HAL
archives-ouvertes.fr

Optimal positioning of active and passive monitoring devices

Claude Chaudet, Eric Fleury, Isabelle Guerin Lassous, Hervé Rivano, Marie-Emilie Voge

▶ **To cite this version:**

Claude Chaudet, Eric Fleury, Isabelle Guerin Lassous, Hervé Rivano, Marie-Emilie Voge. Optimal positioning of active and passive monitoring devices. CoNEXT 2005, Oct 2005, Toulouse, France. 2005.

HAL Id: hal-00429836
https://hal.archives-ouvertes.fr/hal-00429836

Submitted on 4 Nov 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimal Positioning of Active and Passive Monitoring Devices

Claude Chaudet

Claude.Chaudet@enst.fr

GET/ENST
LTCI-UMR 5141 CNRS
46, rue Barrault
75634 Paris, France

Eric Fleury,
Isabelle Guérin Lassous

{Eric.Fleury,
Isabelle.Guerin-
Lassous}@inria.fr

INRIA ARES Project
Laboratoire CITI
INSA de Lyon
21, avenue Jean Capelle
69621 Villeurbanne Cedex
France

Hervé Rivano,
Marie-Emilie Voge*

{Herve.Rivano,Marie-
Emilie.Voge}@sophia.inria.fr

CNRS/I3S/INRIA Mascotte
INRIA Sophia Antipolis
2004 route des lucioles
06902 Sophia Antipolis Cedex
France

ABSTRACT

Network measurement is 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 through the network and the active approach that generates explicit control packets in the network for measurements. One of the key issues in this domain is to minimize the overhead in terms of hardware, software, maintenance cost and 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. In this article, we present a combinatorial view of the problem from which we derive complexity and approximability results, as well as efficient and versatile Mixed Integer Programming (MIP) formulations.

Categories and Subject Descriptors

G.1.6 [**Optimization**]: Constrained optimization

General Terms

THEORY,PERFORMANCE

*This work has been partially supported by the European IP IST-FET CRESCCO.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CoNEXT'05, October 24–27, 2005, Toulouse, France.
Copyright 2005 ACM 1-59593-097-X/05/0010 ...$5.00.

Keywords

Passive Monitoring, Active Monitoring, Optimization

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 (POPs), as shown on Figure 1. 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, enhancing 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. Denial of service attacks, for instance, can be detected by noticing a sudden and important increase in the number of short-lived flows originated at random IP addresses [13].

In this work, we seek to minimize the infrastructure cost of both passive and active monitoring. For passive monitoring, we study the problem of sampling packets and thus we present efficient way of placing monitor devices and how to control their sampling rates. Sampling is crucial since all monitoring devices are not able to sustain a 100 % sampling rate on high speed links (OC-48, OC-192 and higher), since the exploitation cost of the monitoring devices may depend on their sampling rate and also because 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 [12], or to keep track of the values of two important variables associated with TCP connections [10]: the sender's congestion window (cwnd) and the connection round trip time (RTT).

We present a combinatorial view of the problem, giving rise to complexity and approximability results, as well as efficient Mixed Integer Programming (MIP) formulations. The main advantages of such kind of modeling is that it formalizes all greedy solutions that we proposed in prior work [3] and that were also simultaneously and independently applied in [22]. Moreover, from this new model we are able to derive MIP formulations even for the minimization of the deployment and the exploitation cost while maximizing the total amount of traffic monitored whereas in [22], the authors only present a mixed-integer non-linear program formulation of the problem. Finally, this formulation allows tackling slightly different problems. For instance, it is possible to compute incremental solutions. From a set of already installed devices that cannot move, the program can compute the best way to position a new set of monitors. This problem can be derived into the estimation of the expected gain in buying one or a set of new devices. It is also possible, by only adding a constraint in the modeling, to address the problem of finding the best positioning of a limited number of devices.

Since the traffic inside a POP may evolve one may point out that a drastic change in the traffic throughput may invalidate all previous optimizations done and will degrade the results that the operator will get. To overcome this problem, we present an efficient polynomial algorithm that will recompute optimal sampling rates for all monitoring devices already deployed in order to maximize the coverage while minimizing the exploitation cost. 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. The main discussion begins in Section 4 in which we describe our main contribution on passive devices positioning when taking into account the deployment cost and we show simulation results. In section 5 we extend the results on passive monitoring by introducing a sampling capability to each monitoring device and by taking into account an associated exploitation cost. In Section 6 we focus on active monitoring for which a similar strategy is used to improve beacons positioning. Finally Section 7 summarizes the results presented and discusses their implications on current monitoring strategies. Possible extensions to this work open for investigation are discussed.

2. GENERAL ARCHITECTURE

We present in this section the general network architecture considered in our study. We focus on the POP architecture and topology since POPs represent the key place where monitoring can be performed 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 [10].

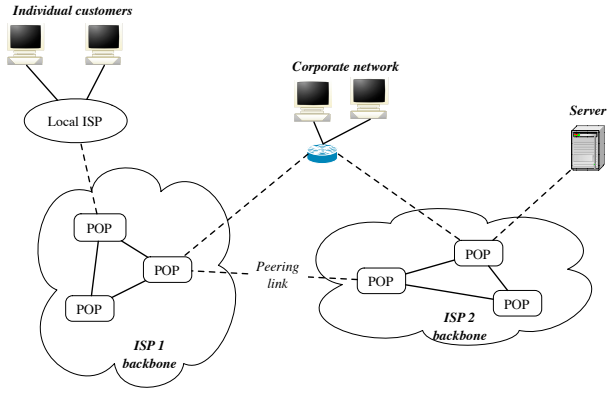

Figure 1: Internet ISP backbone. ISP backbones are composed of several POPs connected together by high bandwidth backbone links.

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 corresponds to 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 through access links customers 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.

