

An Algorithm for Fitting MMPP to IP Traffic Traces

Soshant Bali and Victor S. Frost, *Fellow, IEEE*

Abstract—A method for fitting very long IP traffic traces to a Markov Modulated Poisson Process (MMPP) model is proposed in this paper. We compare our method to a previously published technique using several 24 hour traces. Our method is useful when the traces are very long and exhibit diurnal traffic changes.

I. INTRODUCTION

MMPP is an attractive model for backbone Internet traffic. It has been recognized that at the packet or byte level Internet traffic is not modeled well using the Poisson process [2].

The contribution of this paper is a new algorithm for fitting MMPP to an IP traffic trace. Some of the previous approaches to estimating MMPP parameters use the maximum likelihood estimator (MLE) method [3]. The MLE method is computationally expensive and requires that the number of states be specified as an input in advance. In [1] a computationally inexpensive method was proposed that does not require the user to input the number of states. It was shown that the method from [1] results in a reasonably accurate MMPP model for medium sized traces. However, this method may not be very accurate when there is a need to model longer traces that exhibit diurnal traffic variations [4]. In this paper we modify the method proposed in [1] and show, using several 24 hour traces, that the new method provides a better model for longer traces at the expense of an increased number of states. Although it is not desirable to have a large number of MMPP states, the algorithm proposed in this paper can be used to generate an accurate MMPP model of long traffic traces when computational resources for dealing with an increased number of MMPP states are available.

II. FITTING DATA TO A MMPP

The algorithm for fitting MMPP is divided into four parts. Part A, C and D of this algorithm are similar to the corresponding parts in algorithm 1 (see [1]). Part B is new and allows multiple states to have the same rate. The algorithm in part A computes the basic rates needed for modeling the data. These rates are assigned to states using the procedure discussed in part B. Once the states are assigned, the state transition probability matrix of the discrete MMPP (D-MMPP) is computed in part C. Finally part D computes the rate matrix for MMPP from the state transition probability matrix of D-MMPP.

A. Choosing the Rates

Let $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N$ be the rates, where N is unknown. λ_1 should cover the largest observation, so choose

$$\lambda_1 + 2\sqrt{\lambda_1} = \text{peak of the data.}$$

Authors are with Information and Telecommunication Technology Center, Department of Electrical Engineering and Computer Science, University of Kansas. This work was supported in part by NSF grant ANI-0125410.

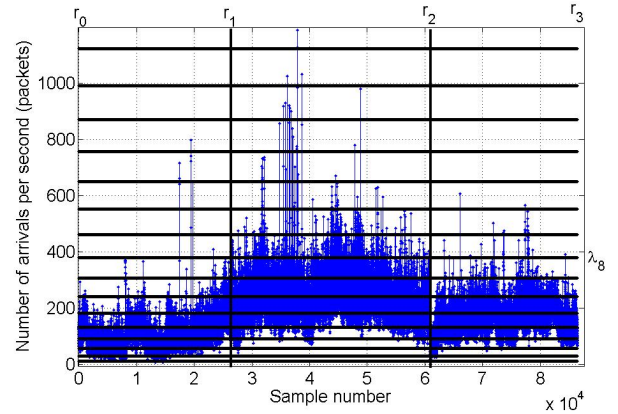


Fig. 1. 24 hour long sample trace with rates shown as horizontal lines

Solving this equation, we get

$$\lambda_1 = (\sqrt{1 + \text{peak}} - 1)^2.$$

The lower bound of data covered by any state with rate λ_1 is $\lambda_1 - 2\sqrt{\lambda_1}$, which is chosen as the upper bound of rate λ_2 , i.e.,

$$\lambda_2 + 2\sqrt{\lambda_2} = \lambda_1 - 2\sqrt{\lambda_1}.$$

Solving, we get

$$\lambda_2 = (\sqrt{\lambda_1} - 2)^2.$$

This procedure can be repeated to get the values of λ_3 from λ_2 , λ_4 from λ_3 and so on. The stopping point is the minimum rate in the data. Rate λ_N covers the minimum of the data.

