Passive Network Tomography Using Bayesian Inference

Venkata N. Padmanabhan, Lili Qiu, and Helen J. Wang
Microsoft Research

I. Introduction

In this paper, we investigate the problem of identifying lossy links in the interior of the Internet by *passively* observing the end-to-end performance of existing traffic between a server and its clients. This is in contrast to the previous work on network tomography (e.g., [1]) that has been based on active probing. The key advantage of a passive approach is that it does not introduce wasteful traffic which might perturb the object of inference, i.e., the link loss rates. Moreover, our techniques depend only on knowing the number of lost and successful packets sent to each client rather than the exact loss sequence required by previous techniques such as [1]. While accuracy of link loss rate inference may consequently suffer, our techniques can still pinpoint the trouble spots in the network (e.g., highly lossy links).

We have developed three techniques for passive network tomography: Random Sampling, Linear Optimization, and Bayesian Inference using Gibbs Sampling. We have evaluated these techniques using simulations and traces gathered at a busy Web server. In this paper, we focus on the Gibbs Sampling technique; more information on the three techniques appears in [3].

II. Bayesian Inference using Gibbs Sampling

We model passive network tomography as a Bayesian inference problem. We first present some background information.

A. Background

Let D denote the observed data and θ denote the (unknown) model parameters. (In the context of network tomography, D represents the observations of packet transmission and loss, and θ represents the ensemble of loss rates of links in the network.) The goal of Bayesian inference is to determine the *posterior* distribution of θ, $P(\theta|D)$, based on the observed data, D. The inference is based on knowing a *prior* distribution $P(\theta)$ and a *likelihood* $P(D|\theta)$. The *joint* distribution $P(D,\theta) = P(D|\theta)P(\theta)$. We can then compute the posterior distribution of θ as follows:

$$P(\theta|D) = \frac{P(\theta)P(D|\theta)}{\int_\theta P(\theta)P(D|\theta)d\theta}$$

Any features of the posterior distribution are legitimate for Bayesian inference: moments, quantiles, etc. All of these can be expressed as posterior expectations of functions of θ:

$$E(f(\theta)|D) = \frac{f(\theta)P(\theta)P(D|\theta)}{\int_\theta P(\theta)P(D|\theta)d\theta}$$

In general, it is hard to compute $E(f(\theta)|D)$ directly because of the complex integrations involved, especially when θ is a vector (as it is in our case). An indirect approach is to use *Monte Carlo integration*. The idea here is to sample underlying posterior distribution and use the sample mean as an approximation of $E(f(\theta)|D)$. One way of doing the appropriate sampling is to construct a Markov chain whose stationary distribution exactly equals the posterior distribution of interest ($P(\theta|D)$). (Hence the name *Markov Chain Monte Carlo (MCMC)* [2] is given to this class of techniques.) When such a Markov chain is run for a sufficiently large number of steps (termed the *burn-in* period), it "forgets" its initial state and converges to its stationary distribution. It is then straightforward to obtain samples from this stationary distribution.

The challenge then is to construct a Markov chain (i.e., define its transition probabilities) whose stationary distribution matches $P(\theta|D)$. *Gibbs sampling* is a widely used technique to accomplish this. The basic idea is that at each transition of the Markov chain, only a single variable (i.e., only one component of the vector θ) is varied. Rather than explain Gibbs sampling in general, we now switch to modeling network tomography as a Bayesian inference problem and explaining how Gibbs sampling works in this context.

B. Application to Network Tomography

To model network tomography as a Bayesian inference problem, we define D and θ as follows. The observed data, D, is defined as the number of successful packet transmissions to each client (s_j) and the number of failed (i.e., lost) transmissions (f_j). (Note that it is easy to compute s_j by subtracting f_j from the total number of packets transmitted to the client.) Thus $D = \bigcup_j (s_j, f_j)$. The unknown parameter θ is defined as the set of links' loss rates, i.e., $\theta = l_L = \bigcup_{i \in L} l_i$ (We denote a specific solution as $l_L = \bigcup_{i \in L} l_i$ where L is the set of all links in the topology, and l_i is the loss rate of link i.) The likelihood function can then be written as:

$$P(D|l_L) = \prod_{j \in clients} (1-p_j)^{s_j} p_j^{f_j} \quad (1)$$

where $p_j = 1 - \prod_{i \in T_j}(1-l_i)$ and represents the end-to-end loss rate observed at client C_j, and T_j is the set of links on the path from the server to client C_j.

The prior distribution, $P(l_L)$, would indicate prior knowledge about the lossiness of the links. For instance, the prior could be defined differently for links that are known to be lossy dialup links as compared to links that are known to be highly reliable OC-192 pipes. However, in our study here, we only use a uniform prior, i.e., $P(l_L) = 1$, since we do not have information, such as the type or nature of individual links, that could serve as the basis of a prior.

