic locations is a key issue, mandatory for deploying scalable monitoring platforms. We propose new solutions based on an Integer Linear Programming formulation.

Key-words: Network measurements, Passive/Active monitoring, Optimization, Integer Linear Programming.

* Modification of the NP-completeness proof

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe, 38334 Montbonnot Saint Ismier (France)
Téléphone : +33 4 76 61 52 00 —Télécopie +33 4 76 61 52 52

Positionnement optimal de sondes actives et passives dans un réseau

Résumé : La surveillance des réseaux est un outil essentiel lorsqu'il s'agit d'optimiser les performances ou de localiser et identifier des problèmes. Les deux approches permettant de réaliser cette tâche sont l'approche passive, attachant des équipements dédiés à des liens du réseau, observant le trafic transitant par ces liens et l'approcha active reposant sur l'envoi périodique de messages de contrôle. Dans un cas comme dans l'autre, le problème de la minimisation du coût en terme d'équipements ou de volume de trafic de contrôle introduit dans le réseau se pose.

Dans ce papier, nous étudions les problèmes du positionnement de sondes pour la surveillance passive et de la sélection de nœuds particuliers pour l'approche active. La minimisation du nombre d'équipements mis en jeu est un problème clé lorsqu'il s'agit de concevoir des plate-formes de mesure capables de passer à l'échelle. Nous proposons une approche basée sur la programmation linéaire en nombres entiers afin de déterminer le positionnement optimale de ces sondes.

Mots-clés : Métrologie de réseaux, surveillance active et passive, optimisation, programmation linéaire en nombres entiers

1 Introduction

The number of users of the Internet is growing fast, as well as the amount of traffic conveyed and the complexity of the network topology. Consequently, the Internet backbones are also growing rapidly, taking advantage of every new speed enhancing technology in order to provide the bandwidth required by new applications. An Internet Service Provider (ISP) network is composed of multiple points of presence or POPs, as shown on Figure 1. Such POPs are sophisticated engineering systems and their expansion yields to complex and irregular topologies. If the growth of the amount of traffic is a key issue in designing POPs architectures, the nature of the traffic is also evolving introducing strong constraints on the network performance. Indeed, the global network performance is becoming more and more critical since many e-business applications rely on the high availability of the network resources. This creates a high level of competition between ISPs, each seeking to accurately measure its POPs performances in order to be able to correctly negotiate service level agreements (SLAs) with customers.

A service level agreement can specify several performance parameters. The ISP shall guarantee that all parameters levels are in concordance to the negotiated values and report any deviation from the initial rules. To fulfill this objective, ISPs have to deploy and maintain specific tools and devices to monitor the network. Analyzing network traffic patterns is essential for managing these complex systems and ISPs have to monitor their POPs status and the traffic they convey, for example to perform provisioning. Provisioning usually requires detailed information on the network capacity and traffic patterns and therefore needs detailed analysis of links usage over time. A constant monitoring is also required to enforce and ensure both connectivity and security of the infrastructure. Permanent monitoring is useful for example to detect unusual traffic amount or patterns resulting from unauthorized activities.

Many strategies are available to monitor networks. The two most common are the *passive* approach and the *active* approach. These strategies are different and have their own interest and therefore should be considered as complementary as they can be used together. The passive approach uses specific devices, generally attached to a link in order to monitor the traffic that goes through it. It does not introduce any additional traffic on the network and measures real flows characteristics. However, the deployment and installation of expensive devices on specific and strategic edges of the network is needed in order to maximize the amount of traffic monitored while minimizing hardware, software and maintenance costs.

The active approach provides explicit control by generating packets for measurement from various measurement points. Each of these measurement points (also called *beacon* [15]) sends IP packets (called *probes* [15]) to other nodes in the network. This set of messages is then used to detect link failures and to infer link delays. Using active monitoring to diagnose faults introduces traffic overhead by constantly sending probes through the networks. Active monitoring also requires the setup and maintenance of hardware or software devices representing an additional cost.

In this work, we are interested in minimizing the infrastructure cost of both passive and active monitoring. For passive monitoring, it may not be useful for an ISP to monitor

every traffic going through its POP. Indeed, capturing 90 % of the traffic may be enough to detect malicious traffic patterns [13], or to keep track of the values of two important variables associated with TCP connections [9]: the sender's congestion window (cwnd) and the connection round trip time (RTT). We show that the problem of placing passive device in order to monitor at least a given amount of the total traffic is NP-complete. However, we derive optimal solutions by using Integer Linear Programming (ILP) formulation of the problem. This solution is compared with the trivial greedy heuristic. Concerning active monitoring, we use the same strategy to improve the two-phased approach presented in [1] and [15] to optimize both the number of devices and the number of generated messages.

The remaining of this paper is organized as follows. Section 2 presents the global architecture. Section 3 discusses related work. Section 4 contains a brief summary of Integer Linear Programming paradigms. The main discussion begins in Section 5 in which we describe our main contribution on passive devices positioning and show simulation results. In Section 6 we focus on active monitoring for which a similar strategy is used to improve beacons positioning. In Section 7, we summarize the results presented and discuss their implications on current monitoring strategies.

2 General architecture

We present in this section the general architecture of our study. We focus on the POP architecture and topology since it appears that POPs represent the key place where monitoring could be done efficiently. Monitoring traffic in a POP may help to analyze the traffic demand between a pair of POPs [2] or to derive methodology that observes the sender-to-receiver and receiver-to-sender segments in a TCP connection, and infers/tracks the time evolution of the sender's congestion window and the connection round trip time [9].

