Practical Passive Lossy Link Inference

Alexandros Batsakis, Tanu Malik, and Andreas Terzis

Computer Science Department
Johns Hopkins University
abat@cs.jhu.edu, tmalik@cs.jhu.edu, terzis@cs.jhu.edu

Abstract. We propose a practical technique for the identification of lossy network links from end-to-end measurements. Our scheme is based on a function that computes the likelihood of each link to be lossy. This function depends on the number of times a link appears in lossy paths and on the relative loss rates of these paths. Preliminary simulation results show that our algorithm achieves accuracy higher than previously proposed heuristic methods and comparable to statistical methods at significantly lower running time.

1 Introduction

Loss inference techniques [1–3] attempt to infer link loss rates from end-to-end measurements. *Active* techniques [1] infer link loss by actively probing the network, while *passive* techniques [2, 3] estimate packet loss by observing the evolution of application traffic. Depending on the method used to infer packet loss, passive techniques can be further divided to *analytical* [2] and *heuristic* [3]. Analytical techniques detect more lossy links while techniques are more efficient.

The insight behind our approach is that users are interested in finding lossy links affecting the performance of their applications, rather than finding the exact loss rate of these links. Based on this insight, we present COBALT, a heuristics-based inference algorithm that detects with high probability the lossy links affecting applications' performance. COBALT assigns a *confidence level* to each link—the higher the confidence, the higher the probability that the link is lossy. The confidence level of a link depends on the number of lossy paths the link belongs to and how lossy are those paths compared to all the paths in the network.

2 Algorithm

We start by presenting the network model we use. In our model clients are connected to servers through a network whose topology is known a-priori. Clients exchange data with the servers using any TCP-based protocol. Using traces collected from the servers, we calculate the loss rate of the path between the server and the each client as the ratio of the retransmitted packets to the total number of packets sent by the server.

COBALT starts by separating good from bad paths. Paths with loss rate higher than a threshold T are labeled as *bad* while the remaining paths are labeled as *good*. The threshold T corresponds to a loss rate above which application performance is disrupted. Subsequently, links are categorized depending on the number of good paths they belong to. This approach is based on the intuition that lossy links dominate the end-to-end path loss rate. If a path contains a lossy link then the path's loss rate will be at least equal to the link's loss rate. Thus, a lossy link cannot be part of a good path. To make COBALT less susceptible to path loss rate estimation errors, we classify a link as good (non-lossy) only if it belongs to at least s good paths. The parameter s, defined as the sensitivity of the algorithm, depends on number of paths in the network. Higher values of s give higher confidence that the identified links are truly lossy. At the same time, the number of false positives can increase because some good links will be classified as lossy if they don't participate in s paths.

After excluding the links found in good paths, COBALT computes the confidence levels of the remaining links. The confidence level $cfd(l)$ for a link l in a network N is computed as:

$$cfd(l) = K^{t(l)} \cdot avl(l) \cdot avp^{-1}(N) \tag{1}$$

In Eq.1, $avl(l)$ is the average loss rate of all paths that l belongs to, while $avp(N)$ is the average loss rate among all bad paths in the network. $t(l)$ denotes the number of times l is found in lossy paths and finally $K \geq 1$ is a constant. Intuitively, a link is bad if it participates in paths whose loss rate is much higher than the average loss rate of all network paths. This effect is covered by the fraction in Equation 1. Second, we can have higher confidence that a link is bad if it belongs to many bad paths. This second effect is covered by the $K^{t(l)}$ term in Equation 1. The greater the value of K the higher the importance of $t(l)$. We use an exponential function of $t(l)$ so small differences in the number of lossy paths a link belongs to will create large difference in confidence level simplifying the final selection of the most problematic links. More details on how K should be selected will be discussed in future work.

As its last step, COBALT ranks the links by their confidence levels. The links with the highest confidence levels are the most likely to be problematic.

2.1 Real-time Extension

Existing loss inference techniques cannot detect transient lossy links existing in the Internet today [4]. To address this limitation, we extend the basic algorithm presented above to work incrementally over shorter timescales. This online algorithm works similarly to its offline variant but uses an exponential moving average formula to compute the confidence level of a link l:

$$cfd_{t_{i+1}}(l) = (1 - w) \cdot cfd_{t_{i-1}}(l) + w \cdot cfd_{t_i}(l) \tag{2}$$

where $cfd_{t_{i-1}}(l)$ the previous confidence level and $cfd_{t_i}(l)$ the confidence level computed by the most recent data. w is the aging constant controlling

the convergence time of the algorithm. An interesting point in our method is that the value of w might not be the same across all links or even for estimates made for the same link. Its exact value is a function of two parameters: **(a)** The interval $\Delta t = t_i - t_{i-1}$. If Δt is large, the significance of $cfd_{t_{i-1}}(l)$ decreases, as it reflects an obsolete network view. Hence, in this case w should be close to one. **(b)** The number of packets ΔP received in Δt. Since our method is based on statistics, the larger the sample the more confident we are about the outcome of our analysis. Therefore, as ΔP increases, w should approach one.

