Practical Passive Lossy Link Inference

Alexandros Batsakis, Tanu Malik, and Andreas Terzis

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

Abstract. We propose a practical technique for the identification of lossy network links. Our scheme is based on a function that computes the likelihood of each link to be lossy. This function mainly 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 solution achieves accuracy comparable to statistical methods (e.g. Bayesian) at significantly lower running time.

1 Introduction and Related Work

Most loss inference techniques [1, 2, 3, 4, 5] attempt to deduce link loss rates from end-to-end measurements. *Active* techniques [1, 2] infer link loss by actively probing the network, while *passive* techniques [3, 4, 5] estimate packet loss by observing the evolution of application traffic. Passive measurements do not require coordination among end points, introduce no additional traffic, and hence, are easier to deploy and do not perturb the state of the network. Depending on the method used to infer packet loss, passive techniques can be further divided to *analytical* [3, 5] and *heuristic* [4]. Analytical techniques detect higher percentage of lossy links while techniques based on heuristics have an advantage on execution speed and resource consumption.

The insight behind this paper is that users care more about finding lossy links affecting the performance of their applications, than finding the exact loss rate of these links. Based on this, 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 lossy paths the link belongs to and how lossy are those paths compared to all the paths in the network.

Preliminary simulation results show COBALT's accuracy to be on par with the analytical methods while its running time is ten times faster. When compared to heuristic methods such as SCFS [4], COBALT infers 20% more truly lossy links, at the expense of higher percentage of false positives. However, the number of false positives decreases as the number of measurement points increases. The low running time of COBALT makes it possible to run the algorithm iteratively with smaller data sets. Each iteration of the algorithm provides new link confidence levels and previous results are incorporated using an exponential moving average. We sketch how this *real-time* variant of COBALT works and argue how it can track the variability in link loss characteristics.

C. Dovrolis (Ed.): PAM 2005, LNCS 3431, pp. 362–367, 2005.

2 Algorithm

Before presenting the details of our method, we briefly introduce the network model we used. In our model clients are connected to servers through a network whose topology is known a-priori. Clients connect and exchange data with the servers using any TCP-based protocol such as FTP or HTTP. We collect a trace of all the packets sent and received by each server. Using these traces, we calculate the loss rate of the path between the server and each client as the ratio of the retransmitted packets to the total number of packets sent by the server. This metric overestimates the actual loss rate, due to the retransmission strategy used by TCP, however our goal is not to estimate the exact link loss rates so this metric is used simply as an estimate of the path loss rate.

COBALT starts by separating lossy from non-lossy paths. Paths with loss rate higher than a user-configurable 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. The default value of T has been set to 1%.

Following path classification, links are categorized depending on the number of good paths they belong to. This approach is similar to the one followed in SCFS and 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. To give a concrete example, if T is set to 2% and link l belongs to a path with estimated loss rate equal to 1.9%, it is difficult to draw conclusions about the probability of the link to be really lossy or not. If on the other hand l belongs to s paths with loss rate less than T we can assume with higher confidence that l is a good link. The parameter s, defined as the sensitivity of the algorithm, depends on the size of the network (i.e. number of hosts and paths) and is user configurable. Higher values of s give higher confidence that the identified links are truly lossy. At the same time the number of false positives increases because some good links might 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 for the remaining links. The confidence level $cfd(l)$ for a link l in a network N is computed as:

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

In the formula above, $avl(l)$ 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 is a constant, with value greater or equal to one. 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. If K is close to one, then $t(l)$ is of little significance in the computation of $cfd(l)$. 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 Algorithm

While existing passive loss inference techniques can detect chronically lossy links, recent results [6] indicate that link characteristics tend to remain stable for only small period of times (approx. 20 minutes). Unfortunately, the running time of current inference methods makes them inappropriate for short timescale changes. Simply put, data analysis requires more time than the time in which link characteristics remain stable. For example, the running time of the Bayesian method in our experiments was 60 minutes for 30 seconds of simulated traffic.

In order to infer network characteristics in a timely manner, while minimizing the storage and computational needs, we propose a real-time variant of our original algorithm. Specifically, we compute the confidence level for each link in short time intervals where the amount of data is small. We then combine the new confidence with our previous knowledge of a link's state, the confidence level estimated at the previous interval, to infer the current conditions.

The online algorithm works similarly to its offline variant but uses an exponential moving average formula to compute the confidence level of a link l:

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

where $cfd_{t_{i-1}}(l)$ the confidence level computed by the previous measurements and $cfd_{t_i}(l)$ the confidence level computed by the most recent data. w is an aging variable ($0 \leq w \leq 1$) controlling the convergence time of the algorithm. If w is close to one, the algorithm will converge faster but it will be more susceptible to oscillations since new data will have more weight. An interesting point in our method is that the value of w might not be constant across all links or even for estimates made for the same link. Its exact value is a function of two parameters:

- The amount of time between two successive runs of the algorithm. If this interval is long, taking into account the temporal link characteristics, the significance of the previous confidence levels decreases, as they reflect an obsolete network image. Hence, in this case the value of w should approach its upper limit. In contrast, if the interval is small then the value of w should be close to zero.
- The number of packets received between t_{i-1} and t_i. Since our method is based on statistics, the larger the sample the more confident we are about the outcome of our analysis. Therefore, as the number of received packets increases, w should approach one.