3. STATE OF THE ART

Several famous projects focused on network performance measurements. Metrology and monitoring are ongoing studies all around the world. The IPPM working group at IETF

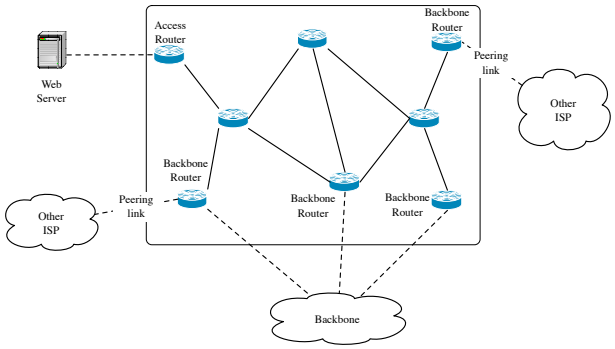

Figure 2: POP architecture composed of backbone routers and access routers.

related to IP Performance Metrics [17] 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] 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 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 [8, 11, 21]. The majority of contributions on network tomography concentrates on either topology discovery, or link delay monitoring. A 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 [10], 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, usually including a time-stamp that indicates their arrival time.

Some recent researches show that active measurements can also be used to locate failures in IP networks [9, 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*, 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: detecting and reporting 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/s on OC-3 access links to 10 Gb/s on OC-192 backbone links, whereas data analysis involves processing terabytes of data.

In all projects and approaches listed above, the key objective is to minimize the overhead (cost, management as well as deployment), in terms of number of tap devices for passive monitoring or in terms of number of active beacons and volume of 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.

[22] present heuristics for positioning passive monitors in POP and controlling their sampling rate, when monitors do only capture a portion of the traffic carried by the link they are attached to. They consider three main problems, the first one consisting in maximizing the volume of captured traffic under cost constraints, each monitoring device having a deployment and an operational cost. The second problem consists in minimizing the deployment cost to achieve a monitoring objective and the last one consists in minimizing both installation and operational cost under the same objective. They show that all these problems are NP-complete and they present heuristics approximating the optimal solution for each one. They evaluate the performance of the proposed algorithms with simulations on topologies discovered by the Rocketfuel utility and with generated traffic matrices.

4. 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 requirements for processing and storing collected data. It is thus very important to minimize the number of

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

such devices to install and maintain in the network. Moreover, as stated in Introduction, it is not necessary to monitor the whole traffic and only a percentage may be enough.

In the following we present a combinatorial view of the problem, giving rise to complexity and approximability results, as well as efficient Mixed Integer Programming (MIP) formulations.

4.1 Combinatorial model

Before formalizing the problem, we describe the network model we use.

Let us consider a POP, this network can be modeled as a 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.

A traffic t in this network is a single path p_t between two routers, or nodes of V, and a weight v_t given by the bandwidth routed along this path. Such a traffic is the aggregation of all IP flows which follows the path p_t through the POP. This path is defined by the internal routing strategy deployed by the ISP.

We call the load of a link the sum of the weights of all the traffic that flow on this link.

In this first study, we consider that a measurement point installed on a link e monitors all the traffic that flows on e. Therefore, monitoring a proportion k $(0 < k \leq 1)$ of the traffics carried by the network consists in selecting a subset of the links where to install measurement points, so that enough traffics are conveyed by monitored links.

The *Partial Passive Monitoring* problem is to find such a subset of a minimum size. This problem is denoted $PPM(k)$ for short, and can be stated as follows:

- **INSTANCE** $k \in]0, 1]$, $G = (V, E)$ a graph, $D = \{(p_i, v_i)\}$ a set of weighted paths (traffics).

  $\mathcal{V} = \sum_i v_i$ is the total bandwidth carried by the network.

- **SOLUTION** A subset $E' \subseteq E$ of selected edges such that $\sum_{i|\exists e \in E', e \in p_i} v_i \geq k\mathcal{V}$ meaning that the sum of the weights of the paths that go across a selected edge is greater than a proportion k of $\mathcal{V}$.

- **MEASURE** Cardinality of E'.

Note that $PPM(1)$ consists in monitoring all the traffics in the network and is henceforth called the *Passive Monitoring* problem.

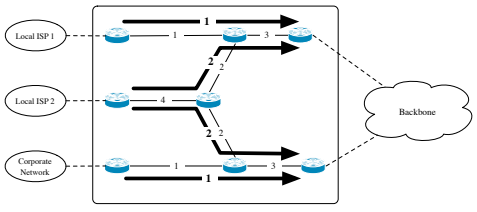

Figure 3: Passive measurement on a POP example

In the following, this combinatorial formulation is used for deriving complexity and approximability results.

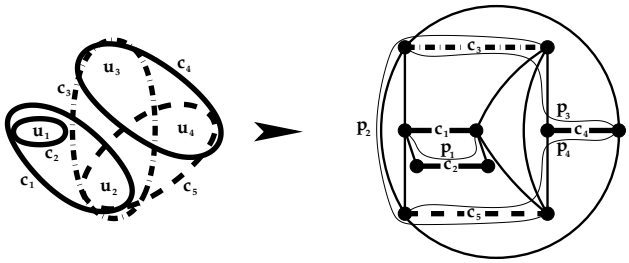

Figure 4: MSC - $PPM(1)$ equivalence

4.2 Complexity of Passive Monitoring

In this part, we focus on the Passive Monitoring, that is the specific case of $PPM(k)$ where $k = 1$. We prove that this case of the monitoring problem is equivalent to the *Minimum Set Cover problem*, yielding NP-completeness and tight approximability results.

Minimum Set Cover. Suppose that a set S of arbitrary items is given, as well as a collection of overlapping subsets of S. The *Minimum Set cover problem*, MSC for short, consists in finding a minimum size sub-collection such that any item belongs to a selected set. MSC can be stated as follows.

- **INSTANCE** Collection $C = \{c_1, \ldots c_m\}$ of subsets of a finite set $S = \{u_1, \ldots, u_n\}$.

- **SOLUTION** A set cover for S, *i.e.*, a subset $C' \subseteq C$ such that every element in S belongs to at least one member of C'.

- **MEASURE** Cardinality of the set cover, *i.e.*, $|C'|$.