The Internet ISP backbones are composed of multiple points of presence or POPs connected together by high bandwidth backbone links, as shown in Figure 1. Each POP is a physical location where the ISP houses a collection of routers. The ISP backbone connects these POPs, and the routers attached to inter-POP links are called *backbone* or *core* routers. Each POP also locally connects customers through access links, ranging from large corporate networks to regional ISPs and web-servers. The POP routers attached to customers are called *access* routers. Within every POP, access routers provide an intermediate layer between the ISP backbone and routers in neighboring networks. Note that peering between POPs is provided either through dedicated links to another backbone (private peering) or through public Network Access Points (NAPs). To summarize, the general topology of a POP may be modeled by a two-level hierarchical structure as depicted in Figure 2. At the lower level, customer links are connected to *access* routers. These access routers are in turn connected to the *backbone* routers. The backbone routers provide connectivity to other POPs and to the peers.

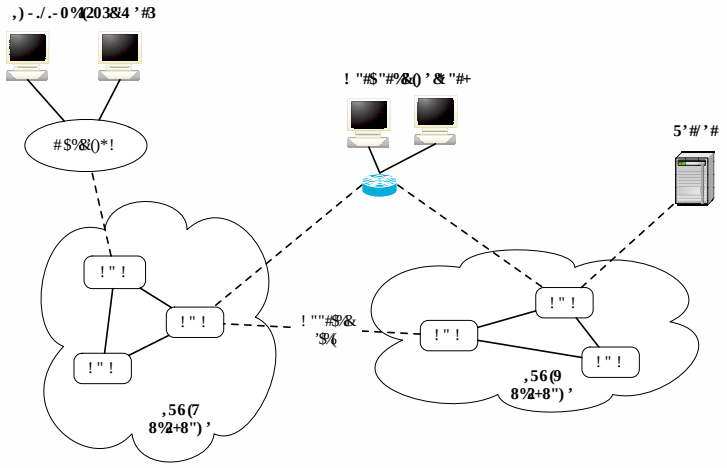

Figure 1: Internet ISP backbone. ISP backbones are composed of several POPs connected together by high bandwidth backbone links.

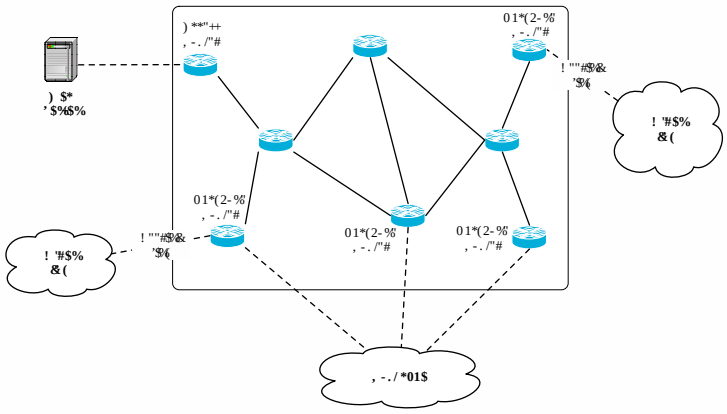

Figure 2: POP's architecture composed of backbone routers and access routers.

3 State of the art

Several famous projects focussed on network performance measurements. Metrology and monitoring are ongoing studies all around the world. First of all, we can cite the IPPM working group at IETF related to IP Performance Metrics [17] that develops a set of standard metrics that can be applied to the quality, performance, and reliability of Internet data delivery services; the IPFIX working group related to IP Flow Information Export [18] that aims to produce standards-track documents describing the IPFIX architecture, *i.e.*, information model and flow export protocol RFCs; the BMWG working group related to Benchmarking Methodology which makes a series of recommendations concerning the measurement of the performance characteristics of various inter-networking technologies; the PSAMP working group related to Packet Sampling and the IMRG research group at IRTF focused on Internet Measurement.

There also exist several large scale platforms and ambitious projects launched to measure the global internet: NIMI (National Internet Measurement Infrastructure) [16], NLANR Measurement and Network Analysis Group (NLANR/MNA) focused on the characterization of the behavior of high performance connection networks, and the IP Monitoring Project (IPMON)[1] at Sprint which is focused on building a general purpose measurement system for IP networks capable of collecting both detailed packet-level traffic statistics as well as delay, loss, and other network performance statistics.

It is obvious that network measurements are essential for assessing performance issues, identifying and locating problems. Network traffic measurements provide essential data for networking research and operation. The strategy to obtain network information through end-to-end measurements, known as Internet tomography, is therefore of great interest to the research community [19, 7, 10]. The majority of work on network tomography concentrates on either topology discovery, or link delay monitoring. A recent research [2] studies traffic demands in an IP backbone (collected at a major POP in a commercial Tier-1 IP backbone), identifies the routes used by these demands, and evaluates traffic granularity levels that are attractive for improving the poor load balancing that exists in POPs. In [9], the authors propose a passive measurement methodology to infer and keep track of the sender's congestion window (cwnd) and the connection round trip time (RTT) in order to provide a valuable diagnostic of end-user-perceived network performance. For passive monitoring, one should place passive devices (generally an optical splitter that copies all the data on the link[2]) to tap the link on which data needs to be collected, and to record to disks a part of all packets with generally a time-stamps indicating their arrival time.

