

# Active Measurement for Multiple Link Failures Diagnosis in IP Networks

Hung X. Nguyen <sup>\*</sup> and Patrick Thiran

EPFL CH-1015 Lausanne, Switzerland

**Abstract.** Simultaneous link failures are common in IP networks [1]. In this paper, we develop a technique for locating multiple failures in Service Provider or Enterprise IP networks using active measurement. We propose a two-phased approach that minimizes both the additional traffic due to probe messages and the measurement infrastructure costs. In the first phase, using elements from max-plus algebra theory, we show that the optimal set of probes can be determined in polynomial time, and we provide an algorithm to find this optimal set of probes. In the second phase, given the optimal set of probes, we compute the location of a minimal set of measurement points (beacons) that can generate these probes. We show that the beacon placement problem is NP-hard and propose a constant factor approximation algorithm for this problem. We then apply our algorithms to existing ISP networks using topologies inferred by the Rocketfuel tool [2]. We study in particular the difference between the number of probes and beacons that are required for multiple and single failure(s) diagnosis.

## 1 Introduction

Routing decisions and content distribution require proper connectivity and latency information to direct traffic in an optimal fashion. The family of Internet protocols collect and distribute only a limited amount of information on the topology, connectivity and state of the network. Hence information of interest for Internet Service Providers (ISPs), such as link delays or link failures, has to be inferred from experimental measurements. The strategy to obtain network information through end-to-end measurements, known as Internet tomography, is therefore of great interest to the research community [2–7]. The majority of work on network tomography concentrates on either topology discovery (e.g. [2–4]), or link delay monitoring (e.g. [5]). Some recent research showed that active measurements can also be used to pinpoint failures in IP networks [6, 7].

In general, 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*. To detect failures, the path that each

---

<sup>\*</sup> Hung X. Nguyen’s work is financially supported by grant DICS 1830 of the Hasler Foundation, Bern, Switzerland.

probe actually follows is compared with the path that the probe should follow according to the current network topology information. If the two paths are different, at least one link in the path determined by the current topology has failed [7]. The authors in [6,7] study the problem of detecting and locating failures under the assumption that at most one failure can happen at a time. Under this assumption, a set of probes that traverse all links in the network is sufficient to diagnose any such single failure. The objective in [6,7] is to determine the smallest set of beacons whose probes cover all links in the network. This problem is proven to be NP-complete and approximation algorithms are given in [6] for general topologies and in [7] for the current Internet topology. Once the beacons are located, the smallest set of probes must still be determined, Bejerano et. al. [6] show that this problem is also NP-complete.

Multiple link failures in IP networks happen much more frequently than one might expect; [1] has recently reported that nearly 35% of link failures in the Sprint backbone are multiple failures. This figure emphasizes the need for a failure detection and location scheme that takes into account the existence of multiple failures. Although the active probing scheme in [6] can work under the presence of several link failures, it cannot detect and locate *simultaneous* link failures. To our knowledge, there are no active monitoring techniques to date that apply to simultaneous failures in IP networks.

In this work, we are interested in using active probing to detect and locate link failures, under the assumption that several links can fail at one time. To achieve this goal, a distributed set of beacons running a special software are deployed at key sites across the entire network. A beacon only needs to send probes to other nodes, and see what routes the probes take. Probe messages can be implemented by tools like traceroute [8] or skitter [9]. Using probes to pinpoint network faults has several advantages over monitoring routing protocol messages (e.g., OSPF LSAs), or using SNMP traps to identify failed links. Firstly, probe-based techniques are routing protocol agnostic; as a result, they can be used with a wide range of protocols like OSPF, IS-IS, RIP, etc. Secondly, SNMP trap messages may be unreliable because they use UDP as the transport protocol [6]. Note here that by using active probing, we may not be able to detect and locate all failures (single or multiple) uniquely. This is especially true when we consider multiple failures or when there are constraints on the set of nodes where beacons can be deployed. Even a probing scheme of maximal detection capacity, that is, a scheme that would use all the available beacon nodes and send probes from these beacons to all nodes in the network would only detect and locate a subset of all possible failures. Therefore, instead of looking for a probing scheme that guarantees the detection and location of every failure of a given multiplicity, we find a probing scheme that can detect and locate every failure that the probing scheme with maximal detection capacity can detect and locate.