The object of network tomography is the posterior distribution, $P(l_L|D)$. To this end, we use MCMC with Gibbs sampling as follows. We start with an arbitrary initial assignment of link loss rates, l_L. At each step, we pick one of the links, say i, and compute the posterior distribution of loss rate for that link alone conditioned on the observed data D and the loss rates assigned to all other links (i.e., $\{\bar{l_i}\} = \bigcup_{k \neq i} l_k$). Note that $\{l_i\} \cup \{\bar{l_i}\} = l_L$. Thus we have

$$P(l_i|D, \{\bar{l_i}\}) = \frac{P(D|\{l_i\} \cup \{\bar{l_i}\})P(l_i)}{\int_{l_i} P(D|\{l_i\} \cup \{\bar{l_i}\})P(l_i)dl_i}$$

Since $P(l_L) = 1$ and $\{l_i\} \cup \{\bar{l_i}\} = l_L$, we have

$$P(l_i|D, \{\bar{l_i}\}) = \frac{P(D|l_L)}{\int_{l_i} P(D|l_L)dl_i} \quad (2)$$

Using equations (1) and (2), we numerically compute the posterior distribution $P(l_i|D, \{\bar{l_i}\})$ and draw a sample from this distribution. This then gives us the new value, $l_i^{'}$, for the loss rate of link i. In this way, we cycle through all the links and assign each a new loss rate. We then iterate this procedure several times. After the burn-in period (which in our experiments lasts a few hundred iterations), we obtain samples from the desired distribution, $P(l_L|D)$. We use these samples to determine which links are likely to be lossy.

III. Performance Evaluation

We evaluate the inference technique using both simulations and real packet traces. Detailed results appear in [3].

A. Simulation Results

The main advantage of simulation is that the true link loss rates are known, so validating the inferences of network tomography is easy. The simulation experiments are performed on topologies of different sizes using multiple link loss models. For each topology, we set the maximum node out-degree, d, and the fraction of non-lossy links, f (non-lossy links are those whose loss rate is smaller than a threshold).

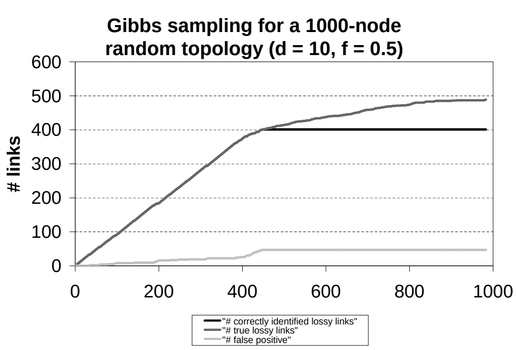

Fig. 1. The performance of Gibbs sampling when the inferences are rank ordered based on a confidence estimate.

Figure 1 shows how Gibbs sampling performs when applied to a 1000-node random topology where about half of the 983 links are lossy. (The number of links is somewhat smaller than the number of nodes because a linear chain of links with no branch points is collapsed into a single virtual link.) Gibbs sampling is able to correctly identify 401 of the 489 lossy links (82%) with only 47 (10.5%) false positives. Other experiments also confirm that inference based on Gibbs sampling has a high coverage and a low false positive rate.

The inferences in Figure 1 are rank ordered based on our "confidence" in the inference. We quantify the confidence as the fraction of Gibbs samples that exceed the loss rate threshold for lossy links. We sort the links in the order of decreasing confidence, and plot 3 curves: the true number of lossy links in the set of links considered up to that point, the number of correct inferences, and the number of false positives. We see that the confidence rating assigned by Gibbs sampling works very well. There are few false positives among the the inferences in which we have the highest degree of confidence.

B. Internet Results

We also applied the Gibbs sampling technique to Internet traffic traces gathered at the *microsoft.com* server site. Network path information was obtained by running *traceroutes* to the clients recorded in the trace.

We found that over 95% of lossy links detected through Gibbs sampling terminate at leaves (i.e., clients). This is consistent with the common belief that the last-mile to clients is often the bottleneck in Internet paths.

Validating our inferences directly is challenging since we do not know the true loss rates of Internet links. We have developed the following approach for indirect validation. The clients in the trace are partitioned into two groups: the tomography set and the validation set. We apply the inference technique to the tomography set to identify lossy links. For each lossy link identified, we examine whether clients in the validation set that are downstream of that link experience a high loss rate on average. If they do, we deem our inference to be correct.

Clearly, this validation method cannot be applied to lossy links that terminate at leaves. For the (small) subset of inferences that could be validated using this method, we found all of the inferences to be correct.

IV. Conclusion

In this paper, we have considered the problem of identifying lossy links in the interior of the Internet based on passive observation at a server of existing end-to-end, client-server traffic. We have develop and evaluated a technique based on Bayesian inference using Gibbs sampling, which has a high coverage (over 80%) and a low false positive rate (below 5-10%).

Acknowledgements

We thank C. Meek, D. Wilson, C. Borgs, J. Chayes, and D. Heckerman for suggesting Gibbs sampling to us.

References

[1] R. Caceres, N. G. Duffield, J. Horowitz, D. Towsley, and T. Bu. Multicast-based inference of network internal characteristics: Accuracy of packet loss estimation. In *Proceedings of IEEE INFOCOM'99*, March 1999.

[2] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1996.

[3] V. N. Padmanabhan, L. Qiu, and H. J. Wang. Server-based inference of internet performance. In *Microsoft Research Technical Report MSR-TR-2002-39*, May 2002.