Some recent researches show that active measurements can also be used to locate failures in IP networks [8, 15, 1]. Indeed, IP networks do not typically generate feedback state information, thus in order to perform traffic engineering, active monitoring should be deployed inside POPs. Active probing can help to detect and to locate link failure. An active probing system consists of several measurement points. Each measurement point, called a *beacon*,

[1] `http://ipmon.sprintlabs.com/`
[2] `http://dag.cs.waikato.ac.nz/`

can send IP messages to all nodes in the network. Each message sent from a beacon to a network node for the purpose of monitoring is called a *probe*. A failure is detected when consecutive probes do not use the same path in the network [15].

All these research studies and projects use extensively monitoring for diagnosis: detect and report problems or anomalies, management, configuration problems, resource provisioning, network dimensioning, value-added service, feedback to customers; Network Intrusion Detection Systems use passive network monitoring extensively to detect possible threats... However, collecting traffic data and analyzing such data from a Tier-1 ISP backbone reveals to be a real challenging task since it is expensive and time-consuming to deploy tap devices or active beacons in operational network. The measurement equipment must be installed in commercial network facilities where physical space and power are constrained, and which are, in some cases, not stalled by any human operators. Moreover, the traffic volume ranges from tens of Mb/sec on OC-3 access links to 10 Gb/sec on OC-192 backbone links, whereas data analysis involves processing terabytes of data.

In all projects and approaches listed above, the key point is to minimize the overhead (cost, management as well as deployment), in terms of tap devices for passive monitoring or in terms of active beacons and additional traffic for active monitoring. Thus, minimizing the number of devices and finding optimal strategic locations is a key issue, mandatory for deploying scalable monitoring platforms.

4 Integer Linear Programming

Mathematical Programming (MP) allows an appropriate and easy formulation of many combinatorial optimization problems [20]. In these problems, the goal is to find an optimal allocation of limited resources under some constraints. There exist different classes of MP. Two important classes are Linear Programming (LP) [11] and Integer Linear Programming (ILP) [21]. In Linear Programming, the function to optimize (also called the *objective function*) and the constraints are linear with respect to the variables (that can be real numbers). A LP problem can be generally expressed as follows:

$$\min\ c^T \cdot x$$
$$\text{s.t. } A \cdot x = b$$
$$c \in \mathbb{R}^n;\ x \in \mathbb{R}^{+n};\ A \in \mathbb{R}^{m \times n};\ b \in \mathbb{R}^m$$

where x is the vector of variables, A is a matrix of known coefficients, and c and b are vectors of known coefficients. $c \cdot x$ is the objective functions and $A \cdot x = b$ are the constraints.

In Integer Linear Programming, the objective function and the constraints are also linear with respect to the variables but some of the variables can only be nonnegative integers. An Integer Linear Programming is a all-integer ILP if all the variables are nonnegative integer. It is a mixed-integer ILP otherwise. If the most famous method to solve a LP problem is the simplex method [4], there exists several different methods (see for example [12, 14]). On the other hand there is no general solution that solves all the instances of the ILP problems. But

some solutions have been proved to be efficient on some instances (see for example [6, 5]). 0-1 Integer Linear Programming is a special case of integer programming where variables are required to be 0 or 1 (and not arbitrary integer). It is also NP-hard.

A wide range of fields, like manufacturing, transportation, telecommunication, etc. have problems that can be expressed in LP and ILP. Telecommunications remain an important field of applications for operational research techniques. Networks design or routing and reservation problems are examples of optimization problems that can find solutions via LP or ILP formulations. In this paper, we formulate the positioning of passive or active monitoring devices into a 0-1 ILP problem. We show that this formulation leads to a substantial improvement to the greedy and the previous proposed solutions.

5 Passive monitoring

In this section, we consider passive monitoring. As mentioned in Introduction, passive monitoring does not introduce traffic overhead in the network. On the other hand, the devices that monitor the traffic may be very expensive due to the process and the storage of data. It is thus very important to minimize the number of such devices to install and maintain in the network. Moreover, as we stated in Introduction, it is not necessary to monitor all the traffic and only a percentage may be enough.

5.1 The problem

Before formalizing the problem, we describe the network model we use. The network can be modeled as an undirected weighted graph $G = (V, E)$ where V is the set of nodes that represent the routers and E is the set of edges that represent the communication links that connect the routers. We consider the traffics in this network: a traffic is the aggregation of all flows that entering the POP at router node s and leaving the POP at router node t. The traffic between nodes s and t follows the internal routing strategy deployed by the ISP. The used bandwidth by a traffic is associated to a weight. The weight of a link is the sum of the weights of all the traffic that flow on this link.

The problem is to place on the links of the newtork the minimum number of measurement points in order to monitor at least $100.k\%$ (k is a given constant, $0 < k \leq 1$) of the traffic. Henceforth, this problem is called *Partial Passive Monitoring* or *PPM(k)* for short. PPM(1) is equivalent to the *Passive Monitoring* problem.

Theorem 1 *The PPM(k) ($0 < k \leq 1$) problem is NP-complete.*

Proof: First, we show that PPM(1) is NP-complete. For the ease of the proof, we assume that each traffic has a weight of 1. If it is not the case, the traffic is splited into traffics of weight 1. PPM(1) is obviously in NP. We are now going to show that the problem of findind a dominating set of size smaller than h in a bipartite graph (called DSBG henceforth), which is a NP-complete problem [3], can be reduced to PPM(1) in polynomial time. DSBG

can be stated like: given $BG = (V_1 \cup V_2, BE)$, a bipartite graph, is there a subset of V_1 of size smaller than h dominating V_2?