Equivalence and complexity. Intuitively, the items represent the traffics while the subsets are the links of the network. MSC models the optimization goal of installing measurement points on a minimum size set of links, such that any traffic belongs to a selected link. The following theorem claims that both MSC and $PPM(1)$ are equivalent.

Theorem 1. *The Monitoring problem for $k = 1$ is equivalent to the Minimum Set Cover problem.*

Proof: At first we construct an instance of the monitoring problem from an arbitrary instance of Minimum Set Cover as depicted in Figure 4. Let G be a graph whose edge set E is defined as follows:

- E contains an edge e_i for each $c_i \in C$,

- if $c_i \cap c_j \neq \emptyset$, E contains an edge e_{ij} and an edge e_{ji}, both adjacent to e_i and e_j so that these four edges form a cycle,

Note that only $2|C|$ vertices are necessary to define E and thus G.

Then the set of traffics, D, contains a traffic t_i for each element u_i of S. The path p_i associated to t_i goes through

edge e_j if and only if u_i belongs to c_j. In addition p_i can use any edge e_{jk} provided it also uses e_j and e_k. Such paths can always be found[3] in polynomial time by construction of G. Since the whole traffic is to be monitored, assigning them a volume is useless.

Now suppose E' is an optimal solution of this monitoring instance. Then we deduce an optimal solution C' for the Minimum Set Cover instance from E' in the following way:

- if $e_i \in E'$, then $c_i \in C'$,
- if $e_{ij} \in E'$, then neither e_i nor e_j belongs to E' otherwise e_{ij} would be redundant and E' would not be optimal. Thus e_{ij} can be replaced either by e_i or by e_j in E', which means, either $c_i \in C'$ or $c_j \in C'$,

The minimum cardinality of E' implies the same property on C' which is therefore an optimal cover for this Minimum Set Cover instance.

Subsequently, consider an instance of the monitoring problem on a graph $G = (V, E)$ for which $k = 1$. Each edge e of G belongs to a set π_e of paths of D. Installing a measurement point on e means that every $p_i \in \pi_e \subseteq D$ is monitored.

An instance of MSC can be constructed from this monitoring problem taking $S = D$ and $C = \{\pi_e,\ e \in E\}$. It is clear that an optimal solution for MSC yields an optimal solution for the monitoring problem. This completes the proof of the equivalence of these two problems. ∎

As far as MSC is a NP-Complete problem, previous theorem implies directly the NP-completeness of $PPM(1)$, hence the NP-completeness of $PPM(k), 0 < k \leq 1$.

If all traffics carry the same bandwidth, the problem becomes unweighted. Following the same scheme as above, one can prove the equivalence of the unweighted version of $PPM(k)$ to the *Minimum Partial Cover Problem* (see [19, 20] for a definition of the problem). This gives a straightforward proof of the NP-completeness of unweighted $PPM(k)$, for any given $k,\ 0 < k < 1$.

Approximability results. Since computing an optimal solution is a NP-complete problem, one can prefer to derive approximate solution. A k-approximation is a *feasible* solution of the problem such that its cost is always bounded by k times the cost of an optimal solution.

The Minimum Set Cover problem is approximable within $\ln|S| - \ln\ln|S| + o(1)$ [19] with a simple greedy algorithm. The preceding equivalence result hence yields a polynomial time $(\ln|D| - \ln\ln|D| + o(1))$-approximation algorithm for the Passive Monitoring problem.

Moreover, the Minimum Set Cover is not approximable within $(1-\varepsilon)\ln|S|$ for any $\varepsilon > 0$, unless NP $\subset$ DTIME$(n^{\log\log n})$ [7], so the Passive Monitoring problem is not approximable within $(1-\varepsilon)\ln|D|$ for any $\varepsilon > 0$, unless NP $\subset$ DTIME$(n^{\log\log n})$.

In the following, we show that $PPM(k)$ can be modeled as a *Minimum Edge Cost Flow* in an auxiliary graph. This combinational model gives rise to efficient MIP formulations improving previous results of the literature, as well as an expressive theoretical framework for developing a more detailed and realistic model.

[3]Arbitrarily order the edges p_i has to use, by construction as u_i belongs both to c_j and c_k there is an edge e_{jk} linking the two consecutive edges e_j and e_k.

4.3 Partial Passive Monitoring

For all $k \leq 1$, we now introduce a model of the partial monitoring problem as a Minimum Edge Cost Flow, $MECF$ for short, in an auxiliary graph. The main advantage of such kind of model is that it leads to a mixed integer program whose computational time is better than those of [3, 22].

Another key advantage of the Minimum Edge Cost Flow model is that it formalizes all greedy solutions generally proposed [3, 22] to solve such a class of problems. All greedy approaches use a natural way to solve $PPM(k)$: the most loaded link is chosen first, and so on and so forth. This algorithm does not of course lead to an optimal solution, but rather to a $(\ln|D| - \ln\ln|D| + o(1))$-approximation since it is also related to the greedy algorithm for the Minimum Partial Cover Problem analyzed in [19]. For example in Figure 3, the POP carries four 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.

Minimum Edge Cost Flow formulation. The $MECF$ is a regular minimum cost flow problem, except a binary cost function, as stated below.

- **INSTANCE** $G' = (W, A)$ a directed graph, each arc $a \in A$ has a capacity u_a and a constant cost c_a, a flow request of volume F from a source vertex $S \in W$ to a sink vertex $T \in W$,
- **SOLUTION** a $S - T$ flow f satisfying the request,
- **MEASURE** the cost of f, note that an arc a costs c_a whenever the flow on arc a $f_a > 0$, and 0 otherwise,

In the following, we show how to convert a $PPM(k)$ instance into a $MECF$ instance. This transformation allows a better understanding of the combinatorial challenges yielded by $PPM(k)$, hence creating a combinatorial framework for heuristics development and analysis, and leading to an efficient MIP formulation.