The cost of using probes for fault diagnosis comprises two components: the additional traffic due to probe packets and the infrastructure cost for beacons. Similarly to [6], we use a two-phased approach to minimize both the number of probes and the number of beacons. Whereas, unlike [6], we first minimize the

number of probes and next the number of beacons. This enables us to use results from max-plus algebra for the probe selection problem. Our main contributions are as follows. (i) We show that, contrary to the single failure case [6] and surprisingly so, the optimal set of probes for multiple failures can be found in *polynomial time*. (ii) However, like the single failure case, we show that the beacon placement problem in the multiple failure case is NP-hard. We provide a constant factor approximation algorithm for the beacon placement problem, which is very close to the best possible bound. (iii) We show that our algorithms perform well on existing networks, and that there is a substantial difference between the number of probes and beacons that are required for multiple failures diagnosis and for single failure diagnosis.

The remainder of this paper is organized as follows. Section 2 introduces the network model. Section 3 presents the probe selection problem, and Section 4 describes the beacon placement problem. Section 5 contains experiment studies of our algorithms on existing ISP networks. Finally, we conclude the paper in Section 6.

## 2 Network Model

We model the network as an undirected graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ , where the graph nodes,  $\mathcal{V}$ , denote the network elements and the edges,  $\mathcal{E}$ , represent the communication links connecting them. The number of nodes and edges is denoted by  $n = |\mathcal{V}|$  and  $e = |\mathcal{E}|$ , respectively. Further, we use  $P_{s,t}$  to denote the path traversed by an IP packet from a source node  $s$  to a destination node  $t$ . If there is no failure in the network, an IP packet that is sent from a node  $s$  to a destination  $t$  will follow the path  $P_{s,t}$ . When there is/are failure(s) of links on the path  $P_{s,t}$ , the probe has to be rerouted around the failed link(s); Therefore the actual path that the probe takes will be different from  $P_{s,t}$ . By comparing the actual path that the probe from  $s$  to  $t$  takes and  $P_{s,t}$ , we can detect if any link in the path  $P_{s,t}$  has failed or not. When a probe detects link failure(s) in its path, we say that the probe has *failed*.

For a known topology  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  and a set of probes  $\mathcal{P}$ , we can compute which probes will fail when a network link goes down. We call the above relations between network links and probes *dependency relations*. Dependency relations of a network can be represented by a *dependency matrix*  $D$  of dimension  $e \times n_p$ , where  $e$  is the number of links and  $n_p = |\mathcal{P}|$  is the number of probes in the network.  $D$  is constructed as follows. Let  $P_{s,t}$  be the path followed by probe  $p_i$  in the normal situation without failures. Then the entry  $D(i, j) = 1$  if the path  $P_{s,t}$  contains the link  $e_j$  and  $D(i, j) = 0$  otherwise. A row of  $D$  therefore corresponds to a probe (more precisely, to the path that the probe take), whereas a column corresponds to a link.

### 3 Probe selection problem

Given a network  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with a set of beacons  $\mathcal{V}_B$ , we denote by  $\mathcal{P}_{max}$  the set of probes generated when each beacon sends probes to all nodes in the network.  $\mathcal{P}_{max}$  represents an upper bound on the probing capability of the network. It is the largest set of probes that can be sent to different destinations in the network. Let  $D_{max}$  denote the dependency matrix when the set of probes is  $\mathcal{P}_{max}$ .