Given $BG = (V_1 \cup V_2, BE)$ a bipartite graph, we can build from BG an instance of PPM(1) in polynomial time: each link of the network is given by a vertex of V_1 and each trafic in the network is given by a vertex of V_2. An edge $(u, v)|(u \in V_1; v \in V_2)$ in BG corresponds to the fact that the trafic v is sent through the link u of the network.

If a solution to DSBG of cardinality at most h exists, then it is the same for PPM(1): by placing the measurement points on the links corresponding to the dominating set given by DSBG, every trafic in the network will be monitored. On the other hand, if a solution to PPM(1) of cardinality at most h exists, then DSBG admits a solution involving at most h vertices: by choosing for DSBG the vertices associated to the monitored links in PPM(1), we dominate all the vertices of V_2 since all the traffic are monitored in PPM(1). Therefore, PPM(1) is NP-complete.

Lastly, PPM(k) ($0 < k \leq 1$) is clearly in NP. Since PPM(1) is NP-complete, then PPM(k) ($0 < k \leq 1$) is NP-complete. ∎

A natural way to solve the PPM(k) problem is to have a greedy approach: the most loaded link is first chosen, and so on. This algorithm does not of course lead to an optimal solution. For example in Figure 3, the POP has three traffics, two of weight 2 and two of weight 1 and we want to find a solution to PPM(1). The greedy approach selects the link with the two traffics of weight 2 first, *i.e.* the link of weight 4. In order to monitor all the traffics, we need to select other links, for instance the two links with weight 1. This solution gives three measurement points, whereas an optimal solution is to place two measurement points on the two links of weight 3.

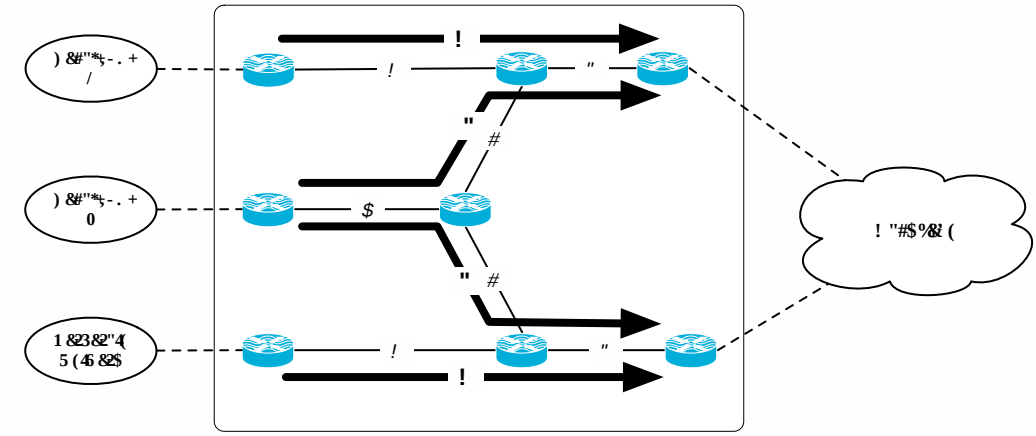

Figure 3: Passive measurement on a POP example

In order to find better solutions, we can express the PPM(k) problem into an Integer Linear Programming problem. Consider the following problem:

$$\min \sum_{i=1}^{m} x_i$$

$$\text{s.t. } \forall j \in [1,t] \sum_{i=1}^{m} M[j,i]x_i \geq \delta_j$$

$$\text{and } \sum_{j=1}^{t} \delta_j v_j \geq k \sum_{j=1}^{t} v_j$$

$$\forall i \in [1,m], x_i \in \{0,1\}; \forall i \in [1,t], \delta_i \in \{0,1\};$$

$$M \in \{0,1\}^{t \times m}; \forall i \in [1,t], v_i \in \mathbb{R}^+; k \in [0,1]$$

where m is the number of links in the network, t is the number of traffics in the network, $v = (v_j)_{j \in [1,t]}$ is the traffic weights vector composed of the aggregated throughputs of each traffic, M is the traffic matrix ($M[j,i] = 1$ if the traffic j passes on link i and $M[j,i] = 0$ otherwise), $\delta = (\delta_j)_{j \in [1,t]}$ is a $(0,1)$-vector representing the status of each traffic ($\delta_j = 1$ if the flow j is monitored) and $x = (x_i)_{i \in [1,m]}$ is the measurement points vector ($x_i = 1$ if a measurement point is placed on link i, $x_i = 0$ otherwise).

It is easy to see that this ILP problem is equivalent to the PPM(k) problem: the first constraint allows to take into account the traffic that will be monitored and thus to count the volume of monitored traffic, the second constraint allows to monitor enough traffic and the goal is to minimize the number of located devices.

Note that the Partial Passive Monitoring problem can be tackled with a different point of view. The devices for the measurement points may be very expensive and the operators may have a very limited number of such devices. Within this context, the problem can be the following: given d the maximum number of available devices, what the maximum amount of traffics that can be monitored is. This problem can also be expressed into an Integer Linear Programming problem:

$$\max \sum_{j=1}^{f} \delta_j v_j$$

$$\text{s.t. } \forall j \in [1,t] \sum_{i=1}^{m} M[j,i]x_i \geq \delta_j$$

$$\text{and } \sum_{i=1}^{m} x_i \leq d$$

$$\forall i \in [1,m], x_i \in \{0,1\}; \forall i \in [1,t], \delta_i \in \{0,1\};$$

$$M \in \{0,1\}^{t \times m}; \forall i \in [1,t], v_i \in \mathbb{R}^+; d \in \mathbb{N}$$

where m, t, v, M, δ and x are the same variables and coefficients as in the previous problem and where d is the maximum number of available devices, *i.e.* the maximum number of measurement points that can be afforded.