Given an arbitrary instance of $PPM(k)$, $0 < k \leq 1$, let us define the following instance of $MECF$:

First a directed graph $G' = (W, A)$ has to be defined:

1. W contains a vertex w_e for each edge $e \in E$,
2. W contains a vertex w_t for each traffic $t \in D$,
3. W contains two additional vertices S and T,
4. there is an arc of unbounded capacity and cost 1 in A from S to each w_e. Thus each arc (S, w_e) corresponds to an edge e of the Monitoring instance,
5. there is an arc in A from w_e to w_t if and only if the path p associated to traffic t uses edge e. The capacity of such a arc is unbounded and its costs is null. These arcs represent the edge-path adjacency relation of the Monitoring instance,
6. there is an arc of capacity v_t, the volume of traffic t, and null cost in A from each w_t to T.

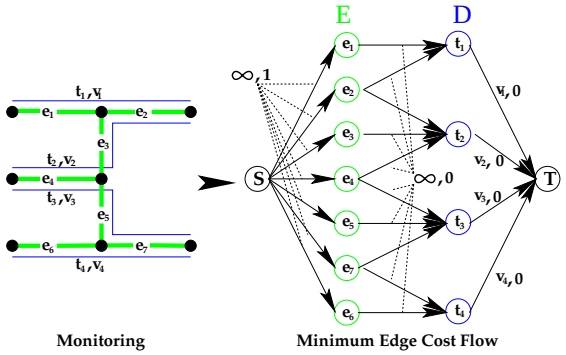

Figure 5: $MECF$ **instance for** $PPM(k)$

Then, the objective is to route from S to T a volume of flow equal to the volume of traffic to be monitored, $k\sum_{t\in D} v_t$.

Intuitively, the link between S and a w_e will support a flow if a measurement point is installed on e. The following theorem claims that the previous $MECF$ instance actually solves $PPM(k)$.

Theorem 2. *An optimal $MECF$ solution on G' yields an optimal solution for $PPM(k)$ on G.*

Proof: Consider a flow f solution of this Minimum Edge Cost Flow instance, it can be interpreted according to the Monitoring instance. Note that the only arcs of non null cost are the (S, w_e) arcs, therefore the cost of a solution equals the number of arcs (S, w_e) supporting non null flow. The set of edges of the Monitoring instance corresponding to these arcs is referred to as E'.

In a solution of the Minimum Edge Cost Flow, the flow on arc (w_t, T) may come from several arcs (w_e, w_t) and thus according to our present interpretation, the traffic t may be partitioned, each part being monitored on a different edge. Although this would not be in accordance with the monitoring without sampling problem, we can assume it never happens, otherwise it would be easy to deport all flow corresponding to traffic t on a single path since capacities are unbounded on all arcs but the (w_t, T) ones.

In addition no more than a volume v_t of traffic t can be taken into account in the Minimum Edge Cost Flow solution since the capacity of an arc (w_t, T) is v_t.

At last the total volume of flow going through vertices $w_e \ \ \forall e \in E'$ is at least $k \cdot \sum_{t\in D} v_t$ and has to go through arcs (w_t, T) which are reachable from these w_e, *i.e.* the arcs (w_t, T) corresponding to traffics using the edges $e \in E'$. E' is therefore a solution of the Monitoring instance, and the volume of flow routed through both w_e and w_t represents the volume of traffic t that the measurement point on edge e has to monitor.

Furthermore, if E^* is an optimal solution of the Minimum Edge Cost Flow instance, it is also an optimal solution of the Monitoring instance. Otherwise, let E'' be an optimal solution of the Monitoring instance, then $|E''| < |E^*|$ because every solution of the Minimum Edge Cost Flow instance is a solution of the Monitoring instance and E^* is not one of its optimal solutions.

On the other hand, a solution of the Minimum Edge Cost Flow instance can be built from E'' in the following way. First note that only one path, p_t^e that goes through both w_e and w_t exists. For each edge $e \in E''$ we add a flow of value v_t on path p_t^e if p_t uses edge e in the Monitoring instance and if t has not been treated by another edge yet. As a traffic t is treated only once, the capacity constraint on arc (w_t, T) in the Minimum Edge Cost Flow instance is respected, and the flow value is at least $k\sum_{t\in D} v_t$ since the volume of traffic monitored is at least of this amount. Thus this flow is a solution of the Minimum Edge Cost Flow but its cost is lower than the cost of $|E^*|$, which contradicts the optimality of E^*. ∎

Heuristics. Several previous papers proposed heuristics for $PPM(k)$ [3, 22]. They share a common general idea which is to always choose the edge which permits to monitor the larger volume of traffic not monitored yet, until the objective is attained.

The $MECF$ framework allows to analyze these heuristics in terms of flow. As a matter of fact, these heuristics appear as the computation of a minimum cost $S-T$ flow in the $MECF$ graph modeling of the Monitoring problem. This is indeed a linear relaxation of $MECF$ where the costs are no more binary but linear. In this relaxation, the link cost on (S, w_e) arcs, is the inverse of the load of edge $e \in E$. On every other arc, the link cost is null, like in the $MECF$ instance. Such a link cost configuration models the greedy behavior of previously defined heuristics.

The $MECF$ framework allows to develop other flow-based heuristics such as randomized rounding or branching algorithms.

Unfortunately, the general case of $MECF$ does not admit a $2^{\log^{1-\epsilon} n}$-approximation, for every constant $\epsilon > 0$, unless NP $\subseteq DTIME(n^{\text{polylog } n})$ [6]. Even though the $MECF$ instances related to $PPM(k)$ are very specific, the results derived for the unweighted case from the Minimum Partial Cover Problem shows non-approximability properties that are to be refined.

MIP formulation. There are two usual linear programming formulations of flow problems, the arc-path one and the vertex-arc one. Program 1 is the arc-path formulation to which binary variables (x_e) are added, representing whether an arc supports a non null flow or not. Corresponding constraints which permit to set these variables are also added.

