Choosing an Accurate Network Path Model*

Almudena Konrad
EECS Department, Computer Science Division
University of California at Berkeley
almudena@cs.berkeley.edu

Anthony D. Joseph
EECS Department, Computer Science Division
University of California at Berkeley
adj@cs.berkeley.edu

ABSTRACT

We present a novel domain analysis methodology that enables network researchers to quickly select the most accurate modeling and analysis method or methods for a given wired or wireless network path and network characteristic of interest (*e.g.*, delay, loss, or error process). Our approach includes two classical models, and two data preconditioning models developed in the *Tapas* project.

Categories and Subject Descriptors

I.6.4 [**Computing Methodologies**]: Model Validation and Analysis

General Terms

Performance

Keywords

Markov, modeling, network characteristics, traces, wireless

1. INTRODUCTION

Perhaps the most common method for evaluating application and network protocol designs while they are under development is the use of network link simulation. Simulation is a low-cost technique that enables networking researchers to quickly, and in a repeatable manner, explore the behavior of a network or application protocol under a variety of network conditions (*e.g.*, varying loss, delay, and error). However, the behavior and thus, the results and performance, of many protocols depends on the characteristics of the network conditions. Designers of such algorithms and protocols make assumptions about the way in which a particular network characteristic varies and encode these assumptions into the algorithms. The testing of application and protocol behavior in a simulator depends upon an accurate modeling of the behavior of the network under test.

In this paper we address one of the most important problems in statistics: the choice of an appropriate model for characterizing a given dataset. We encounter this same problem in computer networks, where many design decisions are the results of some chosen simulation parameters and models. In analyzing computer networks, researchers are faced with measurements whose characteristics experience non-stationarity (time variability) and complex patterns due to a large number of factors, including both internal network elements and external events.

The traditional approach to modeling networks is the use of a classical network model, such as a Bernoulli, Gilbert, high-order Discrete Time Markov Chain (DTMC), or Hidden Markov Model (HMM). The choice of which model to use is usually an ad hoc one, often without adequate consideration of the statistical properties of each model. However, in [2], we show that the non-stationary characteristics of some traces cannot be adequately characterized or modeled using classical techniques. In this paper, we introduce a new modeling methodology, data preconditioning, that takes into consideration the time varying statistical properties of today's networks, and introduce a domain-specific model choice methodology.

2. METHODOLOGY

In [3], we developed a data preconditioning approach, Markov-based Trace Analysis (MTA), that more accurately models error distribution in a digital cellular telephony data network than traditional modeling techniques. In this paper, we generalize that work by applying it to loss and delay networks path traces collected from three different wired and wireless networks, introduce the Modified hidden Markov Model (M^3) algorithm (a new data preconditioning algorithm), and address the challenge of domain analysis: *choosing the most appropriate model to use for modeling the behavior of an arbitrary network path and characteristic.*

We analyze event traces (sequences of 0's and 1's) of: IP packet losses, wireless frame errors, and packet delays. In each trace, a 0 denotes the lack of the event. In a loss trace, a 1 signifies a lost packet, in an error trace, a 1 is a corrupted frame, and in a delay trace, a 1 means that the packet arrived with a delay greater than some maximum threshold. In general, we refer to a 1 in a trace, as an *error frame*. Our research [1] shows that these traces can be divided into short clusters of 1's and 0's (lossy states), and long clusters of just 0's (error-free states). Lossy states begin with a 1 and contain a burst of 1's and 0's, and end with a burst of 0's of length equal to a *change-of-state* variable C. The next 0 element following the burst of C 0's begins an error-free state, which ends with the 0 preceding the next 1 element in the trace. Extensive trace analysis shows that a good approximation for C is the mean plus one standard deviation of the length of error bursts in a trace. We approximate the

*The full version of this paper is available in [1].

SIGMETRICS'03, June 10–14, 2003, San Diego, California, USA.
ACM 1-58113-66401/03/0006.

length distributions of lossy and error-free states using an exponential distribution function, where the smaller the exponential parameter, the larger the average cluster (state) length. We can then characterize traces using a tuple of $(L_{exp}, EF_{exp}, L_{den})$, where L_{exp} and EF_{exp} are the parameters of the lossy and error-free state length exponential distribution, and L_{den} is the error density in the lossy state (*i.e.*, the probability of a 1 event in a lossy state).

3. DATA PRECONDITIONING

Network behavior is inherently time-varying (non-stationary), an observation that led us to propose a new research methodology that analyzes and preconditions trace data *before* the data is fed into traditional models. Intuitively, we use pattern recognition to decompose datasets that experience non-stationarity into subsets that exhibit stationary behavior, and hence are easier to accurately model with traditional models. For a particular network characteristic, we first identify data patterns that exhibit stationarity [1] and suggest an underlying process consisting of some number of states (specific data patterns corresponding to a particular network behavior). We typically identify two distinct states: lossy and error-free. Second, we concatenate subtraces from the same states to form stationary subtraces of the original trace. These subtraces have the property that they can be modeled using a high-order DTMC. Note that there will be as many subtraces as states. Finally, we use Markov models (or other similar modeling techniques) to calculate the transition probabilities between states.