Note also that this formulation allows to compute an incremental solution: if new measurement devices are available, then one problem may be to maximize the number of monitored traffic with these new devices without moving the devices already located. The variables x_i associated to the previously monitored link remain to 1, and the integer linear programming is applied to the problem where the unknown variables correspond to the links not monitored.

5.2 Simulation results

In order to evaluate and compare the greedy approach (that first selects the link with largest weight, and so on) and our Integer Linear Programming formulation of the Partial Passive Monitoring problem we run simulations on several POP topologies. We use ISP topologies that are inferred by the Rocketfuel tool [19].

For the sake of simplicity, we assume as in [15] that the traffic insides a POP uses shortest path routing from router s where it is entering the POP to router t where it is leaving the POP. As opposed to [1] we do not make the assumption that the routing uses symmetric path, that is, that the path $P_{u,v}$ used for routing from u to v is the routing path in the opposite direction from node v to node u. Note that we consider the traffic entering and leaving the POP. Therefore the generated network has some virtual nodes that represent sources and targets of the traffic and that are not considered as routers in the POP.

Since we do not have real available data of traffic matrice issue from the considered POP topologies, we randomly generate several traffic matrices. In [2], the authors's analysis shows that the geographical spread of traffic across egress POPs is far from uniform. They do explain that this non-uniform behavior comes from the intrinsic way the Internet is designed (*e.g.*, some POPs would sink higher traffic demands than others because of their geographical location). In order not to generate uniform traffic distribution between all access routers and backbone routers, we randomly pick some prefered pairs of high traffic (for example between two backbone routers or between one backbone router and one access router that would host a popular web site). Figure 4 shows a simple POP and the traffic load generated randomly.

All the results are an average over 20 simulations. To solve this $0-1$ ILP problem we use the LP_SOLVE library[3]. This linear programming code can handle integer programming.

Figure 5 presents the results for the devices placement on a POP with 10 routers. In this configuration, the POP has 27 links and 132 traffics go through this POP. We compare our algorithm with the greedy algorithm. The x-axis corresponds to the percentage of traffic that is monitored (we start from 75%), and the y-axis is the number of devices located by the solutions.

[3] ftp://ftp.es.ele.tue.nl/pub/lp_solve

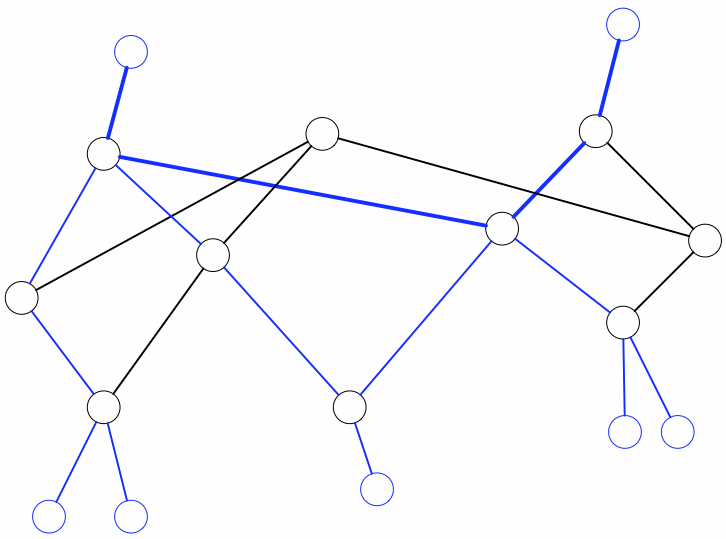

Figure 4: Traffic weight on a simple POP. The thickness of an edge represents the percentage of traffic on this edge. Our traffic matrice does not generate uniform traffic.

First we see that, until 95%, with our solution, the number of located devices is almost linear in the percentage of the monitored traffic. But when the percentage switches from 95% to 100%, the number of required devices drastically increases: we need twice more devices to monitor extra 5% percent of the traffic. This result indicates that it can be worthy in terms of cost overhead to not monitor all the traffics but only 95% of them.

We see also that our solution matches the greedy solution. In average, the greedy solution is twice as large as our solution.

Figure 6 presents the results for the devices placement on a POP with 15 routers. This POP has 71 links and there are 1980 traffic flowing in the POP. We also compare our solution with the greedy solution and the axes are equivalent to Figure 5. In this case, we can observe three stages: from 75% to 85%, the increase of located devices linearly increases with the percentage, then from 85% to 95%, the increase is also linear but the slope is larger, and finally there is a big increase in the number of located devices when we swith from 95% to 100% of the monitored traffic. In that case, the number of devices ranges from 16 to 41. It leads us to the same conclusion than with the previous result: it can be very cost effective to monitor only 95% of the traffic.