Let  $\mathcal{S} \subseteq \mathcal{E}$  be a set of links. The set of failed probes triggered by the failure of all links in  $\mathcal{S}$  is made of all the failed probes that would be triggered by the individual failure of each link in  $\mathcal{S}$ . The failure of links in  $\mathcal{S}$  can thus be represented by a vector  $\vec{d}^{\mathcal{S}}$  that is obtained by OR-ing all the column vectors of  $D_{max}$  representing individual links in  $\mathcal{S}$ . Let us define the failure of a set of network links as the failure of all links in that set. A probe  $p_k$  is said to distinguish the failures of two subsets  $\mathcal{E}_1, \mathcal{E}_2 \subseteq \mathcal{E}$  if and only if the corresponding  $k$ th entries in  $\vec{d}^{\mathcal{E}_1}$  and  $\vec{d}^{\mathcal{E}_2}$  are different, i.e.,  $d^{\mathcal{E}_1}(k) \neq d^{\mathcal{E}_2}(k)$ . The probe set  $\mathcal{P}$  is said to distinguish the failures of two subsets  $\mathcal{E}_1, \mathcal{E}_2 \subseteq \mathcal{E}$  if and only if there exists a probe  $p \in \mathcal{P}$  such that  $p$  distinguishes  $\mathcal{E}_1, \mathcal{E}_2$ . We are interested in the following optimization problem.

**Definition 1. [PS problem]** *The probe selection problem is the determination of the smallest subset  $\mathcal{P}^*$  of  $\mathcal{P}_{max}$ , such that any two subsets of  $\mathcal{E}$  whose failures are distinguished by  $\mathcal{P}_{max}$  are also distinguished by  $\mathcal{P}^*$ .*

Let  $D^*$  be the dependency matrix for the system with the set of probes  $\mathcal{P}^*$ . In terms of dependency matrices, the probe selection problem amounts to removing some rows of  $D_{max}$  in order to obtain a new matrix  $D^*$  that verifies the following properties: (i)  $D_{max}$  and  $D^*$  have the same number of columns, and (ii) whenever two vectors obtained by OR-ing up to  $e = |\mathcal{E}|$  columns of  $D_{max}$  are different, the two vectors obtained by OR-ing the same columns in  $D^*$  are also different.

The set of all binary vectors that represent single and multiple link failures of a network has a special property that the OR-ing of any two vectors is also a vector in the set. Any set of vectors with the above property is called a vector span [10]. To solve the probe selection problem, we need to employ some special properties of a vector span. We, therefore, first study in Section 3.1 properties of general vector spans, and then show how these properties can be applied to develop an algorithm for the probes selection problem in Section 3.2.

#### 3.1 Mathematical basis

Let  $\mathcal{D} = \{\vec{d}_i\}_{1 \leq i \leq h}$  be a set of binary vectors of equal length, and let  $I = \{1, \dots, h\}$  be the index set of  $\mathcal{D}$ . A vector span  $\mathcal{S}$  can be defined on  $\mathcal{D}$  as follows.

**Definition 2. [Vector span]** *The vector span of  $\mathcal{D}$  is*

$$\mathcal{S} = \langle \mathcal{D} \rangle = \left\{ \bigvee_{i \in I} \alpha_i \cdot \vec{d}_i \mid \alpha_i \in \{0, 1\}, \vec{d}_i \in \mathcal{D} \right\}$$

where “ $\vee$ ” denotes the binary max operation, and “ $\cdot$ ” denotes the usual multiplication operation. Vectors in  $\mathcal{D}$  are called the *generator vectors* of  $\mathcal{S}$ .

On the set  $\mathcal{D}$ , we define the following independence property [10].

**Definition 3. [IP]** *The set  $\mathcal{D} = \{\vec{d}_i\}_{i \in I}$  is independent if for all  $i \in I$  and  $I_2 \subseteq I \setminus \{i\}$ ,  $\vec{d}_i \notin \langle \{\vec{d}_j\}_{j \in I_2} \rangle$ .*

Merging Definition 2 and Definition 3, we obtain the following definition.

**Definition 4. [Basis]** *A basis  $\mathcal{B}$  of a span  $\mathcal{S}$  is a set of independent vectors of  $\mathcal{S}$  such that  $\langle \mathcal{B} \rangle = \mathcal{S}$ .*

Assume we have a span  $\mathcal{S}$  that is generated by a set of generator vectors  $\mathcal{D}$  and has a basis  $\mathcal{B}$ . The following lemma follows from Definition 4, and is needed to solve the probe selection problem.