LINEAR PROGRAM 1 (PPM(K)).

$$
\begin{aligned}
\textit{Minimize} \quad & \sum_{e\in E} x_e & & & & \\
\textit{s.t.} \quad & \sum_{t\in\pi_e} {f_t}^e & \leq & x_e \sum_{t\in\pi_e} v_t & \forall e \in E \\
& \sum_{e\in p_t} {f_t}^e & \leq & v_t & t \in D \\
& \sum_{t\in D}\sum_{e\in p_t} {f_t}^e & \geq & k\sum_{t\in D} v_t & \\
& {f_t}^e & \geq & 0 & \forall e \in E \ \forall t \in \pi_e \\
& x_e & \in & \{0,1\} & \forall e \in E
\end{aligned}
$$

- ${f_t}^e$: volume of flow on the path which goes through both w_e and w_t $\forall e \in E \ \forall t \in \pi_e$,
- x_e: 0 if the flow on arc (S, w_e) is null, 1 otherwise,

The first constraint means that the flow on paths going through vertex w_e cannot be positive if the arc (S, w_e) has not been payed for, the second constraint represents the capacity constraint on every arc (w_t, T), the third one represents the satisfaction of the flow request of volume $k \sum_{t \in D} v_t$. The cost function is the number of arcs (S, w_e) supporting a non null flow.

This formulation can be slightly modified to let appear it is a relaxation of the binary programs of [3, 22].

Actually rename δ_t the sum $\frac{1}{v_t} \sum_{e \in p_t} {f_t}^e$ and note that the first constraint can be replaced by:

$$ {f_t}^e \leq x_e v_t \quad \forall t \in D \quad \forall e \in p_t $$

and that these new constraints can be added to obtain:

$$ \delta_t \leq \sum_{e \in p_t} x_e \quad \forall t \in D. $$

Then the following formulation is obtained:

LINEAR PROGRAM 2 (PPM(K)).

$$
\begin{aligned}
\textit{Minimize} \quad & \sum_{e \in E} x_e & & & \\
t.q. \quad & \sum_{e \in p_t} x_e & \geq & \delta_t & \forall t \in D \\
& \sum_{t \in D} \delta_t \cdot v_t & \geq & k \sum_{t \in D} v_t & \\
& \delta_t & \in & [0,1] & \forall t \in D \\
& x_e & \in & \{0,1\} & \forall e \in E
\end{aligned}
$$

- x_e is equal to 1 if a measurement point is installed on e, to 0 otherwise,
- δ_t is the percentage of the volume of traffic t monitored,

This formulation also allows to compute an incremental solution: suppose that a whole monitoring architecture is already set-up and 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 are fixed to 1 and treated as constants, and the mixed integer programming is applied to the problem in which the unknown variables correspond to the links not monitored.

It is also possible, with only a slight modification of the program, to address situations in which an operator seeks how to optimally position a limited number of monitoring devices, simply by adding a constraint on the maximum number of affordable measurement points.

4.4 Simulation results

In order to evaluate and compare the greedy approach that selects links in decreasing weight order and our mixed integer programing 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 [21].

For the sake of simplicity, we assume as in [15] that the traffic inside 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 includes 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 matrix issued from the considered POP topologies, we randomly generate several traffic matrices. In [2], the authors' 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 preferred 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 6 shows a simple POP and the traffic load generated randomly.

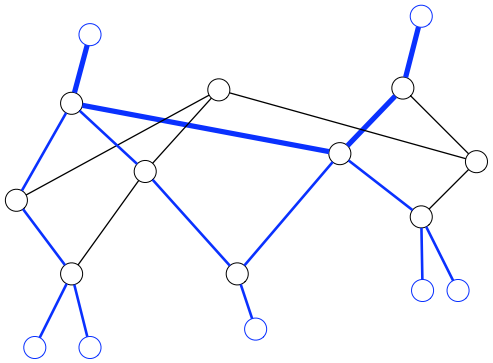

Figure 6: Traffic weight on a simple POP. The thickness of an edge represents the percentage of traffic on this edge. Our traffic matrix does not generate uniform traffic.

All the results are an average over 20 simulations. To solve this $0-1$ MIP problem we use CPLEX solver. Nevertheless, this linear programming code can handle integer programming.

Figure 7 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.

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 not to monitor all the traffics but only 95% of them.

Of course, our solution is better than the greedy, which is not surprising, but we also see that in average, the greedy solution is twice as large as our solution.

Figure 8 presents the results for the devices placement on

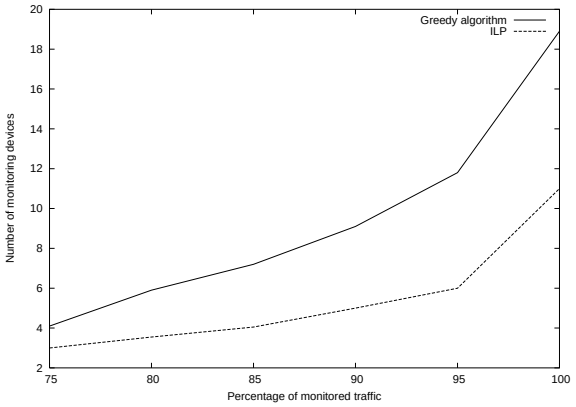

Figure 7: Passive monitoring: devices placement on a 10 routers POP

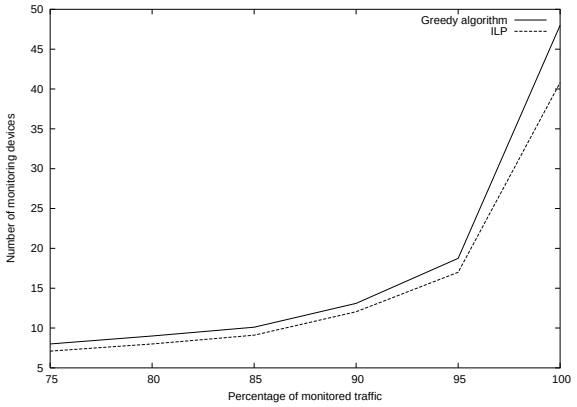

Figure 8: Passive monitoring: devices placement on a 15 routers POP

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 axis are equivalent to the ones of Figure 7. In this case, we can observe three steps: 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 switch from 95% to 100% of the monitored traffic. In that case, the number of devices ranges from 16 to 41. This leads us to the same conclusion as with the previous result: it can be very cost effective to monitor only 95% of the traffic.