We see also that our algorithm is better than the greedy solution, but the gap in that case is smaller than the one obtained on a POP with 10 routers. This is probably due to the fact that the traffic, even if it is non-uniform, is more concentrated in the POP with 10 routers and thus more well balanced than in a POP with 15 routers. With the presence of

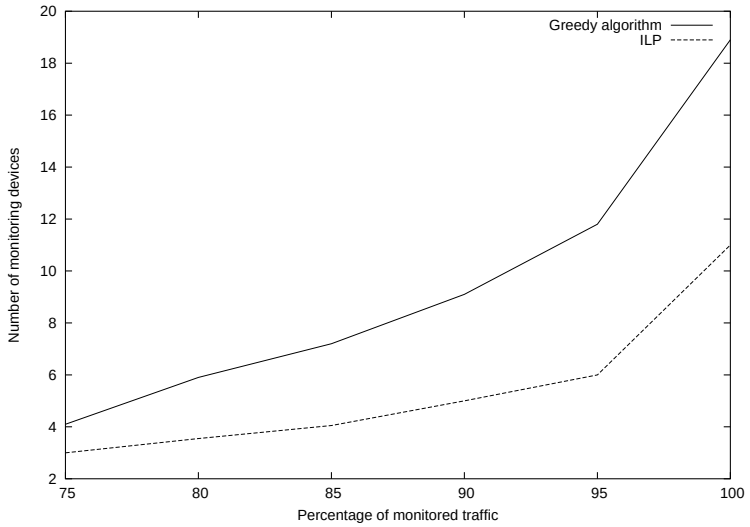

Figure 5: Passive monitoring: devices placement on a 10 routers POP

more uniform traffics, it is probably easier to find counter-example topologies as the one we presented in Figure 3 and therefore the optimization is more effective.

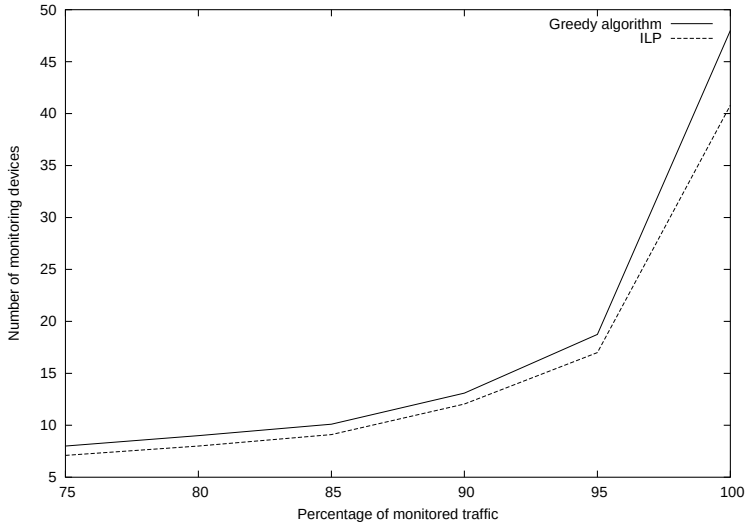

Figure 6: Passive monitoring: devices placement on a 15 routers POP

6 Active monitoring

Active monitoring has received much more attention than passive monitoring if we compare the literature. If this approach implies overhead traffic, it keeps a control on the measurement. Usually, the objective is to find the minimum number of beacons whose probes cover all the links in the network (see [1, 8] for recent references). When the beacons are chosen, the smallest set of probes has to be computed. Recently, the authors of [15] propose a different approach: starting from a set of possible beacons, they first compute an optimal set of probes and then locate the beacons. They show that the beacon placement problem is NP-hard and use a greedy algorithm for this problem: they first select a beacon, remove the set of probes that can be sent with this beacon, and so on.

6.1 The problem

For this problem, we use the network model of [15], *i.e.* an undirected graph $G = (V, E)$ with V the set of nodes that represent the network elements and E the set of edges that represent the links connecting the elements. The network has a set of possible beacons, called V_B henceforth ($V_B \subseteq V$). Starting from this set V_B, the authors of [15] designed a polynomial algorithm that computes the optimal set of probes. Then from this set of probes, they choose the effective beacons. In this section, we propose to optimize this placement phase. Note that in this problem, the beacons are placed on the nodes (the routers) and not on the links unlike the passive monitoring.

Once again the beacon placement problem can be translated into a $0-1$ Integer Linear Programming problem. Assume that P is the optimal set of probes obtained with the algorithm of [15]. Each probe $p \in P$ is given by its two extremities p_s and p_t, knowing that the probe from p_s to p_t is equal to the probe from p_t to p_s. The Integer Linear Programming problem is the following:

$$\min \sum_{i=1}^{n} x_i$$

$$\text{s.t. } \forall i \in V \backslash V_B \; x_i = 0$$

$$and \; \forall p \in P, x_{p_s} + x_{p_t} \geq 1$$

$$\forall i \in [1, N], x_i \in \{0, 1\}$$

where $n = |V|$ is the number of nodes in the network and $x = (x_i)_{i \in V}$ is the variable ($x_i = 1$ places a beacon on node i in the network, $x_i = 0$ otherwise).

It is easy to see that this ILP problem is equivalent to the beacon placement problem: the first constraint prevents from placing beacons on forbidden nodes, *i.e.* nodes not in V_B, the second constraint ensures that each probe of P will be sent by one beacon and the goal is to minimize the number of located beacons.

Note that we can also propose a greedy solution that should give better results than the one of [15]. Rather than arbitrarily choosing beacons, we can select the beacon that will

generate the most probes first, then remove these probes to the set of probes, and so on. We also test this greedy solution in our simulations.

6.2 Simulation results

The POP topology is generated with the same way as in Section 5. We implemented the algorithm of [15] that computes the optimal set of probes. From this set P, we compute the beacons placement with the algorithm proposed in [15], our greedy algorithm and our ILP solution. Again, to solve the $0-1$ ILP problem we use the LP_SOLVE library. All the results are the average over 20 simulations.