**Lemma 1.** *If  $\mathcal{D}$  is finite, then  $\mathcal{S}$  has a unique basis  $\mathcal{B}$  that is the subset of  $\mathcal{D}$  with smallest cardinality such that  $\langle \mathcal{B} \rangle = \mathcal{S}$ .*

*Proof.* Wagner [10] proved that spans over general vector sets have a unique basis, and hence this conclusion is also true for spans over binary vectors.

We prove the second assertion of the lemma by contradiction. Assume that there is a smaller subset of  $\mathcal{D}$ , namely  $\mathcal{B}'$ , which satisfies  $\langle \mathcal{B}' \rangle = \mathcal{S}$ , i.e., that there exists at least one vector  $\vec{v}$  of  $\mathcal{B}$  that does not belong to  $\mathcal{B}'$ . Let us denote by  $I_{\mathcal{B}}$  the index set of  $\mathcal{B}$ , and by  $I'_{\mathcal{B}}$  the index set of  $\mathcal{B}'$ . Let  $\vec{d}_i$ ,  $i \in I_{\mathcal{B}}$  be the vectors of  $\mathcal{B}$ . Since  $\langle \mathcal{B}' \rangle = \mathcal{S}$ , there exists a non empty subset  $I'_v \subseteq I'_{\mathcal{B}}$  such that:

$$\vec{v} = \bigvee_{i \in I'_v} \vec{d}_i. \quad (1)$$

Furthermore, since  $\mathcal{B}$  is also a basis of  $\mathcal{S}$ , for each  $\vec{d}_i \in \mathcal{B}'$  there is a nonempty subset  $I_i \subseteq I_{\mathcal{B}}$  such that:

$$\vec{d}_i = \bigvee_{j \in I_i} \vec{d}_j. \quad (2)$$

Substituting (2) in (1) yields  $\vec{v} = \bigvee_{j \in I_v} \vec{d}_j$ , where  $I_v = \bigcup_{i \in I'_v} I_i$ . Since  $\mathcal{B}$  is independent, the only case where this can happen is that there exists an index  $k \in I_v$  such that:  $\vec{v} = \vec{d}_k$  and  $\vec{d}_k \vee \vec{d}_l = \vec{d}_k$  for all  $l \in I_v \setminus \{k\}$ . From (2), this implies that there exists  $i \in I'_v$  such that  $\vec{d}_i = \vec{v}$ , which in turn indicates that  $\vec{v} \in \mathcal{B}'$ ; a contradiction to the assumption  $\vec{v} \notin \mathcal{B}'$ .  $\square$

### 3.2 Probe selection algorithm

Denote by  $C(D_{max})$  and  $R(D_{max})$  the set of column vectors and row vectors of the matrix  $D_{max}$ . Let  $\langle C(D_{max}) \rangle$  be the span generated by column vectors, called column span of  $D_{max}$ , and let  $\langle R(D_{max}) \rangle$  be the span generated by row vectors, called row span of  $D_{max}$ . A vector in  $\langle C(D_{max}) \rangle$  represents subsets of

$\mathcal{E}$  whose failures generate the same set of failed probes. We call the set of all the subsets of  $\mathcal{E}$  whose failures are represented by the same vector a *failure set*. Two different vectors in  $\langle C(D_{max}) \rangle$  represent two failure sets that are distinguished by  $\mathcal{P}_{max}$ . Therefore, its cardinality  $|\langle C(D_{max}) \rangle|$  is the number of failure sets that can be distinguished by  $\mathcal{P}_{max}$ . Similarly, let  $D^*$  be the dependency matrix of the system with the set of probes  $\mathcal{P}^*$ . Let  $R(D^*)$  and  $C(D^*)$  be respectively the set of column vectors and row vectors of  $D^*$ .  $|\langle C(D^*) \rangle|$  is the number of failure sets that can be distinguished by  $\mathcal{P}^*$ .