We see also that, not surprisingly, our algorithm still performs 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 better balanced than in a POP with 15 routers. With the presence of 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.

5. PASSIVE MONITORING AND PACKET SAMPLING

Due to the huge number of packets going through a router on a high speed link (OC-48, OC-192, OC-255), the necessity of reducing the volume of monitored data is perfectly understandable. Reducing the amount of packets processed and stored may reduce the *exploitation cost* of the monitoring devices deployed inside the network. The ratio of packet sampling will depend on the exploitation cost of the devices and thus of the cost per packets and it may vary from one device to another one. When sampling is available, the passive monitoring consists in placing devices in order to monitor at least $k\,\%$ of the total traffic while minimizing the setup cost induced by each device installed and the exploitation cost induced by the sampling ratio assigned to each device. In the remainder we consider multi-routing. Indeed, for the sake of load balancing, the internal routing strategy deployed by the ISP might use several routes between a pair of source/destination routers. In previous sections, such a situation was tackled by considering each weighted route as a whole traffic. In the following, a traffic is given as a set of weighted routes between the source and the destination routers of the traffic. In combinatorial terms, a traffic is therefore a set of weighted paths between the same pair of source/destination nodes (u,v). Let either $\mathcal{P}_{u,v}$ or $\mathcal{P}_t$ denote the set of paths associated with traffic t of source u and destination v and $\mathcal{P} = \cup_t \mathcal{P}_t$.

Consequently, the administrator of the POP might need to monitor a part of each traffic, without necessarily monitoring every path. We therefore introduce h_t, the minimum monitoring ratio of a traffic t. Note that we have $h_t \leq k$ since h_t is related to the minimum cover of a traffic t whereas k is related to the minimum cover of the global amount of traffic.

5.1 Reducing the amount of data

Techniques to reduce the amount of data treated and stored may be classified into three main classes:

- **Filtering:** it consists in capturing only a subset of the frames based on some networking criteria (protocol, port number, ...);
- **Classification:** packets can also be classified into classes, statistics being calculated class by class;
- **Sampling:** packets can be captured randomly.

Sampling has many advantages. First, it does not require much computation, compared to the two other techniques, to filter or classify frames. Secondly, it does not require any configuration and is therefore more adaptive to new traffic patterns and therefore more able to detect malicious traffic.

5.2 Sampling techniques

Sampling, and reduction of the number of considered frames in general, raised many problems. Using only a subset of the frames to compute statistics biases the estimation and it is not always easy or even possible to infer the characteristics

of the original traffic from the sampled trace. The way packets are sampled has a great influence on the conclusions it is possible to draw from the reduced trace. In [4], Duffield presents different sampling techniques and their associated trade-offs.

- **Time-based sampling:** the monitor captures frames at regular time-intervals. This technique can suffer from time-constrained applications that send packets regularly. On low-speed links especially, there is a risk of only considering a subset of the flows and systematically missing important information.
- **Regular sampling:** the monitor captures exactly one frame every N frame. This technique exhibits better results than the previous one, as it is more likely to capture packets belonging to a burst. Nevertheless, it is still influenced by periodical traffics.
- **Probabilistic sampling:** the monitor captures frames with a probability $1/N$.
- **Probability distribution-based sampling:** the monitor captures one frame every X, X being a random variable following a given law (geometric, exponential) with mean N.

The french national project Metropolis[4] has studied the influence of this type of sampling on the perception of the flows in the network. Considering only one frame out of 1000, they use the classical mice (designing short flows) and elephant (designing long flows) separation of the flows and show sampling creates problems related to flows identification. With only one packet out of 1000, it is difficult to decide in which category fits one flow, as there is a low probability to monitor more than one or two packets of each elephant flow. Concerning mice, which is the most common type of flows, most of these flows will not be monitored and statistics drawn on sampled traces tend to over-estimate the volume of mice flows while increasing the corresponding estimated volume.

Some contributions [5, 14] study the problem of enhancing the estimation of the characteristics of the traffic from the sampled statistics. [14] studies more specifically the problem of identifying elephant flows with periodically sampled frames. They use the Bayes theorem to estimate the probability that a flow presenting more than y frames in the sampled trace is composed of more than x frames in the complete trace. [5] proposes that monitors count SYN packets, identifying the start of most of TCP connections, in order to estimate more accurately the number of flows. From this estimation, it is easier to infer real statistics from the sampled trace.

[22] considers sampling in the Budget Constrained Max Coverage Problem, *i.e.* the problem of finding the best positioning of monitoring devices under cost constraints, limiting the number of these devices. They consider that multiple devices monitoring different links carrying the same flow will only monitor this flow once. On the opposite, by using packets marking techniques, successively monitoring the same flow can lead to a monitoring percentage equal to the sum of the sampling rates.

[4] `http://www.laas.fr/~owe/METROPOLIS/metropolis_eng.html`

Nevertheless, one can expect that monitoring several times a single flow in a "cascade" of tap devices may produce more detailed statistics than a single tap device would.

5.3 Model for sampling & monitoring

In this section, we represent the setup cost of a tap device on a link e by $cost_i(e)$ and the exploitation cost of the same monitoring device $cost_e(e)$. These two cost functions can be general and this will not impact on the following linear program 3. However, the exploitation cost is generally a nondecreasing concave function [22] that allows to take into account the scale factor effect. Note also that the model of [22] is a mixed non linear program, while the one presented in this section is a MILP that can be solved much faster, even though it keeps being non-polynomial.