Figure 7 presents the results for the beacons placement on a POP with 15 routers. We compare the algorithm of [15] (called Thiran in the figure), our greedy algorithm (called greedy in the figure) and our solution based on an ILP formulation. The x-axis is the size of V_B (*i.e.* the potential beacons) and y-axis gives the number of located beacons. We see that our solution always places the fewest number of beacons and the gap between the algorithm [15] and our solution increases with the number of possible beacons (size of V_B). This may be explained by the fact that when V_B is small there is few possible optimizations, whereas when V_B is large there are more opportunities to optimize the beacons placement, and in that case the ILP formulation is effective. When $|V_B| = 15$, our solution decreases by a factor 2, the solution of [15]. Note that our greedy solution gives also good results compared to the algorithm of [15] and is quite close to the ILP solution since that for 8 possible beacons they differ only from 1 in the number of located beacons.

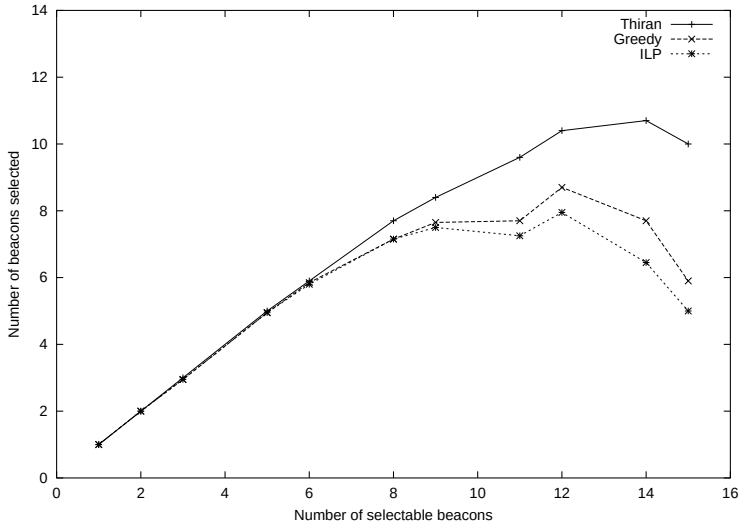

Figure 7: Active monitoring: beacons placement on a 15 routers POP

Figure 8 presents the results for the beacons placement on a POP with 29 routers. We have the same kinds of results than the ones obtained with 15 routers. The ILP solution matches the two greedy solutions. The best result is obtained on a POP with 29 routers: the number of beacons is reduced by 33%. Our greedy algorithm is also very close to the ILP solution: they differ for 15 possible beacons and the difference is at most 2 beacons.

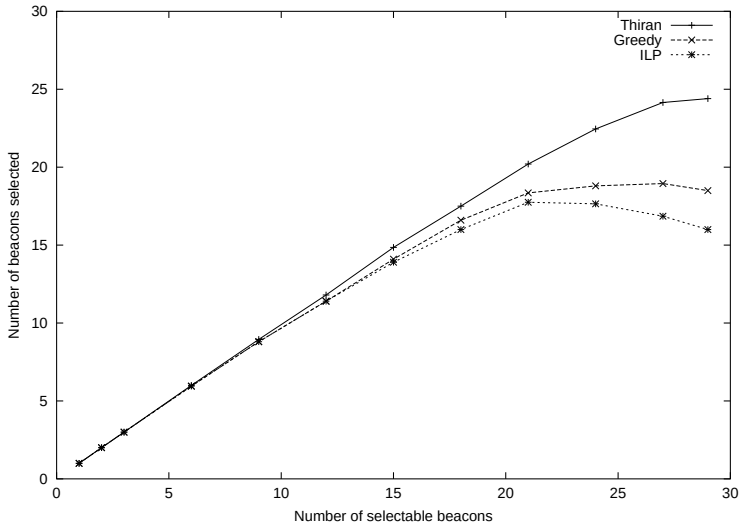

Figure 8: Active monitoring: beacons placement on a 29 routers POP

Figure 9 presents the results for the beacons placement on a POP with 80 routers. Once again the same kind of conclusions can be drawn. The number of beacons is also reduced by 33% when we use our algorithm instead of the algorithm of [15]. Note that in that case, the differences between our greedy solution and our ILP solution are more noteworthy than in the other POPs tested. With 80 possible beacons, the greedy solution places 7 extra beacons.

In all the curves, the number of located beacons decreases from a certain threshold on V_B with the ILP solution (it is also the case for the other solutions but not with all the topologies). It seems that having more opportunities to place the beacons allows a better placement of the beacons. Therefore, it may be more interested to offer a larger set of routers to place the beacons.