3 Simulation Results

We used ns-2 to simulate our network of clients and servers. The simulated network topologies were created using BRITE's two-level hierarchical topologies [7]. The net-

Table 1. Comparison of four passive 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

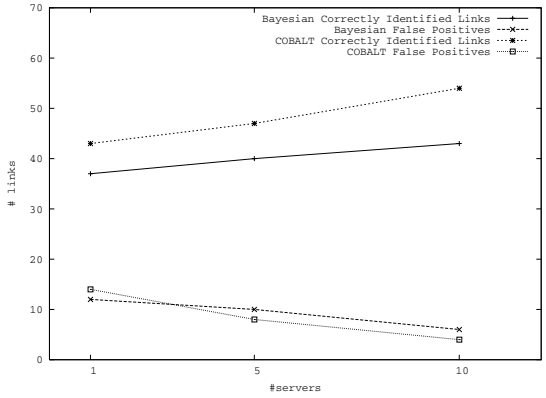
work consists of 800 nodes and about 1400 links. We randomly chose 100 clients out of a pool of 250 to download a large file from the server using HTTP. We also picked a fraction f of the links to be lossy. We then randomly assigned a loss rate to each of the lossy links from a configurable range of loss rates. We used a bimodal loss model, where good links, have a loss rate between $0 - 0.5\%$, and bad links have loss rate between $1.0 - 3.0\%$. This model represents a challenging case for inference algorithms since the difference in loss rates between good and bad links is not significant. We further assume that packet losses are independent.

We compare COBALT to three other methods: Random Sampling [3], Bayesian with Gibbs Sampling [3] and the SCFS algorithm proposed by Duffield [4]. 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}. Thus, non-lossy links whose confidence level exceed this threshold count as false positives. The value of T_{lossy} used in the experimental evaluation is equal to K, the constant used in Equation 1.

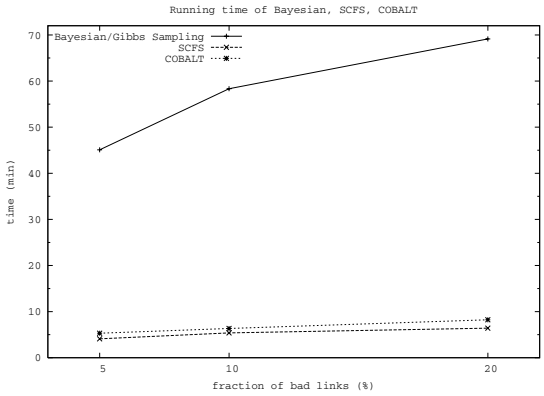
For every experiment presented in this section, we ran the algorithm three times, each time with a different topology. The reported confidence level is the average of the confidence levels obtained over the three executions. We noticed little or no variation on the outcome over the three runs. We choose K to be $3/2$, the sensitivity s of the algorithm is 3, while T is equal to 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. Similarly, for the Bayesian method, the "burn in" period in Gibbs sampling is chosen as 1000 iterations, and links are marked lossy if 99% of the samples found are above the loss rate threshold.

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 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 different from the results presented by Padmanabhan *et al* in [3]. This is because we used a different link loss model in which the loss margin between good and lossy links is narrow. Furthermore, Padmanabhan *et al* used random tree topologies while we use Internet-like topologies.

Figure 1(a) illustrates the performance of COBALT and Bayesian when multiple servers are used. In this scenario, traces from all the servers are combined and both algo-



(a) Performance of COBALT and Bayesian as the number of measurement points increases



(b) Running time comparison of Bayesian with Gibbs Sampling, SCFS, COBALT in a network with 800 nodes and 1400 links

Fig. 1.

rithms run over the aggregate collected data. Notice that by using ten servers, randomly distributed across the network, COBALT improves its coverage and almost eliminates the false positives observed in the previous table.

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.

4 Future Work

We are currently evaluating COBALT across a wider range of simulated topologies in networks with thousands of nodes. We are also evaluating the real-time variant of the algorithm, in terms of its accuracy, execution time, and responsiveness to loss models that change over time.

The biggest challenge is to evaluate our method in the Internet. We are currently collecting and analyzing traces from two campus networks with thousands of users. This analysis will allow us to explore the loss patterns these users experience and provide insights about the actual lossy links in the Internet.

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. Ratul Mahajan, Neil Spring, David Wetherall, and Thomas Anderson, "User-level Internet Path Diagnosis," in *Proceedings of SOSP 2003*, Oct. 2003.
3. Venkat Padmanabhan, Lili Qiu, and Helen J. Wang, "Server-based Inference of Internet Link Lossines," in *Proceedings of IEEE INFOCOM 2003*, Apr. 2003.
4. Nick Duffield, "Simple Network Performance Tomography," in *Proceedings of Internet Measurement Conference (IMC) 2003*, Oct. 2003.

5. G. Liang R. Nowak R. Castro, M. J. Coates and B. Yu, "Internet Tomography: Recent Developments," *Statistical Science*, Mar. 2004.
6. 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.
7. Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers, "Brite: An approach to universal topology generation," in *Proceedings of MASCOTS 2001*, Aug. 2001.