LINEAR PROGRAM 3 (PPME(H,K)).

$$
\begin{array}{lrll}
\textit{Minimize} & \sum_{e\in E} & (cost_i(e)\cdot x_e + cost_e(e)\cdot r_e) & \\
 & & \textit{set up cost and exploitation cost} & \\
s.t. & \sum_{e\in p} r_e & \geq & \delta_p \quad \forall p \in \mathcal{P} \\
 & x_e & \geq & r_e \quad \forall e \in E \\
 & \sum_{p\in\mathcal{P}_t} \delta_p \cdot v_p & \geq & h_t \cdot \sum_{p\in\mathcal{P}_t} v_p \quad \text{for all traffic } t \\
 & \sum_{p\in\mathcal{P}} \delta_p \cdot v_p & \geq & k \cdot \sum_{p\in\mathcal{P}} v_p \\
 & \delta_p, r_e & \in & [0,1] \quad \forall p \in \mathcal{P}, \forall e \in E \\
 & x_e & \in & \{0,1\} \quad \forall e \in E
\end{array}
$$

As for the linear program 2, the variable x_e reflects the fact that a monitoring device is setup on the link e. The variable δ_p here represents the amount of monitored traffic going through path p. We introduce here the variables r_e that represent the sampling ratio of the monitoring device located on the link e.

We also need to introduce some constraints. The first one is trivial and only models the fact that it is necessary to setup a device on a link if we want to sample traffic on this link. Next constraints ensure that a minimum percentage h_t of each traffic t is monitored and that at least k percent of the total amount of traffic is also monitored.

5.4 Dynamic traffic

Being able to minimize the number of devices under the deployment cost and exploitation cost is possible thanks to the integer linear problems described above. However, these techniques capture static network state while the real traffic inside a POP evolves. A drastic change in the traffic throughput may invalidate all previous optimizations done and will degrade the results the operator will obtain. If it is really not conceivable to migrate a tap device from one link to another one at each traffic fluctuation since it implies human maintenance on each router, it is still possible to consider that the sampling ratio will be adapted to the traffic changes. The problem is thus to find a solution to the problem $PPME(h,k)$ when all x_e are a priori fixed since all devices are already installed. We will call this problem $PPME^*(x,h,k)$.

The $PPME^*(x,h,k)$ problem can be written as the linear program 3 where all x_e are now constants. Thus there is no

more binary variables and it is possible to derive optimal solution in a polynomial time. In fact, it is worthy to note that this problem can be expressed as a minimum cost flow problem for which efficient polynomial time algorithms are available without the need of linear programming anymore.

If an operator has to respect a minimum percent of monitoring h_t per traffic t and at least k percent of the total amount of traffic, we can define a tolerance threshold $T < k$ under which the degradation of monitoring becomes critical and the solution has to be updated. One can therefore derive a simple strategy to maintain the monitoring constraints inside a POP:

1. While $\sum_{p\in\mathcal{P}} \delta_p \cdot v_p \geq T \cdot \sum_{p\in\mathcal{P}} v_p$, wait;
2. When $\sum_{p\in\mathcal{P}} \delta_p \cdot v_p < T \cdot \sum_{p\in\mathcal{P}} v_p$, compute $PPME^*(x, h, k)$, update all sampling rates;
3. Goto 1.

The resolution of the $PPME$ problem can be considered as the initial phase when building the POP. For such an initial phase the time complexity is not really crucial. However, during the life time of a POP, being able to adapt to traffic changes may be important and thus the time complexity becomes a key factor. As mentioned above, the computation of $PPME^*$ is efficient and since it is equivalent to a minimum cost flow computation it does not require a large amount of resources.

6. ACTIVE MONITORING

Active monitoring has received much more attention than passive monitoring in 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 (*i.e.* nodes in charge of the monitoring task and emitting packets) whose probes (*i.e.* the packets emitted by the beacons) cover all the links in the network (see [1, 9] for recent references). When the beacons are chosen, the smallest set of probes has to be computed. Recently, the authors of [15] proposed 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.

The beacon placement problem can be translated into a $0-1$ Integer Linear Programming problem. Assume that Φ is the optimal set of probes obtained with the algorithm of [15]. Each probe $\varphi \in \Phi$ is identified by its two extremities φ_u and φ_v, knowing that the probe from φ_u to φ_v is equal to the probe from φ_v to φ_u. The Integer Linear Programming problem is the following:

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

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

$$and \; \forall \varphi \in \Phi, y_{\varphi_u} + y_{\varphi_v} \geq 1$$

$$\forall i \in V, y_i \in \{0, 1\}$$

where n is the number of nodes in the network and $y = (y_i)_{i\in V}$ is the variable ($y_i = 1$ places a beacon on node i in the network, $y_i = 0$ 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 Φ 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 greatest number of probes first, then remove these probes from 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 4. We have implemented the algorithm of [15] that computes the optimal set of probes. From this set Φ, 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 CPLEX solver. All the results are the average over 20 simulations.

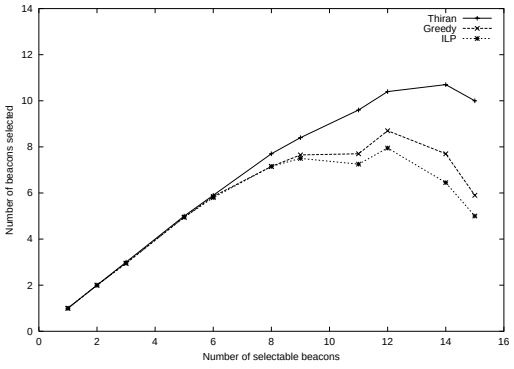

Figure 9: Active monitoring: beacons placement on a 15 routers POP

Figure 9 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, not surprisingly, 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 are 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 for 8 possible beacons they differ only by 1 in the number of located beacons.

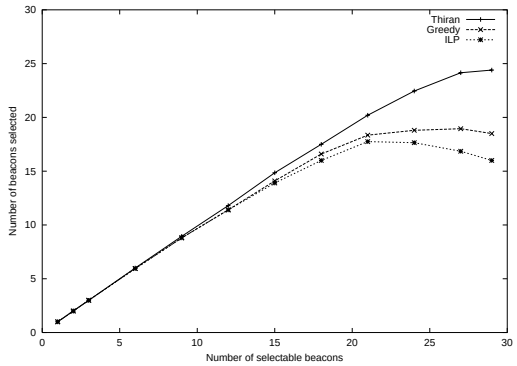

Figure 10: Active monitoring: beacons placement on a 29 routers POP

Figure 10 presents the results for the beacon placement on a POP with 29 routers. They are similar to the results obtained with 15 routers. The ILP solution matches the two greedy solutions and 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 of at most 2 beacons for 15 possible beacons.