We summarize two preconditioning examples, the MTA and M^3 algorithms. The MTA algorithm [2] decomposes traces into two states, lossy and error-free. MTA concatenates all the lossy states to form a *lossy subtrace*, and concatenates the error-free states to form *error-free subtrace*. Lossy subtrace now exhibits stationarity and can be modeled using a high-order DTMC, and MTA computes the memory and transitions probabilities. Finally, MTA approximates both lossy and error-free states' length distributions using exponential distribution functions and computes the exponential functions' parameters using a fitting function that yields a Cumulative Distribution Function (CDF) that has the highest correlation coefficient, *cc*, with the trace's CDF curve (for accurate models, *cc* is greater than 0.96 [1]).

Unlike MTA, the M^3 algorithm can model traces with two or more data patterns and non-exponential state length distributions. Similar to a HMM, M^3 views each data pattern as a hidden state, and it models the transition between states with a high order DTMC. Using the data preconditioning approach, the M^3 algorithm concatenates subtraces from each of the same hidden states encountered in the original trace to form subtraces, and then models each subtrace with a high order DTMC. Intuitively, this new algorithm can be viewed as a new type of hidden Markov process [4], where the output independence assumption is *not* taken.

4. DOMAIN OF APPLICABILITY

We apply our data preconditioning models and two classical models to synthetic traces to identify the domain of applicability for each model (*i.e.*, which model best models a given trace characteristic). First, we generate artificial traces for values of L_{exp}, EF_{exp}, and L_{den}. Next, for each model and each trace, we calculate the mean of the *cc* values for the error and error-free burst CDFs.

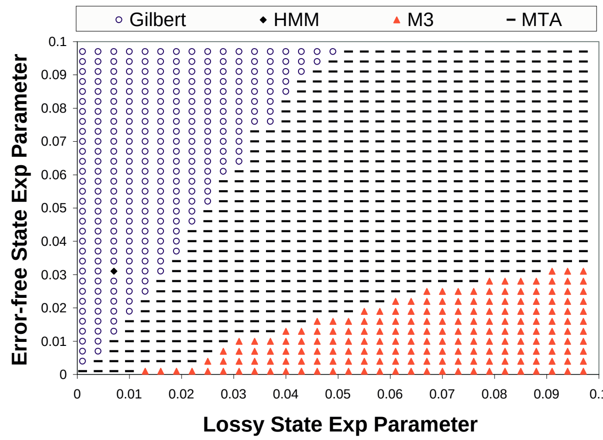

Figure 1: Optimal model for $L_{den} = 0.4$.

We construct a Domain Applicability Plot (DAP) for an L_{den} value, where each point in a plot indicates the best model (maximum mean *cc* value) for a (L_{exp}, EF_{exp}) tuple.

Due to space restrictions, we only show the $L_{den} = 0.4$ DAP (Figure 1). Observe that there are three optimal regions: "optimal-Gilbert" (mean *cc* 0.99), "optimal-MTA" (0.98), and "optimal-M^3" (0.97). In each region, other models have mean *cc* values less than 0.93.

We expect these results as the synthetic trace uses a Bernoulli process to generate losses in the lossy state yielding relatively small error burst lengths. Thus, as the length of the error-free state increases, the region occupied by the Gilbert model decreases and the M^3 and MTA models become better choices.

The $L_{den} = 0.2$ DAP shows that the Gilbert model is best for a large portion of the graph (mean *cc* 0.99), however M^3 also performs very well for this region (0.98). In the "optimal-M^3" region (0.97), the Gilbert model mean *cc* is 0.96. Thus, for $L_{den} = 0.2$, the M^3 model always yields highly accurate models, while the Gilbert model only performs best for a subset of the network parameter space.

Finally, the $L_{den} = 0.7$ DAP consists almost entirely of an "optimal-M^3" region (0.98), where MTA also performs well (0.97), while both the Gilbert and HMM perform very poorly. We believe that this result is due to the inability of traditional models to capture the long error bursts inside lossy states, a characteristic found in most wireless traces. In contrast, the data preconditioning models accurately capture both low and high error densities inside lossy states.

5. REFERENCES

[1] A. Konrad and A. D. Joseph. Choosing an accurate network path model. Technical Report UCB//CSD-03-1236, UC Berkeley, 2003.

[2] A. Konrad, B. Zhao, A. D. Joseph, and R. Ludwig. A Markov-based channel model algorithm for wireless networks. In *Proc. of MSWiM '01*, 2001.

[3] R. Ludwig, A. Konrad, and A. D. Joseph. Optimizing the end-to-end performance of reliable flows over wireless links. In *Proc. of ACM MobiCom '99*, 1999.

[4] I. L. MacDonald and W. Zucchini. *Hidden Markov and Other Models for Discrete-valued Time Series*. 1997.