Since  $\mathcal{P}^*$  is a subset of  $\mathcal{P}_{max}$ , the number of failure sets that can be distinguished by  $\mathcal{P}^*$  is always less than or equal to the number of failure sets that can be distinguished by  $\mathcal{P}_{max}$ . Furthermore, any two subsets of  $\mathcal{E}$  that can be distinguished by  $\mathcal{P}^*$  can also be distinguished by  $\mathcal{P}_{max}$ . Thus,  $\mathcal{P}^*$  distinguishes any two subsets of  $\mathcal{E}$  that  $\mathcal{P}_{max}$  distinguishes if and only if the number of failure sets that are distinguishable by  $\mathcal{P}^*$  is equal to the number of failure sets that are distinguishable by  $\mathcal{P}_{max}$ . Consequently, the probe selection problem amounts to find  $\mathcal{P}^*$  such that the number of failure sets that can be distinguished by  $\mathcal{P}^*$  and by  $\mathcal{P}_{max}$  are equal. Since each failure set is respectively represented by a column of  $D^*$  or  $D_{max}$ , the solution of the probe selection problem is the smallest subset  $R(D^*)$  of  $R(D_{max})$  such that  $|\langle C(D^*) \rangle| = |\langle C(D_{max}) \rangle|$ . Theorem 1 below gives the solution to the probe selection problem.

**Theorem 1.** *The solution to the probe selection problem is the set of probes whose corresponding rows in  $D_{max}$  form the basis of  $\langle R(D_{max}) \rangle$ .*

*Proof.* Let  $D^*$  be a matrix whose rows are the basis of the span  $\langle R(D_{max}) \rangle$ , i.e., such that  $\langle R(D^*) \rangle = \langle R(D_{max}) \rangle$ . From [11], Theorem 1.2.3, the row span and column span of any binary matrix have the same cardinality. Therefore,  $|\langle C(D^*) \rangle| = |\langle R(D^*) \rangle| = |\langle R(D_{max}) \rangle| = |\langle C(D_{max}) \rangle|$ , which yields that  $R(D^*)$  is a solution for the probe selection problem. Now, Lemma 1 yields that  $R(D^*)$  is the smallest subset of  $\langle R(D_{max}) \rangle$  such that  $|\langle R(D^*) \rangle| = |\langle R(D_{max}) \rangle|$ . Therefore,  $R(D^*)$  is the unique solution to the probe selection problem.  $\square$

We now give an algorithm, which we call the the **Probe Selection (PS)** algorithm, that finds the basis of  $\langle R(D_{max}) \rangle$ . The *weight* of a vector is defined as the number of 1 entries in that vector. Let us denote the elements of  $R(D_{max})$  by  $\{\vec{r}_1, \vec{r}_2, \dots, \vec{r}_{n_{\mathcal{P}_{max}}}\}$ , where  $n_{\mathcal{P}_{max}} = |\mathcal{P}_{max}|$ . The PS algorithm constructs the set  $R(D^*)$  as follows.

**The PS algorithm**

- Step 1:* Initialize  $R(D^*)$  to an empty set, and set an integer  $i$  to 1;
- Step 2:* Sort and re-index the vectors in  $R(D_{max})$  in increasing weight order.
- Step 3:* Until  $R(D_{max}) \neq \emptyset$  repeat the loop: remove  $\vec{r}_i$  from  $R(D_{max})$ ; increase  $i$  by 1; if  $\vec{r}_i \notin \langle R(D^*) \rangle$ , then append  $\vec{r}_i$  to  $R(D^*)$ .
- Step 4:* return  $R(D^*)$ .

**Theorem 2.** *The set of vectors  $R(D^*)$  returned by the PS algorithm is the basis of  $\langle R(D_{max}) \rangle$ .*

*Proof.* First, we prove that the vectors in  $R(D^*)$  are independent. Let us re-index the vectors  $\vec{r}_i$  in  $R(D^*)$  in the order of inclusion to  $R(D^*)$  by the PS algorithm. Step 3 of the PS algorithm prevents any  $\vec{r}_i \in R(D^*)$  to be obtained by OR-ing any combination of vectors in  $\{\vec{r}_{i-1}, \vec{r}_{i-2}, \dots, \vec{r}_1\}$ . Furthermore,  $\vec{r}_i$  is not an OR-ing of any combination of vectors in  $\{\vec{r}_{i+1}, \dots, \vec{r}_{|R(D^*)|}\}$ . Indeed, if this was true, then the weight of  $\vec{r}_i$  would be larger than the weight of these vectors, which is impossible because by construction the weight of  $\vec{r}_i$  is smaller than or equal to the weight of  $\vec{r}_j$  for all  $j > i$ . From the above results, any vector  $\vec{r}_i \in R(D^*)$  is not an OR-ing of any combination of other vectors in  $R(D^*)$ , hence the set  $R(D^*)$  is independent.