B. Choosing the States

Figure 1 shows a plot of 24 hours of data collected in Feb-April 2000 at the University of Auckland Internet uplink¹. Rates determined using step A are shown as horizontal lines in this figure. Consider rate λ_8 (marked in the figure) and note that this rate is visited rarely by data points between reference points r_0 and r_1 ; and frequently between r_1 and r_2 . If in the MMPP model of this trace we decide to assign only one state with rate λ_8 , then the mean time in this state determined using the entire trace is going to be too small to model data points between r_1 and r_2 and too high to model data points between r_0 and r_1 . However, if we decide to assign two states with rate λ_8 , one state having a mean time determined using data points between r_0 and r_1 , and the other state having a mean time in state determined using data samples between r_1 and r_2 , then the trace can be modeled more accurately. The algorithm that follows identifies regions in a given traffic trace that can be modeled using same set of states (referred to as state sets). Regions that cannot be modeled using the same set of states are assigned new state sets as needed.

¹Data from NLANR Measurement and Network Analysis Group

Let L be a set consisting of all rates, i.e., $L = \{\lambda_1, \lambda_2, \dots, \lambda_N\}$. Define “state set” (denoted as SS) to be a set of states with rates chosen from set L in such a way that no two states in a state set have the same rate. For example, consider a state set SS_i consisting of states $\{s_{i1}, s_{i2}, \dots, s_{ip}\}$. Here, state s_{ij} is a state in set SS_i and has a rate λ_j , $\lambda_j \in L$. Let there be a total of n_{SS} different state sets. Then, $SS_1 \cup SS_2 \cup \dots \cup SS_{n_{SS}} = U$, where U is the universal set consisting of all the MMPP model states.

Given an IP traffic trace, the new algorithm forms the state sets SS_i and the number n_{SS} using the following procedure. Divide the IP traffic trace into non-overlapping windows, each of size W samples. If there are a total of n_t samples in the IP traffic trace, then the trace can be divided into at most $n_W = \lfloor \frac{n_t}{W} \rfloor$ non-overlapping windows of size W .

Two sets D_i and V_i are associated with each window i . Set D_i is the set of dominant rates (defined below) in window i and set V_i is the set of rates visited in window i . A sample j from IP traffic trace having a value x_j is defined to “visit” rate λ_k if $\lambda_k - 2\sqrt{\lambda_k} < x_j < \lambda_k + 2\sqrt{\lambda_k}$. If rate λ_k is visited by at least one of the samples in window i , then $\lambda_k \in V_i$. Let $n_{\lambda_k}^i$ be the number of samples in window i that visit rate λ_k . Dominant set D_i is the set of rates most frequently visited by samples in window i , and is formed using the following procedure. Sort elements of set V_i in decreasing order of the number of times the rates are visited to form the set V_i^s . Let D_i be an empty set. Traverse the elements of set V_i^s in the forward order (rate visited maximum number of times first). If the current element from set V_i^s is λ_k , add rate λ_k to set D_i if

$$\frac{\sum_{\lambda_l \in D_i} n_{\lambda_l}^i + n_{\lambda_k}^i}{W} < \eta, \quad (1)$$

where η is a constant (e.g., $\eta = 0.85$). Keep adding rates to D_i until the first element λ_k from set V_i^s is encountered for which (1) does not hold. Set D_i is now the dominant set.