3 Simulation Results

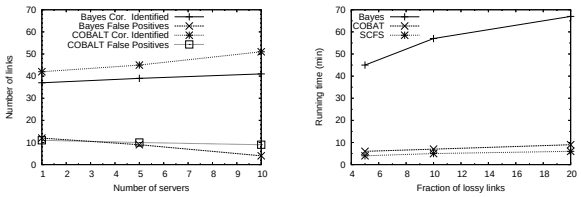
We used ns-2 to simulate our network of clients and servers and The simulated network was created using BRITE's two-level hierarchical topologies [5]. The network consists of 800 nodes and about 1400 links. We randomly chose 100 clients out of a pool of 250 to download a large file from the server using HTTP. We also picked a fraction f of the links to be lossy. Good links, have loss rates between $0 - 0.5\%$, while bad links have loss rate between $1.0 - 3.0\%$.

We compare COBALT to three other methods: Random Sampling [2], Bayesian with Gibbs Sampling [2] and the SCFS algorithm [3]. Given the difference of COBALT to previous techniques we need to redefine coverage and false positives in terms of the algorithm's parameters. In our evaluation, a correctly identified lossy link is one whose confidence level exceeds a threshold T_{lossy}. Consequently, non-lossy links whose confidence level exceed this threshold count as false positives. We set $T_{lossy} = K$ in our experiments. We ran each algorithm three times, each time with a different topology. The reported confidence level is the average of the confidence levels obtained over the three executions. We choose K to be $3/2$, the sensitivity $s = 3$, while $T = 0.01$. For random sampling, the mean link loss rate is chosen over 500 iterations. If the mean exceeds the loss rate threshold of bad links, the link is said to be lossy. For the Bayesian method, the "burn in" period in Gibbs sampling is 1000 iterations, and links are marked lossy if 99% of the samples found are above the loss rate threshold.

Table 1. Comparison of the Random(R),Bayesian(B),SCFS(S),COBALT(C) lossy link inference methods

Fraction of Lossy Link	**5%**				**10%**				**20%**			
Number of Bad Links	64				105				224			
	R	B	S	C	R	B	S	C	R	B	S	C
Correctly Identified	13	37	39	43	30	75	51	81	46	140	57	144
False Positives	20	12	4	14	62	23	9	24	100	45	15	54

A comparison of COBALT with the three other methods is shown in Table 1. The results on this table are based on measurements from a single server. It is



(a) False positives as a function of monitored servers

(b) Running time as a function of fraction of lossy links

Fig. 1. Performance and running time of COBALT

evident that COBALT provides the best coverage at the expense of a relatively high false positive rate compared to SCFS. SCFS has the lowest false positive rate but its coverage drops dramatically when lossy links are not rare. The Bayesian method finds about 70% of the truly lossy links with false positive rate close to 20%. Finally, random sampling fails to identify more than 30% of the lossy links, while at the same time the number of false positives is very high. Our findings about the Bayesian and random sampling methods are slightly different from the results presented by Padmanabhan *et al* in [2] mainly due to our different loss and topology model.

Figure 1(a) shows that the number of false positives for COBALT, as well as the Bayesian method decreases as the number of measurement points increases. In this scenario, traces from all the servers are combined and both algorithms run over the aggregate collected data. Figure 1(b) shows the running time of Bayesian, SCFS and our approach as the fraction of lossy links increases. The execution time of the heuristics-based methods, SCFS and COBALT, is almost ten times faster than the Bayesian method with Gibbs sampling.

References

1. M. Rabbat, R. Nowak, and M. J. Coates, "Multiple Source, Multiple Destination Network Tomography," in *Proceedings of IEEE INFOCOM 2004*, Apr. 2004.
2. Venkat Padmanabhan, Lili Qiu, and Helen J. Wang, "Server-based Inference of Internet Link Lossines," in *Proceedings of IEEE INFOCOM 2003*, Apr. 2003.
3. Nick Duffield, "Simple Network Performance Tomography," in *Proceedings of Internet Measurement Conference (IMC) 2003*, Oct. 2003.
4. Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the Constancy of Internet Path Properties," in *Proceedings ACM SIGCOMM Internet Measurement Workshop*, Nov. 2001.
5. Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers, "Brite: An approach to universal topology generation," in *Proceedings of MASCOTS 2001*, Aug. 2001.