7 Conclusion

In this paper, we have provided novel contributions and addressed several issues concerning the positioning of active and passive monitoring devices. We have shown that the problem of minimizing the number of tap devices in order to monitor a fraction k of the total traffic

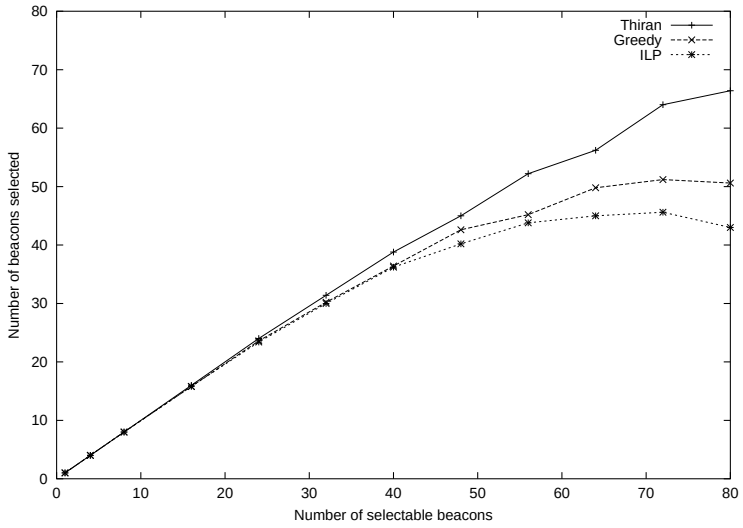

Figure 9: Active monitoring: beacons placement on a 80 routers POP

is NP-complete, even for $k = 1$ (*i.e.*, all the traffic is monitored). We use Integer Linear Programming in order to minimize the infrastructure cost of both passive and active monitoring. Based on the ILP theory, we are able to prove that for the passive monitoring it may be very costly for an ISP to monitor all the traffic and that it is worthy to monitor only a certain amount of the total traffic.

Our approach based on ILP is also useful for Active monitoring when the goal is to minimize the number of beacons set up in the POP network. We proposed one very simple greedy algorithm and one ILP based approach that both out perform the heuristic proposed in [15]. Note that our greedy solution has good performance on not too large POP (like 15 and 29 routers).

For the future, several possible extensions of this work are open to investigation. We are currently working on the generalization of our results since the ILP based approached may be used in an incremental approach. Indeed, once tap devices are deployed, ISP may try to deploy additional probed devices without changing actual devices. Once more, it is simple to modified the constraint of the ILP formulation to incorporate such new constraint. We also plan to study the effect of the traffic uniformity on the solutions: how the results are affected by the uniformity. We are also currently testing our solution on larger POPs, with at least 150 routers.

References

[1] Y. Bejerano and R. Rastogi. Robust Monitoring of Link Delays and Faults in IP Networks. In *IEEE INFOCOM'03*, 2003.

[2] S. Bhattacharyya, C. Diot, J. Jetcheva, and N. Taft. POP-Level and Access-Link-Level Traffic Dynamics in a Tier-1 POP. In *ACM SIGCOMM Internet Measurement Workshop (IMW)*, San Francisco, November 2001.

[3] A. Brandstädt. *Topics in Combinatorics and Graph Theory*, chapter On the Domination Problem for Bipartite Graphs, pages 145–152. Physica-Verlag Heidelberg, 1990.

[4] G. B. Dantzig. Programming of interdependent activites: II. mathematical model. *Econometrica*, 17:200–211, 1949.

[5] A. Geoffrion and R. Marsten. Integer programming algorithms: a framework and state-of-the-art survey. *Management Science*, 18:465–491, 1972.

[6] R. E. Gomory. Outline of an algorithm for integer solutions to linear programming. *Bulletin of the American Mathematical Society*, pages 275–278, 1958.

[7] G. Govindan and H. Tangmunarunkit. Heuristics for internet map discovery. In *INFOCOM'00*. IEEE, 2000.

[8] Joseph D. Horton and Alejandro Lopez-Ortiz. On the Number of Distributed Measurement Points for Network Tomography. In *IMC'03*, 2003.

[9] Sharad Jaiswal, Gianluca Iannaccone, Christophe Diot, Jim Kurose, and Don Towsley. Inferring TCP Connection Characteristics Through Passive Measurements. In *To appear in IEEE Infocom*, Hong Kong, March 2004.

[10] S. Jamin, C. Jin, Y. Jin, D. Raz, and L. Zhang. On the placement of internet instrumentation. In *INFOCOM'00*. IEEE, 2000.

[11] H. Karloff. *Linear Programming*. Birkhauser, 1991.

[12] N. Karmakar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[13] M. Kodialam and T. V. Lakshman. Detecting Network Intrusions via Sampling: A Game Theoretic Approach. In *INFOCOM*. IEEE, 2003.

[14] Kurzhanskiand, A. B. and Vàlyi, I. *Ellipsoidal Calculus for Estimation and Control*. MA:Birkhäuser, 1996.

[15] H. X. Nguyen and P. Thiran. Active Measurement for Multiple Link Failures Diagnosis in IP Networks. In *5th International Workshop on Passive and Active Network Measurement (PAM 2004)*, number 3015 in LNCS, pages 185–194, Antibes Juan-les-Pins, France, April 2004. Springer.

[16] A. K. Paxson, V. Adams and M. Mathis. Experiences with NIMI. In *Passive & Active Measurement Workshop (PAM 2000)*, Hamilton, New Zealand, April 2000.

[17] V. Paxson, G. Almes, J. Mahdavi, and Mathis M. Framework for IP Performance Metrics. RFC 2330, IETF, May 1998.

[18] J. Quittek, T. Zseby, B. Claise, and S. Zander. Requirements for ip flow information export. DRAFT draft-ietf-ipfix-reqs-16.txt, IETF, 2004.

[19] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with rocketfuel. In *SIGCOMM*. ACM, 2002.

[20] W. L. Winston. *Introduction to Mathematical Programming: Applications and Algorithms*. PWS-Kent, 1991.

[21] L. A Wolsey. *Integer Programming*. J. Wiley, 1998.

Unité de recherche INRIA Rhône-Alpes
655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399