Second, we prove that  $\langle R(D^*) \rangle = \langle R(D_{max}) \rangle$ . Indeed, by construction, any vector in the set  $\{\vec{r}_1, \dots, \vec{r}_{n_{P_{max}}}\}$  either belongs to  $R(D^*)$ , or is an OR-ing of some combinations of vectors in  $R(D^*)$ .  $\square$

Note here that for any two nodes  $s$  and  $t$ , if  $P_{t,s}$  and  $P_{s,t}$  contain the same links, then the two row vectors that represent the probe from  $s$  to  $t$ , and the probe from  $t$  to  $s$  are equal. Furthermore, the vector that represents the path  $P_{t,s}$  can only be generated by probes from at most two beacons (at  $s$  and  $t$ ).

## 4 Beacon placement problem

We have shown in Section 3 that given a set of beacons  $\mathcal{V}_B$ , there is an optimal set of row vectors  $R(D^*)$  of the dependency matrix that the beacons need to generate for multiple failure diagnosis. Once the optimal set of row vectors  $R(D^*)$  is determined, to generate these vectors we may not need all the available beacons. We are now interested in finding the minimal number of beacons needed to generate the probes corresponding to the vectors in  $R(D^*)$ ; this is the beacon placement problem defined as follows.

**Definition 5.** [*BP problem*] Given a network  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , with  $\mathcal{V}_B$  possible positions where we can place the beacons, the beacon placement (BP) problem is the determination of the smallest subset  $\mathcal{V}^*$  of  $\mathcal{V}_B$  such that any two sets of links  $\mathcal{E}_1$  and  $\mathcal{E}_2$  that are distinguished by probes of  $\mathcal{V}_B$  are also distinguished by probes of  $\mathcal{V}^*$ .

We solve the beacon placement problem as follows. We first assume that all nodes in  $\mathcal{V}_B$  are used as beacons and find the optimal set of row vectors  $R(D^*)$  for this setting using the PS algorithm described in Section 3. We then look for the smallest subset  $\mathcal{V}^*$  of  $\mathcal{V}_B$  such that the set of vectors corresponding to probes from nodes in  $\mathcal{V}^*$  contains  $R(D^*)$ .

### 4.1 Hardness of the beacon placement problem

**Theorem 3.** *The beacon placement problem is NP-hard.*

*Proof.* The proof for this theorem is straightforward when we realize that in a network that uses shortest path routing with all links have equal weight, and with  $\mathcal{V}_B = \mathcal{V}$ , the BP problem reduces to the well-known vertex cover problem [12].  $\square$

## 4.2 A greedy algorithm for the beacon placement problem

We now present a greedy beacon placement (BP) algorithm, which constructs  $\mathcal{V}^*$ , given  $R(D^*)$ . Let  $\mathcal{V}_{approx}$  be the set of beacons returned by the BP algorithm, and let  $\mathcal{A}$  and  $\mathcal{C}$  be two subsets of  $R(D^*)$ . For a beacon  $u$ , let us denote by  $\mathcal{R}(u)$  the set of all vectors representing probes that can be generated from  $u$ . The steps of the BP algorithm are as follows.

### The BP algorithm

*Step 1:* Initialize  $\mathcal{V}_{approx}$  and  $\mathcal{A}$  to empty sets,  $\mathcal{C}$  to  $R(D^*)$ .

*Step 2:* Until  $\mathcal{C} \neq \emptyset$  repeat the loop: pick a vector  $\vec{r} \in \mathcal{C}$  and append  $\vec{r}$  to  $\mathcal{A}$ ; if for all nodes  $u$  in  $\mathcal{V}_B$ ,  $\vec{r} \in \mathcal{R}(u)$ , then include  $u$  in  $\mathcal{V}_{approx}$  and remove all the vectors in  $\mathcal{R}(u)$  from  $\mathcal{C}$ .