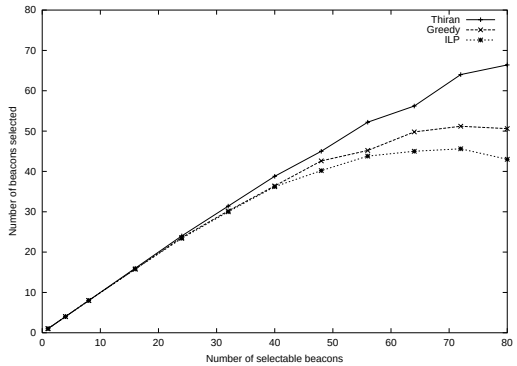

Figure 11: Active monitoring: beacons placement on a 80 routers POP

Figure 11 presents the results for the beacon 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 interesting 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 passive and active monitoring devices. We have provided a powerful combinatorial model of the partial passive monitoring problem in terms of Min Edge Cost Flow, Minimum Set Cover and Minimum Partial Cover.

This model yields a theoretical framework for understanding the combinatorial challenges of measurement point placement. It also permits to develop an efficient mixed integer program, greatly improving on previous formulations given in the literature, and giving rise to an efficient polynomial algorithm for managing dynamic traffic.

The mixed integer programming formulation is flexible enough to easily tackle different problems, or sub-problems, such as computing the best way to position a new set of monitors over an already installed fixed monitoring architecture, to estimate the expected gain in buying one or a set of new devices or the problem of finding the best position for a limited number of devices.

Our approach based on MIP 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 MIP based approach that both outperform 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 three different perspectives. First, the model of sampling capable devices has to be refined in order to get a tighter bound on the actual monitoring ratio achieved by several measurement points on one path. Second, we are considering multi-routing that can arise from load balancing processes in order to get rid of the actual multiplicative impact on the complexity. Third, we are investigating on solutions for measurement campaign, where the operator of a POP or an AS can modify the routing strategy in order to maximize the monitoring ratio, given a set of already installed measurement point. For this last perspective, the flow-based model is expected to apply perfectly. We are also currently testing our solution on larger POPs, with at least 150 routers.

8. REFERENCES

[1] Yigal Bejerano and Rajeev Rastogi. Robust Monitoring of Link Delays and Faults in IP Networks. In *Proceedings of IEEE Infocom*, 2003.

[2] Supratik Bhattacharyya, Christophe Diot, and Jorjeta Jetcheva. POP-Level and Access-Link-Level Traffic Dynamics in a Tier-1 POP. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement (IMW)*, San Francisco, November 2001.

[3] Claude Chaudet, Eric Fleury, and Isabelle Gurin Lassous. Optimal positioning of active and passive monitoring devices. Resdearch Report 5273, INRIA, July 2004.

[4] Nick Duffield. Sampling for passive internet measurement: a review. *Statistical Science*, 19(3), 2004.

[5] Nick Duffield, Carsten Lund, and Mikkel Thorup. Estimating flow distributions from sampled flow statistics. In *Proceedings of the ACM SIGCOMM 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Karlsruhe, Germany, October 2003.

[6] G. Even, G. Kortsarz, and W. Slany. On network design problems: fixed cost flows and the Covering Steiner Problem. *Transactions on Algorithms*, 2004. To be published.

[7] Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, July 1998.

[8] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for internet map discovery. In *Proceedings of IEEE Infocom*. IEEE, 2000.

[9] Joseph D. Horton and Alejandro Lopez-Ortiz. On the Number of Distributed Measurement Points for Network Tomography. In *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement (IMC)*, Miami Beach, USA, October 2003.

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

[11] Sugih Jamin, Cheng Jin, Yixin Jin, Danny Raz, and Lixia Zhang. On the placement of internet instrumentation. In *Proceedings of IEEE Infocom*, Tel Aviv, Israel, March 2000.

[12] Murari Kodialam and T. V. Lakshman. Detecting Network Intrusions via Sampling: A Game Theoretic Approach. In *Proceedings of IEEE Infocom*, San Francisco, USA, March 2003. IEEE.

[13] David Moore, Geoffrey M. Voelker, and Stefan Savage. Inferring Internet Denial of Service Activity. In *Proceedings of the 10th Security Symposium (USENIX Security '01)*, Washington D.C., USA, August 2001.

[14] Tatsuya Mori, Masato Uchida, Ryoichi Kawahara, Jianping Pan, and Shigeki Goto. Identifying elephant flows through periodically sampled packets. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, Taormina, Italy, October 2004.

[15] Hung X. Nguyen and Patrick 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] Andrew K. Paxson, Vern Adams and Matt Mathis. Experiences with NIMI. In *Passive & Active Measurement Workshop (PAM 2000)*, Hamilton, New Zealand, April 2000.

[17] Vern Paxson, Guy Almes, Jamshid Mahdavi, and Mathis Matt. Framework for IP Performance Metrics. RFC 2330, IETF, May 1998.

[18] Jergen Quittek, Tanja Zseby, Benoit Claise, and Sebastian Zander. Requirements for IP Flow Information Export. RFC 3917, IETF, October 2004.

[19] Petr Slavik. A tight analysis of the greedy algorithm for set cover. In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 435–441, New York, NY, USA, 1996. ACM Press.

[20] Petr Slavik. Improved performance of the greedy algorithm for partial cover. *Inf. Process. Lett.*, 64(5):251–254, 1997.

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

[22] Kyoungwon Suh, Yang Guo, Jim Kurose, and Don Towsley. Locating network monitors: complexity, heuristics, and coverage. In *Proceedings of IEEE Infocom*, Miami, USA, March 2005.