Once the sets D_i and V_i are computed $\forall i, i \in \{1, 2, \dots, n_W\}$, the algorithm divides the windows into groups that can be modeled using the same state sets. Initialize $n_{SS} = 1$ and $SS_1 = \emptyset$ (empty set). Perform $SS_1 \leftarrow V_1$, where the operation $SS_1 \leftarrow V_1$ creates new states in SS_1 such that SS_1 contains all states with rates in $SS_1 \cup V_1$. For example, if $V_x = \{\lambda_3, \lambda_4\}$ and $SS_y = \{s_{y2}, s_{y3}\}$, then after the operation $SS_y \leftarrow V_x$, $SS_y = \{s_{y2}, s_{y3}, s_{y4}\}$. Also initialize $D_1^{SS} = D_1$, where D_1^{SS} is the dominant set associated with state set SS_1 (each state set SS_k will have a dominant set D_k^{SS} associated with it). For each window i , let w_i^{SS} be the state set number for the state set that is best suited to model the samples in that window. Traverse the windows in the forward order from $i = 2$ to $i = n_W$. Let j be the state set index, then for each j , $j \in \{1, 2, \dots, n_{SS}\}$, compute $D_i - D_j^{SS}$ and $D_j^{SS} - D_i$ (set subtraction). Also compute $|D_i - D_j^{SS}|$ and $|D_j^{SS} - D_i|$ (number of elements in the set). Check for the condition

$$|D_i - D_j^{SS}| + |D_j^{SS} - D_i| \leq 2. \quad (2)$$

If (2) holds for some $j = l$, then samples in window i can be modeled using state set SS_l . Thus perform the operations $w_i^{SS} = l$ and $SS_i \leftarrow V_i$. However, if (2) does not hold for all j , $j \in \{1, 2, \dots, n_{SS}\}$, then a new state set should be created. Perform $n_{SS} = n_{SS} + 1$, $SS_{n_{SS}} = \emptyset$ and $SS_{n_{SS}} \leftarrow V_i$. These operations create a new state set and assign appropriate states

to the state set. Once the new state set is created, assign $w_i^{SS} = n_{SS}$ where i is the current window. Also associate a dominant set with the new state set, i.e., $D_{n_{SS}}^{SS} = D_i$. After the algorithm has traversed all of the n_W windows and performed the operations discussed above, the resultant state sets contain the desired states of the D-MMPP. The set U contains all of the states for the D-MMPP model of the traffic trace.

C. Fitting the Markov Chain

Consider a sample x_k that belongs to some window i . The samples in window i can be modeled using states in state set $SS_{w_i^{SS}}$. Then there exists exactly one state $s_{w_i^{SS}l} \in SS_{w_i^{SS}}$ such that $\lambda_l - 2\sqrt{\lambda_l} < x_k < \lambda_l + 2\sqrt{\lambda_l}$. The state $s_{w_i^{SS}l}$ defines the phase of sample x_k and is denoted as ϕ_k . As the next step, the phase of each sample is computed. Then for each observation $\{x_k, k = 1, 2, \dots, (n_W \times W)\}$ there exists a corresponding phase $\{\phi_k, k = 1, 2, \dots, (n_W \times W)\}$ (Since set U consists of all the states, $\phi_k \in U$ for all k). Let the set U contain n_u states, i.e., $U = \{u_1, u_2, \dots, u_{n_u}\}$. Let $P = (p_{u_i u_j})$ be the transition matrix of the phase process. The MLE of $p_{u_i u_j}$ is

$$p_{u_i u_j} = \frac{\text{number of transitions from } u_i \text{ to } u_j}{\text{number of transitions out of } u_i}. \quad (3)$$

Matrix P obtained using (3) is the required state transition matrix for the D-MMPP.

D. Fitting an MMPP

The rate matrix Q of an MMPP can be obtained from P matrix of the D-MMPP using

$$q_{u_i u_j} = p_{u_i u_j}, i \neq j; q_{u_i u_i} = p_{u_i u_i} - 1. \quad (4)$$

If Q is derived from P using (4), then the mean sojourn time in state u_i (for every i) is the same for the two processes (D-MMPP and MMPP).