*Step 3:* Return  $\mathcal{V}_{approx}$ .

The BP algorithm is said to have an approximation ratio  $\rho(n)$  if for every graph  $\mathcal{G}$  with  $n$  nodes,  $|\mathcal{V}_{approx}|/|\mathcal{V}^*| \leq \rho(n)$ .

**Theorem 4.** *The BP algorithm has a constant approximation factor of 2.*

*Proof.* From Step 2 of the algorithm, for each vector in  $\mathcal{A}$ , we add one or two nodes (as explained at the end of Section 3.2, there is a maximum of two beacons that can send probes taking the same path and hence are represented by the same vector) to  $\mathcal{V}_{approx}$ , hence  $|\mathcal{V}_{approx}| \leq 2 \cdot |\mathcal{A}|$  (\*).

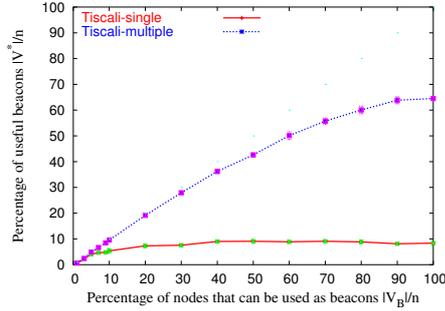
Any solution must include, for every vector in  $\mathcal{A}$ , at least one beacon that can generate it. Furthermore, by construction, no two vectors in  $\mathcal{A}$  can be generated by the same beacon. So the optimal beacon selection  $\mathcal{V}^*$  is of size  $|\mathcal{V}^*| \geq |\mathcal{A}|$  (\*\*).

Combining (\*) and (\*\*), we get  $|\mathcal{V}_{approx}| \leq 2 \cdot |\mathcal{V}^*|$ . □

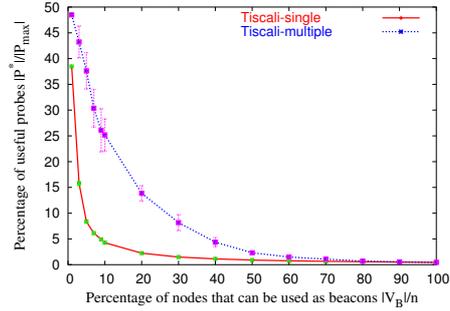
## 5 Experimental study

We apply our algorithms on ISP topologies that are inferred by the Rocketfuel tool [2]. We investigate the number of beacons and the number of probes required for multiple failures diagnosis by our algorithms to that needed for single failure diagnosis by algorithms in [6] on three backbone ISP topologies with sizes ranging from small (Exodus: 80 nodes and 147 links) to medium (Telstra: 115 nodes and 153 links), and large (Tiscali: 164 nodes and 328 links). For the sake of simplicity, we assume that all the ISPs use shortest path routing to route traffic.

Recall that  $n$  is the number of nodes in the network. In our experiments, the number of nodes that can be used as beacons (beacon candidates)  $|\mathcal{V}_B|$  is varied from  $n/100$  to  $n$ . We select the beacon candidates randomly by picking a random permutation of the set of nodes in the network. After building the dependency matrix as in Section 3, we run the PS algorithm to find the optimal set of probes and then the BP algorithm to find the set of beacons that are required to generate these probes. We also run the algorithms described in [6] to find the set of beacons and the set of probes for single fault diagnosis. For space constraints, we only present in this paper the results for the Tiscali topology.



**Fig. 1.** The number of beacons for single and multiple failure(s) diagnosis.



**Fig. 2.** The number of probes for single and multiple failure(s) diagnosis.

Similar results are obtained for the two other topologies. In Fig. 1, we plot the percentage of nodes that are actually used as beacons for multiple failures diagnosis and single failure diagnosis. We also plot the percentage of useful probes returned by the PS algorithm and the single fault algorithm [6] in Fig. 2.

The number of useful beacons  $|\mathcal{V}^*|$  for multiple failures diagnosis is notably smaller than the number of possible beacon nodes for all sizes of the set  $|\mathcal{V}_B|$  of beacon candidates. This is especially true when the number of beacon candidates is large. We also observe a fast increase in the number of beacons useful for multiple failure diagnosis when the number of beacon candidates increases. This can be explained by the increase in number of multiple failures that can be distinguished by the beacons. However, for single fault diagnosis, the number of beacons increases very slowly, as probes from even a small set of beacons are enough to detect and locate all single failures.

We observe that for all sizes of the set of beacon candidates, the number of useful probes  $|\mathcal{P}^*|$  is less than a half of the total number of probes  $|\mathcal{P}_{max}|$  that can be sent in the network. The percentage of useful probes also decreases rapidly as the number of beacons increases for both multiple and single failure(s) cases. This observation can be explained for both the single and multiple failure cases as follows. In the single failure case, the number of useful probes remains relatively unchanged (approximately equal to the number of links) for various numbers of beacon candidates. Hence, when the total number of probes that can be sent in the network increases with the number of beacon candidates, the percentage of useful probes decreases. In the multiple fault case, the number of useful probes decreases as the number of beacon candidates increases, because these additional beacon candidates make it possible to replace a number of different probes by a single probe that distinguishes the same failure sets. There remains however a large difference in the number of probes for multiple and single failure diagnosis in Fig. 2.

## 6 Conclusions

In this paper, we have investigated the use of active probing for link failure diagnosis in ISP networks. We have shown that for multiple failures diagnosis the optimal set of probes can be found in polynomial time. On the contrary, the problem of optimizing the number of beacons for multiple failures diagnosis is NP-hard. We have studied the performance of our algorithms on existing ISP topologies. Our studies show that there is a great reduction in the number of probes and beacons that are available and the number of probes and beacons that are actually useful for multiple failures diagnosis. However, there are also remarkable differences in the number of probe and the number of beacons for single and multiple fault diagnosis in all analyzed ISP topologies. These figures can be used by ISP operators to trade off accuracy of diagnosis for probing costs.

We are working on various extensions of the present work. We obtained in this paper the optimal set of useful probes. The next step is to determine the failure detection capacity of the optimal set of probes, that is, to find the number of failures of a given multiplicity (double, triple, etc.) that can be detected and located by this probing scheme. We are also investigating how an active probing scheme can cope with missing and false probes. The challenge is to improve the robustness of the probing scheme to probe errors without much increasing in the probing costs.

## References

1. A.Markopoulou, G.Iannaccone, S.Bhattacharyya, C.N.Chuah, C.Diot: Characterization of Failures in an IP Backbone. Proc. IEEE INFOCOM'04 (2004)
2. N.Spring, R. Mahajan, D.Wetherall: Measuring ISP topologies with Rocketfuel. Proc. of ACM SIGCOMM (2002)
3. G.Govindan, H.Tangmunarunkit: Heuristics for internet map discovery. Proc. IEEE INFOCOM'00 (2000)
4. S.Jamin, C.Jin, Y.Jin, D.Raz, L.Zhang: On the placement of internet instrumentation. Proc. IEEE INFOCOM'00 (2000)
5. M.Adler, T.Bu, R.K.Sitaraman, D.Towsley: Tree layout for internal network characterizations in multicast networks. Networked Group Comm. (2001) 189–204
6. Y.Bejerano, R.Rastogi: Robust monitoring of link delays and faults in IP networks. Proc. IEEE INFOCOM'03 (2003)
7. J.Horton, A.Lopez-Ortiz: On the number of distributed measurement points for network tomography. IMC'03 (2003)
8. S.W.Richard: TCP/IP illustrated. Addison-Wesley Publishing Company (1994)
9. CAIDA: Cooperative Association for Internet Data Analysis. (<http://www.caida.org/Tools/Skitter>)
10. E.Wagneur: Moduloids and pseudomodules: 1 dimension theory. Discrete Mathematics **98** (1991) 57–73
11. J.Kim : Boolean Matrix theory and applications. Marcel Dekker (1982)
12. M.R.Garey, D.S.Johnson: Computers and Intractability: A guide to the theory of NP-completeness. W. H. Freeman (1979)