III. RESULTS

Note that in Figure 1 the mean arrival rate is about 100 packets/sec from the beginning of the trace to sample number 2×10^4 . In the region between sample numbers 2×10^4 and 3×10^4 mean arrival rate increases from 100 packets/sec to about 250 packets/sec. It then stays high at about 250 packets per second until sample number 6×10^4 , where it decreases to 200 packets/sec. We fitted MMPP to this trace using both algorithm 1 from [1] and the new algorithm proposed in this paper. Part A of the algorithm generates a set of 16 rates for this trace. Since algorithm 1 allows only one state per rate, MMPP model fitted using algorithm 1 has 16 states, each having a unique rate. About 24 hours of data generated using algorithm 1’s MMPP model is plotted in Figure 2. Note that mean arrival rate is constant throughout the trace, unlike in the original trace where the mean arrival rate changes with the time of the day. MMPP model fitted to the same trace (shown in Figure 1) using the new algorithm has a total of 48 states with a total of 4 state sets ($n_{SS} = 4$). The four state sets have 12, 12, 16 and 8 states. About 24 hours of data generated using new algorithm’s MMPP model is shown in Figure 3. The mean arrival rate changes with time for this trace as the state transitions from one state set to another.

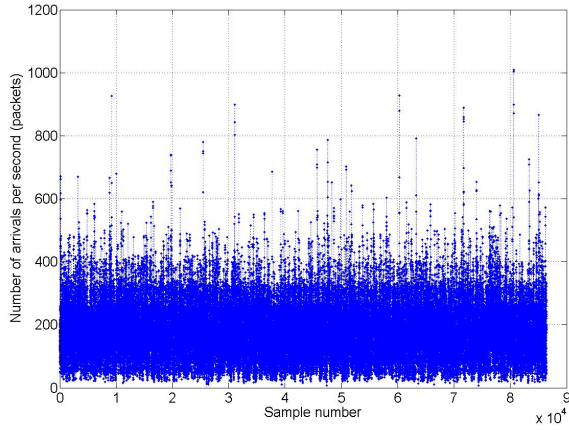


Fig. 2. Trace generated from a MMPP model of the original trace in Figure 1 (using Algorithm 1)

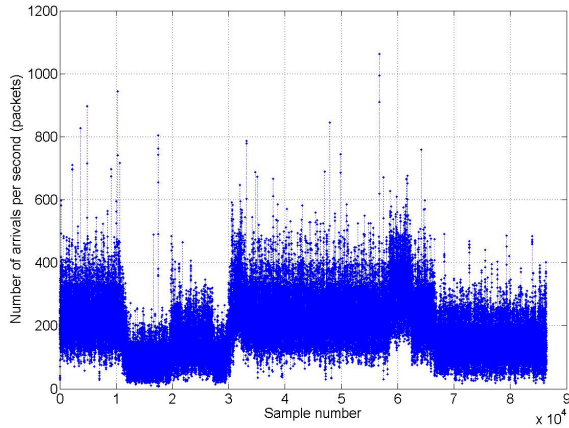


Fig. 3. Trace generated from a MMPP model of the original trace in Figure 1 (using the new algorithm)

We used two different methods to evaluate how well the traces generated using the two MMPP models fit the original trace. The first compares the mean queuing delays and the second compares the loss rates of the original trace with that of the traces generated using the two models. A fluid flow simulation model [5] was used to determine the mean queuing delays and loss rates. Packet size is assumed to be constant (1000 bits) for the simulation. Service rate of the queue is computed from the mean arrival rate of the entire trace and the load (mean arrival rate/load). Mean queuing delays for the three traces are plotted in Figure 4. Queuing delays for the trace generated using algorithm 1 are close to the queuing delays of the original trace only for low loads (less than 0.6). Queuing delays for the trace generated using the new algorithm follows the queuing delays of the original trace very closely even for loads greater than 0.6. Loss rate plots for the three traces are shown in Figure 5. The load ranges from 0.4 to 0.99 since the three traces have very similar loss rates for loads less than 0.4. When the buffer size is 50 packets, loss rates for the three traces are almost the same. With a buffer size of 5000 packets, the new algorithm models the loss rate better than algorithm 1. Clearly the new method's MMPP model is a better fit than algorithm 1's MMPP model. The MMPP was fitted to two other 24 hour long data trace using the two algorithms. Both these traces were collected at the University of Auckland Internet uplink, the first one

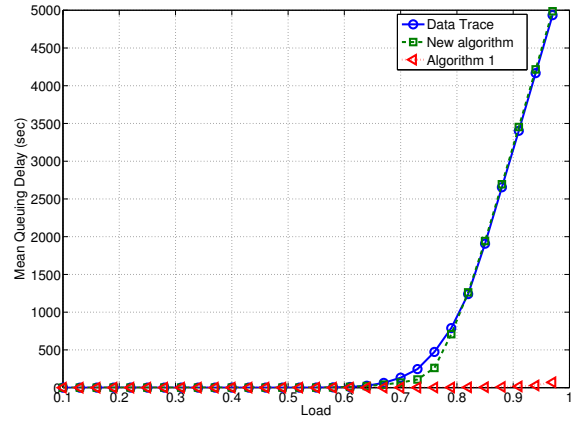


Fig. 4. Mean queuing delays for the three traces (Data set 1)

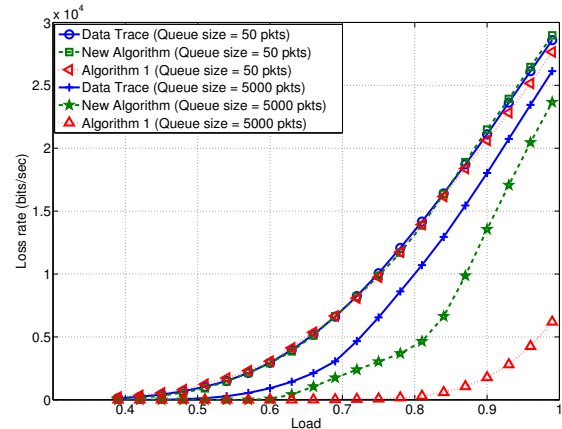


Fig. 5. Loss rates for queue sizes 50 pkts and 5000 pkts (Data set 1)

in Feb-April 2000 and the second one in December 2003. Mean queuing delays and loss rates for these data sets are similar to the ones above. For a data trace with 86,400 data points, Algorithm 1 took 50.8 sec and the new algorithm took 154.7 sec to generate the transition probability matrix from that 24 hour trace on a machine with a Pentium Xeon 3.2Ghz processor.

IV. CONCLUSION

A new method for modeling MMPP to IP trace was proposed. It was shown by comparing mean delay and packet loss rates that this method models the trace better than a previously published method.

REFERENCES

- [1] Daniel P. Heyman and David Lucantoni, Modeling Multiple IP Traffic Streams with Rate Limits, *IEEE/ACM Transactions on Networking*, Vol. 11, No. 6, pp. 948-958, December 2003.
- [2] T. Karagiannis, M. Molle, and M. Faloutsos, Long-range dependence ten years of Internet traffic modeling, *IEEE Internet Computing*, Vol. 8, No. 5, pp. 57-64, Sep.-Oct., 2004.
- [3] A. Andersson and T. Ryden, Maximum Likelihood Estimation of a Structured MMPP with Applications to Traffic Modeling, *13th ITC Specialist Seminar*, Monterey, CA, 2000.
- [4] S. Bali, Y. Jin, V. S. Frost and T. Duncan, Characterizing User-perceived Impairments Using End-to-end Measurements, *International Journal of Communication Systems*, Vol. 18, No. 10, pp. 935-960, December 2005.
- [5] Cameron Kiddle, Rob Simmonds, Carey Williamson and Brian Unger, Hybrid Packet/Fluid Flow Network Simulation, *Seventeenth Workshop on Parallel and Distributed Simulation*, June 10-13, 2003.